BECOMING SECURE AND ANONYMOUS

In this chapter, we look at how we can navigate the world wide web anonymously or as close as we can get using four methods:

1. The Onion Networks
2. Proxy Server
3. Virtual Private Networks
4. Private Encrypted Email

These methods will likely make the tracker's job more difficult.

HOW THE INTERNET GIVES US AWAY:

First: our IP address identifies us as we traverse the internet. Data sent from our machines is generally tagged with our IP address, making our activities easy to track.

Second: Google and other email services will *read* our email, looking for keywords to more efficiently serve us ads.

When we send a packet data across the internet, it contains the IP address of the source and destination for the date. Each packet hops through multiple internet routers until it finds its destination and then hops back to the sender. For general internet surfing, each hop is a router the packet passes through to get to the destination. There can be as 20-30 hops between the sender and receiver, but usually any packet will find its way to the destination in fewer than 15 hops.

As the packet traverse the internet, anyone intercepting the packet can see who sent it, where it has been, where it's going.

This is one-way websites can tell us who we are when arrived and log in automatically, and it's also how someone can track where we've been on the internet.

To see what hops a packet might make between us and destination, we can use the `traceroute` command.

```
# Syntax: traceroute des_IP_address/domain
```

This command will send out packets to the destination and trace the route of those packets.

```
# Viz: traceroute google.com
```

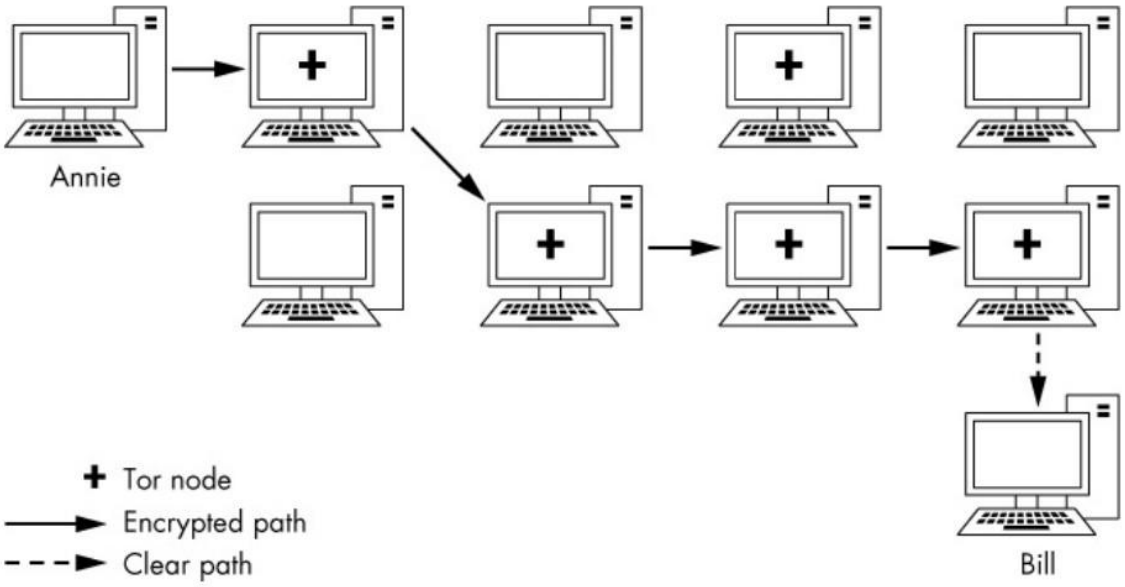
Results will likely be different because request would be coming from a different location and because google has many servers across the globe and also packets don't always take the same route across the internet.

THE ONION ROUTER SYSTEM:

In the 1990s, the US Office of Naval Research (ONR) set out to develop a method for anonymously navigating the internet for espionage purposes. The plan was to set up a network of routers that was separate from the internet's routers, that could encrypt the traffic, and that only stored the unencrypted IP address of the previous router – meaning all other router address along the way encrypted. The idea was that anyone watching the traffic could not determine the origin or destination of the data. This research became known as "The Onion Router (TOR) Project" in 2002, and now it's available to anyone to use for relatively safe and anonymous navigation on the web.

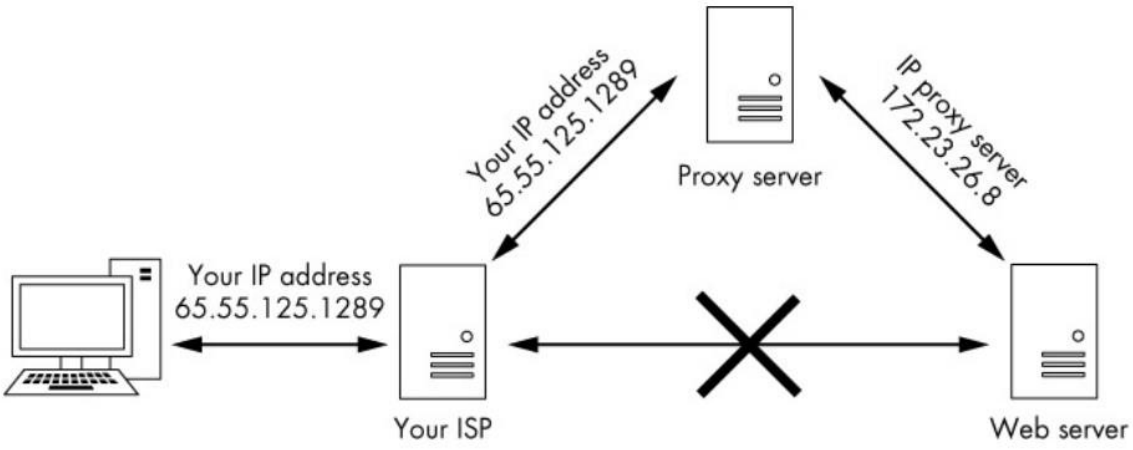
HOW TOR WORKS:

On top of using a totally separate router network, Tor encrypts the data, destination and the sender IP address of each packet. At each hop, the information is encrypted and then decrypted by the next hop when it's received. In this way, each packet contains info about only the previous hop along the path and not the IP address and the origin.



PROXY SERVERS:

An intermediate system that acts as a middleman for traffic: the user connects to a proxy, and the traffic is given the IP address of the proxy before it's passed on. When the traffic returns from the destination, the proxy sends the traffic back to the source. In this way, the traffic appears to come from the proxy and not to change the originating IP address.



To make our traffic even harder to trace, we can use more than one proxy, in a strategy known as a proxy chain.

Kali linux has an excellent proxying tool called proxy chains that we can set up to obscure our traffic.

`Syntax: proxychains <the command we want proxied> <arguments>`

`Viz: proxychains nmap -sT -Pn <IP address>`

This would send the `nmap -sS` stealth scan command to the given IP through a proxy. The tool then builds the chain of proxies itself, so you don't have to worry about it.

SETTING PROXIES IN THE CONFIG FILE:

As with nearly every application in Linux/Unix, configuration of proxychains is managed by the config file–specifically `/etc/proxychains.conf.`

On the line number 61 [ProxyList], we can add proxies by entering the IP addresses and ports of the proxies we want to use in the list.

A free proxy: http://www.hidemy.name

On that list, add proxy:

Syntax: Type IP Port

Viz: http 103.122.202.2 8080

It is important to note that proxychains defaults to using TOR if we don't enter any proxies of our own.

The default TOR config: socks4 127.0.0.1 9050

If we are not adding our own proxies and want to use TOR, leave the file as it is. And if we are not using TOR, we will need to comment out this (socks4 127.0.0.1 9050) line.

The command is:

```
Kali> proxychains firefox www.google.com
```

This succefully opens the google in firefox through our chosen proxy and returns the results to us. To anyone tracking this traffic, it appears that it was our proxy that navigated to the google rather than our IP address.

SOME MORE INTERESTING OPTIONS:

We can put multiple proxies and use all of them, we can use a limited number from the list, or we can have `proxychains change` the order randomly.

ADDING MORE PROXIES:

Add a few more of these proxies to our `/etc/proxychains.conf` file and then save this config file and try running the same command:

```
Kali> proxychains firefox www.google.com
```

We won't notice any difference, but our packet is now traveling through several proxies.

DYNAMIC CHAINING:

With multiple IPs in our config file, we can set up dynamic chaining, which runs our traffic through every proxy on our list and, if one of the proxies is down or not responding, automatically goes to the next

proxy in the list without throwing an error. If we didn't set up this up, a single failing proxy would break our request.

Go back into our */etc/proxychains.conf* file, and find the *dynamic_chain* line at line 10, and uncomment it. Also make sure to comment out the *strict_chain* line if it isn't already.

This will enable dynamic chaining our proxies, thus allowing for greater anonymity and trouble-free. Save the config file and feel free to try it out.

RANDOM CHAINING:

Our final proxy trick is random chaining option, where *proxychains* will randomly choose a set of IP addresses from our list and use them to create our proxy chain. This means that each time we use *proxychains*, the proxy will look different to the target, making it harder to track our traffic from its source. This option is also considered "dynamic" because if one of the proxies is down, it will skip to the next one.

Let's go back into the config file and comment out the lines *dynamic_chains* and *strict_chain* by adding #, then uncomment the *random_chain* line. We can only use one of these three options at a time, so comment out the other options before using the *proxychains*.

And also uncomment the *chain_len* and given it a value of 3, meaning *proxychains* will now use three proxies from our list in the config file, choosing them randomly and moving onto the next one if a proxy is down.

VIRTUAL PRIVATE NETWORK:

A VPN is used to connect to an intermediary internet device such as a router that sends our traffic to its ultimate destination tagged with the IP of the router.

Using a VPN can certainly enhance our security and privacy, but it's not a guarantee of anonymity. The internet device we connect to must record or log our IP to be able to properly send the data back to us, so anyone able to access these records can uncover info about us.

ENCRYPTED MAIL:

ProtonMail was founded by a group of young scientists at the CERN supercollider facility in Switzerland. The swiss have a long and storied history of protecting secrets, and ProtonMail's servers are based in the European Union, which has much stricter laws regarding the sharing of personal data than does the US. It is important to note that when exchanging email with non-ProtonMail users, there is the potential for some or all the email not to be encrypted.