Brooklyn Dressel
Assignment 7

Abstract

This assignment explores the use of the Tor browser over an onion network. We capture packets of the generated network traffic only to discover very little regarding the content of the data being sent back and forth.

Introduction

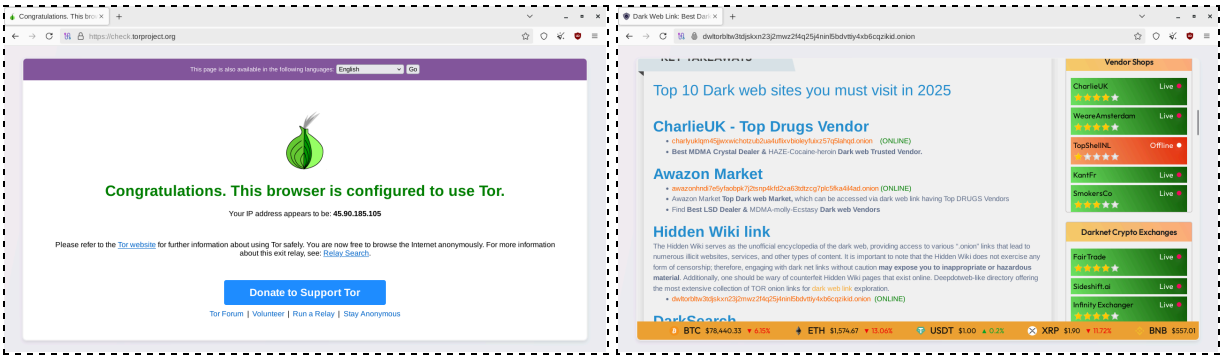
This assignment uses the Tails virtual machine iso, run inside of VirtualBox. Tail provides access to the Tor browser, the unsafe browser, GPG encryption tools, as well as OnionShare. WireShark, run inside the Kali virtual machine, allows us to analyze the captured packets.

Summary of Results

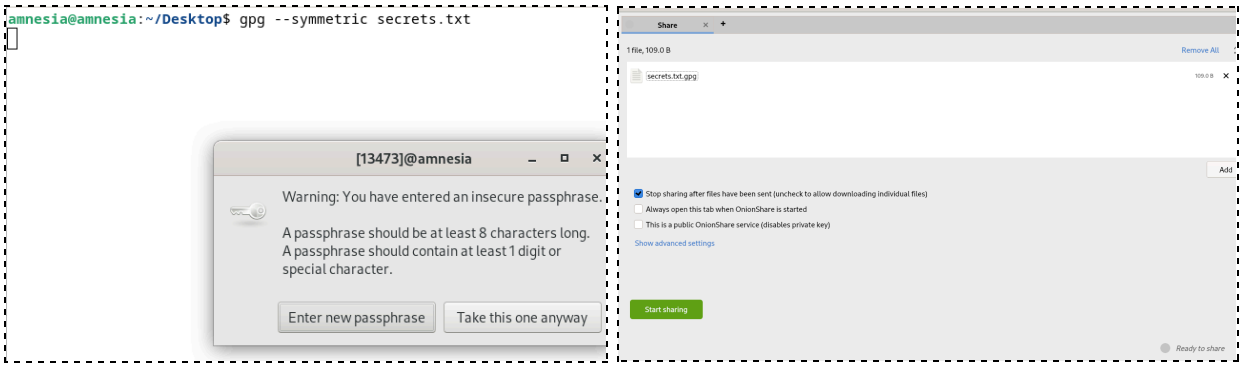
The initial step of this is to download and install Tails. Using the link `https://tails.net/install/download-iso/index.en.html` I installed the iso file. I used VirtualBox to run Tails from the iso image, after giving it more memory and CPUs I booted it up. Once inside I edited the settings to allow for a root password and set it, this enabled me to run `command 1`, shown below. This created a packet capture of all traffic in and out of the Tail's network, and stored it in a file on the desktop.

```
amnesia@amnesia:~$ tcpdump -i any -s 65535 -w /home/amnesia/Desktop/packets.pcap
tcpdump: any: You don't have permission to perform this capture on that device
(socket: Operation not permitted)
amnesia@amnesia:~$ sudo tcpdump -i any -s 65535 -w /home/amnesia/Desktop/packets
.pcap
[sudo] password for amnesia:
tcpdump: data link type LINUX_SLL2
tcpdump: listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot leng
th 65535 bytes
```

Next I started to generate a variety of traffic, starting off with traffic completely on the onion network. So I first ensured that the browser was properly configured to use Tor by going to the site `https://check.torproject.org/`. Then, I looked up some onion likes using the DuckDuckGo search, and randomly selected a few. Next I used the unsafe browser to generate traffic from non-onion, but still https sites. It is notable that loading times in the Tor browser took significantly longer than in the unsafe browser.



The final type of traffic I generated was to upload a file to OnionShare. In order to do this, I created a text file, filled it with a secret message, named the file `secrets.txt`, appropriately. Using `command 2` I encrypted the file with symmetric encryption. Then proceed to upload the symmetrically encrypted `secrets.txt.gpg` to OnionShare.



Now that all traffic had been generated, I used `ctrl+C` to stop the packet capture. Since the command used to start the packet capture was run using `sudo`, I next had to change the permissions for `packets.pcap` using `commands 3 and 4`. These commands change the ownership and permissions of the file, respectively. In order to move this, and other files, I ran `command 5` to allow me to upload my files to the Tor browser and then upload them into my Google Drive.

```
amnesia@amnesia:~$ sudo chown amnesia:amnesia /home/amnesia/Desktop/packets.pcap
[sudo] password for amnesia:
amnesia@amnesia:~$ sudo chmod 777 /home/amnesia/Desktop/packets.pcap
[sudo] password for amnesia:
amnesia@amnesia:~$
```

With the files in Google Drive I was able to close Tails and access the packet capture and analyze them inside WireShark. From these results we can see that all data is encrypted, and therefore unreadable. Some of the data that we can see from this are the source and destination IP addresses, the Tails machine being 10.0.2.15, and the outside IP being 46.228.199.128. We can also see that no outgoing packets are larger than 596, while the incoming packets range from 66 to 1442. We can also see that only ports 9001 and 47986 are used, and only TCP and TLSv1.2 traffic is sent. It's also interesting to note that there are quite a few packets with the description of "application data."

Conclusion

In conclusion the Tor Onion Router is a fully encrypted browser that uses a series of encrypted tunnels and VPNs (the onion) to facilitate communication between client and server. Because Tor uses a series of encryption and proxies, it helps to hide traces of the users by hiding meta data that is leaked, providing both anonymity as well as data confidentiality. This server works best if websites are on the Tor network as well (`.onion` links), otherwise the full connection between the client and server does not take place on the Tor site, meaning there is a portion of the connection that is not as secure. Because of how secure and anonymous Tor is, it is filled with lots of illegal sites, many of which could be riddled with malicious software. This can come in the form of viruses or worms, or website hooks.

The Tail virtual machine helps to improve security by including tools such as the Tor browser, OnionShare, and Thunderbird. These applications facilitate better encrypted internet access, file transfer and sharing, as well as emailing. Additionally, there are tools for scrubbing metadata which can help to aid the anonymity of uses by erasing any traces of them on files. Tails adds extra security when using Tor as it applies non-persistence, meaning that in the event any malware does make it's way onto the device (through the risky sites on Tor) that once Tails is closed all data is reset, and the malicious software is wiped from the site. Additionally, Tails has many restrictions for the user that prevent any malicious software from doing too much damage or elevating the attackers status to root or the such. Finally, in Tails there is no root password (unless otherwise set) ensuring that the attacker can never gain root access at all.