

Staying Anonymous Online

Evan King and Zander Bolgar
Cornell Hacking Club

3/8/2016

Introduction

The code and content of the instruction manual can be found online on the **Piazza page**.

The content of this manual is for educational purposes only, Evan King, Zander Bolgar and the Cornell Cyber Security Club / Cornell Hacking Club in no way endorse or promote illegal actions or activities.

1 Tor

1.1 What is Tor?

Tor is free software that is both the easiest and most notable way to browse the web anonymously. This document will speak at a high level about what exactly Tor is, as well as how we can get it up and running on our own Kali machines.

Tor is short for The Onion Router and was initially a global network of servers developed with the U.S. Navy that enabled people to browse the Internet anonymously. It is now a non-profit organization which researches and develops online privacy tools. The Tor network disguises your identity by moving your traffic across different Tor servers, and encrypting that traffic so it isn't traced back to you. Anyone who tries would see traffic coming from random nodes on the Tor network, rather than your computer. Tor isn't only used by hackers it is also useful for anyone who wants to keep their Internet activities out of the hands of advertisers, ISPs, and web sites. That includes people getting around censorship restrictions in their country, police officers looking to hide their IP address, or anyone else who doesn't want their browsing habits linked to them.

Tor can also provide anonymity to websites and other servers. Servers configured to receive inbound connections only through Tor are called hidden services. Rather than revealing a server's IP address (and thus its network location), a hidden service is accessed through its onion address. It is on one of these hidden service sites that the infamous "Dark Web" exists.

Wikipedia provides a brief, high level overview as to how Tor manages to achieve its anonymity. "Onion routing is implemented by encryption in the application layer of a communication protocol stack, nested like the layers of an onion. Tor encrypts the data, including the destination IP address, multiple times and sends it through a virtual circuit comprising successive, randomly selected Tor relays. Each relay decrypts a layer of encryption to reveal only the next relay in the circuit in order to pass the remaining encrypted data on to it. The final relay decrypts the innermost layer of encryption and sends the original data to its destination without revealing, or even knowing, the source IP address."

1.2 Is Tor Broken?

Tor is incredibly easy and useful, but it is far from perfect. It is not safe to assume that just because you're using Tor that you're perfectly anonymous. With enough work, the government can (and has) figure out who Tor users are. Leaked Snowden documents revealed the following:
The NSA targeted Tor users who didn't keep their software up to date—using

custom-built tools with the codename “EgotisticalGiraffe.” Instead of attacking the Tor network directly, the NSA targeted older versions of the Firefox browser utilized by careless Tor users.

“We will never be able to de-anonymize all Tor users all the time,” the leaked NSA documents state. With malware and manual analysis, “a very small fraction” can be unmasked, the documents allege. However, the agency has never deanonymized a specifically targeted user.

You can read more about Tor’s vulnerabilities as well as how the government has been successful raiding it **here** and **here**. Despite these flaws, given that we are ethical hackers, Tor more that satisfies our needs.

1.3 Installation

First, it is important that we create a new user. This is because it is never wise under any circumstance to download anything from the Internet as root user. If a hacker is able to exploit your system, we want to guarantee that their privileges are limited and can, thus, cause less damage. In order to make this new user, we must run the following command:

```
root@kali:~# adduser <username>
```

Here, I will add the user ‘example’. It will ask you to give the user a password as well as fill in some personal information. Feel free to leave the personal info sections blank by hitting enter (as I do below).

```
root@kali:~# adduser example
Adding user ‘example’ ...
Adding new group ‘example’ (1002) ...
Adding new user ‘example’ (1001) with group ‘example’ ...
Creating home directory ‘/home/example’ ...
Copying files from ‘/etc/skel’ ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for example
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
```

Now we must log out and log back in as your newly created user. From here we will open another terminal and run the following command in order to install Tor. The -y tag just indicates that we will be agreeing with “yes” to all questions that the install prompts.

```
example@kali:~# sudo apt-get install tor -y
```

Open firefox/iceweasel and visit **torproject.org**. Navigate to the downloads page and download ‘Tor Browser for 64-bit GNU/Linux’

Tor Browser for 64-Bit GNU/Linux

Version 5.5.2 - Linux, Unix, BSD (64-Bit)

[Read the release announcements!](#)


Everything you need to safely browse the Internet. This package requires no installation. Just extract it and run.

[Learn more »](#)



Not Using GNU/Linux? Download for [Mac](#) or [Windows](#)

(sig) [What's This?](#)

English 

Looking For Something Else? [View All Downloads](#)

Navigate to your downloads folder and click extract. Extract the Tor download into the folder of your choice (feel free to just choose Desktop).

2 Proxychains

The worst thing that can happen to any hacker is being detected by a security admin, security technologies (IDS, firewall, etc.), or a forensic investigator.

Every time we send a packet to our intended target, that packet contains our IP address in the packet header. When we make a TCP connection, the target system will log our IP address as it logs all connections. If we set off any security alarms or alerts, our IP address will be logged. All of these events increase the possibility of detection.

A proxy could be explained as a gateway between the user computer and the destination web page. Normally while browsing through the website, your original IP is identified by the website, which could compromise your privacy. By the use of proxy chaining we bounce through a number of proxy servers and reach the destination. While using a proxy server you are not directly connected to the website, thus, making your IP much harder to trace.

Lets begin by looking in our proxychains configuration file.

```
example@kali:~# nano /etc/proxychains.conf
```

I recommend looking through this file. It is well documented and provides tons of useful options to help enhance your proxy chains. One thing I immediately recommend doing is changing how your ProxyList will be treated from the default static to dynamic.

This is so that if we add multiple IPs to our proxychains.conf we will run our traffic through every proxy on our list, and if one of the proxies is down or not responding, it will automatically go to the next proxy in the list without throwing an error.

To do this, uncomment out **dynamic_chain** and comment out **strict_chain** so that it looks just like mine does below.

```
proxychains.conf  VER 3.1
#
#           HTTP, SOCKS4, SOCKS5 tunneling proxifier with DNS.
#
# The option below identifies how the ProxyList is treated.
# only one option should be uncommented at time,
# otherwise the last appearing option will be accepted
#
dynamic_chain
#
# Dynamic — Each connection will be done via chained proxies
```

```
# all proxies chained in the order as they appear in the list
# at least one proxy must be online to play in chain
# (dead proxies are skipped)
# otherwise EINTR is returned to the app
#
#strict_chain
#
```

You can see at the top of proxychains.conf that there are three types of proxies. HTTP is classic proxy protocol and is typically considered less anonymous because some tend to leak your IP. SOCKS4 and SOCKS5 both belong to SOCKS protocol.

SOCKS4 only supports TCP application, while SOCKS5 supports TCP and UDP applications as well as IPv6 (which SOCKS4 does not). However, because of the fact that SOCKS5 also supports various authentication mechanisms and domain name resolution (DNS), which does not go with SOCKS4, the outgoing SOCKS proxy is normally SOCKS4 proxy. As a result, UDP applications are not supported normally. We will use SOCKS5 whenever we can.

The bottom of your proxychains.conf should look something like mine does below. By default proxychains utilize the Tor network. Tor, by default, runs on port 9050. 127.0.0.1 just resolves to our local IP also known as localhost. We should add socks5 directly below socks4, as I have done.

You can also feel free to add any number of free proxies. In order to do this, visit any number of free proxy websites such as www.socks-proxy.net. Below are some examples that I have added. Please note that these change frequently and my examples will most likely not work for you.

```
# add proxy here ...
# meanwhile
# defaults set to "tor"
socks4 127.0.0.1 9050
socks5 127.0.0.1 9050

# proxies I have added from www.socks-proxy.net
#socks5 201.208.60.120 9000
#socks5 168.167.132.150 64101
#socks5 178.161.173.23 47112
#socks5 46.4.88.203 9050
```

Now that we have configured our proxychains.conf we can start Tor and check to see that it is running by typing the following commands:

```
example@kali:~#service tor start
```



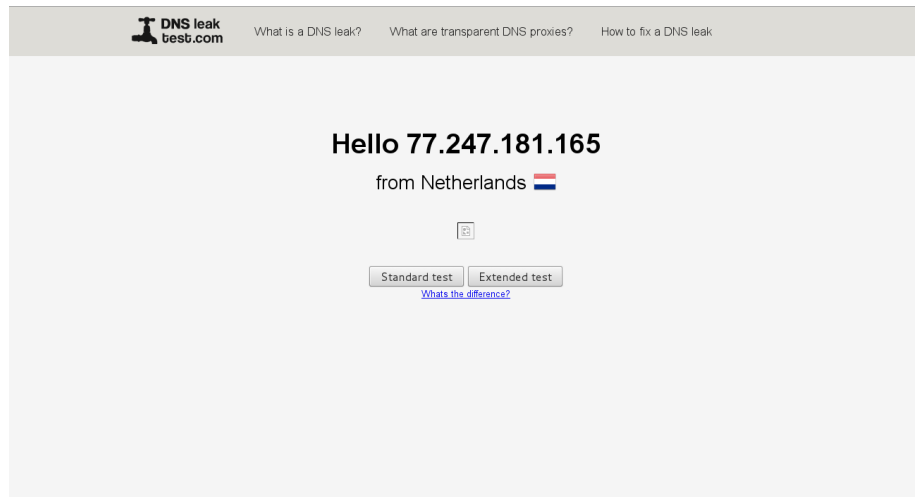
```
example@kali:~#service tor status
tor.service - LSB: Starts The Onion Router daemon processes
   Loaded: loaded (/etc/init.d/tor)
   Active: active (running) since Sun 2016-03-06 17:04:53 EST; 1h 5min ago
   Process: 2059 ExecStart=/etc/init.d/tor start (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/tor.service
           2071 /usr/bin/tor --defaults-torrc /usr/share/tor/tor-service-de...

Mar 06 17:04:53 kali tor[2059]: Starting tor daemon...done.
```

We're almost there! Now we just need to type 'proxychains' before any command that we would like to run with proxychains. We can check to see if it is working by opening up Firefox in this way.

```
example@kali:~#proxychains firefox google.com
```

We can Google 'DNS leaks' and click on the first result. This will show us where others will think that we are based on our last proxy in the chain. In my case, I am currently "located" in the Netherlands and given that I am actually sitting on my couch in Ithaca, we can tell it is working :)



3 MAC Address Changing

Quick note: A Media Access Controll (MAC) address has nothing to do with Apple or Macintosh computers.

3.1 What is a MAC Address

A MAC address is a physical address tied to the physical network interface (such as an Ethernet port or a Wi-Fi antenna). Computers can't actually directly talk to another computer if they do not share a direct connection (such as being connected through a switch or being part of the same Wi-Fi network). For a computer to talk with another computer, it addresses it with a MAC address. MAC address are unique, so they identify a particular device.

All routers have a MAC address - for example one eduroam router has the address `00:1a:1e:62:bf:41`. The first 6 hexadecimal digits (first 3 pairs) define the hardware manufacture - Cornell appears to use routers made by Aruba Networks. Another common prefix is `AC:BC:32` for Apple, Inc.

3.2 How are MAC addresses used

So far we have primarily focused on IP addresses. These IP addresses allow your computer to communicate with other computers both on the local network and far away across the world. There is actually a lot going on to talk to a computer far away as a computer can only actually talk to computers it is directly connected to. MAC addresses are used to make these close communications.

For example, it takes 17 of these jumps to reach `google.com` (or its IP `173.194.123.33`) in under 10ms. You can use the tool `tracert` `<ip|hostname>` to see the jumps taken to a particular remote computer.

Each computer listed is directly connected to the last in the list. Each computer sees where you are trying to send some information (maybe to `google.com`) and then sends it to the MAC address of a computer closer to your destination.

3.3 Why are MAC addresses important in regards to staying anonymous

To appear truly anonymous, there shouldn't be any record you existed. Proxies allow you to smudge this record from the view of the remote, but your local network operators still know exactly what device sent packets (even if the

destination and content is hidden). If it is known that someone you sent a large amount of data to a proxy, and the proxy did something bad, it could be inferred you originally sent something bad. One protection against this is to hide who it comes from too.

If I can be associated with a particular computer, which can be associated with a particular MAC address, which can be associated with particular network traffic, the network traffic can be associated with me. While this may sound hard to do at first, it is in practice extremely easy to figure out who sent packets on a network.

To not be associated with network traffic, you must break a link in the chain just described. Decoupling network traffic from a MAC address is not possible, so that link is not ideal to break. Buying and discarding computers could work, but that is extremely inconvenient and possibly costly. Instead making it impossible to associate a MAC address with a particular computer is the easiest attack vector - just claim your MAC address is some randomly generated address instead of the one assigned to your hardware.

3.4 Actually changing a MAC address

Note: Changing your MAC address on Cornell's network will likely disconnect you until your MAC address is reset to the one assigned to your computer and registered with CIT. It may even alert CIT and they could take action - I do not know. While changing your MAC address may work, we have not tried this (nor should you) - my guess is based solely on observing the device registration process.

Most networking devices are able to fake their own MAC address. There are numerous tools that handle the specifics of changing the address. Unlike most of the tools we talk about, this one can't actually be fully contained inside a Kali VM. If we fake the MAC address of our a VM, all network traffic still goes through the host machine. Instead, it is important the host machine fakes its own MAC address.

Each host OS has a different method for changing MAC addresses. I have found the easier way to change the MAC address on most OSes is to use a Python script called **SpoofMAC**. By running `spoof-mac list --wifi` you can see the names of Wi-Fi interfaces on your computer. On Mac, the Wi-Fi interface is generally `en0`. A random MAC address can be set by running `spoof-mac randomize <interface>` with administrator privileges. The interface's MAC address can be reset to the assigned one by running `spoof-mac reset en0` with administrator privileges. An example output is below:

```
$ spoof-mac list --wifi
```

```
- “Wi-Fi” on device “en0” with MAC address AC:BC:32:00:00:00
# spoof-mac randomize en0
$ spoof-mac list --wifi
- “Wi-Fi” on device “en0” with MAC address AC:BC:32:00:00:00
  currently set to 08:00:27:00:00:00
# spoof-mac reset en0
$ spoof-mac list --wifi
- “Wi-Fi” on device “en0” with MAC address AC:BC:32:00:00:00
```

Can you figure what company made my laptop?