Navid Malek
navidmalekedu@gmail.com
navidmalek.blog.ir

Transparently Proxy All Traffic Through TOR

Requirements:

- Linux OS (tested on alpine and ubuntu)
- iptables (Linux firewall)
- RedSocks

What is RedSocks?

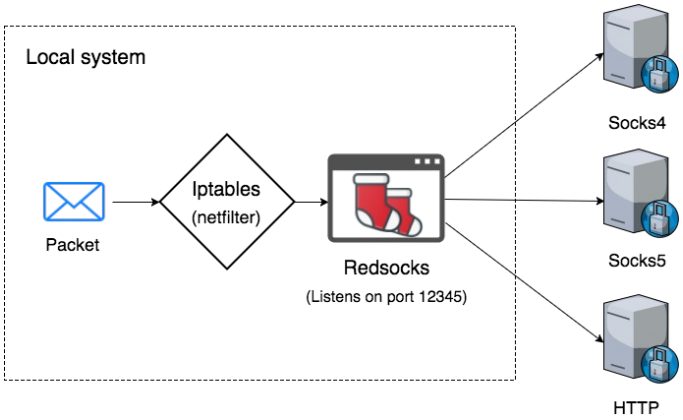
Reference: https://github.com/darkk/redsocks

Redsocks is the tool that allows you to proxify(redirect) network traffic through a **SOCKS4**, **SOCKS5** or **HTTPs** proxy server. It works on the lowest level, the kernel level (**iptables**). The other possible way is to use **application level proxy**, when **the proxy client** is implemented in the same language as an application is written in. **Redsocks** operates on the lowest system level, that's why all running application don't even have an idea that network traffic is sent through a proxy server, as a result it is called a **transparent proxy redirector**.

TCP Traffic

System's Architecture and Setup

So this is the big image, almost every tcp packet will be redirected to port 12345 which redsocks service listens for incoming packets; after that, redsocks will redirect the received traffic to another ip and port in socks protocol format.



Navid Malek
navidmalekedu@gmail.com
navidmalek.blog.ir

Walkthrough Steps

1. Configure Redsocks to listen for tcp traffic on port 12345

   For this, edit or create `/etc/redsocks.conf` file with the following configuration:

```
base {
 log_debug = on;
 log_info = on;
 log = "stderr";
 daemon = off;
 redirector = iptables;
}

redsocks {
    local_ip = 127.0.0.1;
    local_port = 12345;

    ip = DESTINATION_IP ;
    port = DESTINATION_PORT ;
    type = socks5;
      // known types: socks4, socks5, http-connect, http-relay

    // login = username;
    // password = password;
}
```

sample configuration file for my work:

```
base {
 log_debug = on;
 log_info = on;
 log = "stderr";
 daemon = off;
 redirector = iptables;
}

redsocks {
    local_ip = 127.0.0.1;
    local_port = 12345;

      ip = 192.168.0.3;  #IP OF TOR CONTAINER
      port = 9150;  #PORT OF TOR IN CONTAINER
    type = socks5;
      // known types: socks4, socks5, http-connect, http-relay
```

Navid Malek
navidmalekedu@gmail.com
navidmalek.blog.ir

```
    // login = username;
    // password = password;
}
```

the tor container I've used: https://hub.docker.com/r/peterdavehello/tor-socks-proxy/

2. After configuring redsocks service, you need to configure iptable rules:

```
iptables -t nat -N REDSOCKS

iptables -t nat -A REDSOCKS -d 0.0.0.0/8 -j RETURN
iptables -t nat -A REDSOCKS -d 10.0.0.0/8 -j RETURN
iptables -t nat -A REDSOCKS -d 127.0.0.0/8 -j RETURN
iptables -t nat -A REDSOCKS -d 169.254.0.0/16 -j RETURN
iptables -t nat -A REDSOCKS -d 172.16.0.0/12 -j RETURN
iptables -t nat -A REDSOCKS -d 192.168.0.0/16 -j RETURN
iptables -t nat -A REDSOCKS -d 224.0.0.0/4 -j RETURN
iptables -t nat -A REDSOCKS -d 240.0.0.0/4 -j RETURN

iptables -t nat -A REDSOCKS -p tcp -j REDIRECT --to-ports 12345

iptables -t nat -A OUTPUT -p tcp -j REDSOCKS

iptables -t nat -A PREROUTING -p tcp -j REDSOCKS
```

Note that this rules are temporary! For persisting rules see here:
https://askubuntu.com/questions/119393/how-to-save-rules-of-the-iptables

main reference for step one and two:
https://superuser.com/questions/1401585/how-to-force-all-linux-apps-to-use-socks-proxy

3. start redsocks service with this command:

```
sudo redsocks -c /etc/redsocks.conf
```

4. now all your TCP traffic will transparently go through tor network! Just test it

Consider defining a systemd service for redsocks.
Reference: https://medium.com/@benmorel/creating-a-linux-service-with-systemd-611b5c8b91d6

Navid Malek
navidmalekedu@gmail.com
navidmalek.blog.ir

UDP Traffic

For udp traffic, at first glance things seem to be more complicated but instead thanks to DNSPort of TOR the problem can be easily solved!

Main references:

https://en.wikibooks.org/wiki/How_to_Protect_your_Internet_Anonymity_and_Privacy/TOR_VPN

https://trac.torproject.org/projects/tor/wiki/doc/TransparentProxy

1. first edit tor's configuration file /etc/tor/torrc and add the following line at the end:

   DNSPort 53

2. start tor and add the following rules to iptables:

```
iptables -t nat -A OUTPUT -p udp --dport 53 -j DNAT --to TOR_IP:TOR_DNSPort
iptables -t nat -A OUTPUT -p tcp --dport 53 -j DNAT --to TOR_IP:TOR_DNSPort
```

   sample iptable rules for my set:

```
iptables -t nat -A OUTPUT -p udp --dport 53 -j DNAT --to 192.168.0.3:9053
iptables -t nat -A OUTPUT -p tcp --dport 53 -j DNAT --to 192.168.0.3:9053
```

3. You are all setup! Just start the redsocks and now all your traffic goes through tor transparently.

Navid Malek
navidmalekedu@gmail.com
navidmalek.blog.ir

More Technical Information

When I was looking for more information on the Internet I've come to this article which is pretty awesome and detailed from Linux's technical networking perspective.

The article: https://crosp.net/blog/administration/install-configure-redsocks-proxy-centos-linux/