CS790 Assignment 2

The Evil Eye
Report

Kavya Gupta (22B1053)
Department of Computer Science,
Indian Institute of Technology Bombay

Approach

I observed that the Vanilla TOR uses (18) specific cipher suites (See Figure 1). So, for any TLS Client Hello, I get the cipher suite list and match it with these 18. If they match exactly, I deem the destination IP address as a TOR guard IP (and add it to a list). After this, any TCP packet, with (source/destination) IP same as any present in the list, is marked as a TOR packet.

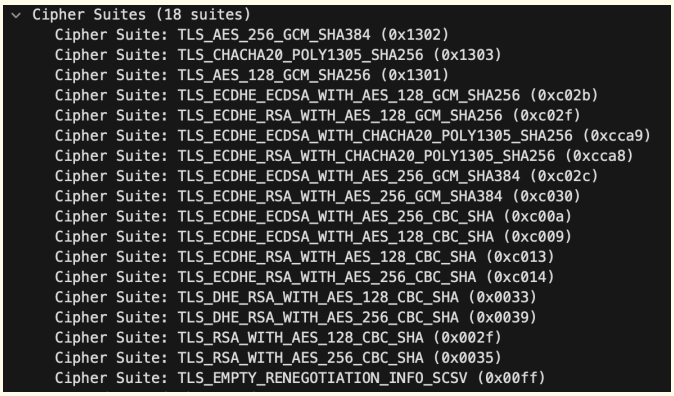

Figure 1: Vanilla TOR Cipher Suites

Note: I have made an assumption that a TOR IP remains a TOR IP (atleast for the time the Wireshark capture is recorded).

Implementation

- Programmed in Lua.
- Wireshark has this class called `Dissector` which is used to decode (dissect) packets. They extract the fields, call upper layer dissectors (if needed) and present the extracted info as part of a 'tree' in the packet info. So I had to create my own dissector function that will extract necessary information and deem a packet as TOR as told in the above approach.

- To do that, I had to first create a `Proto` (protocol) object and then wrote the dissector. In this function, I call the dissector of TCP first (which was acquired using `DissectorTable.get()`). This will chop down the packet and make fields (of TCP and TLS (if present)) available.
- Then I check if the packet is a TLS Client Hello packet by checking if `tls.handshake.type` is 1. I acquired its value using the class `Field`. Using `Field.new(<fieldname>)`, I can get the value of any field in the packet. Value of `tls.handshake.ciphersuites` field gives the list of cipher suites used in the packet and I check if it matches with the TOR cipher suites. If it does, I add the destination IP address (got using `Field.new("ip.dst")`) to a list (of TOR IPs).
- Then in the function, I extract the source IP and destionation IP addresses and check if any of them is present in the list. If yes, I mark the packet as a TOR packet (to show this I just append `" (Tor!)"` to the Protocol column, see figure 2).
- Finally this dissector has to be added to the `DissectorTable`. Using `.add()`, I coded that if the protocol field of the IP header is 6 (which is TCP), then call my dissector function.

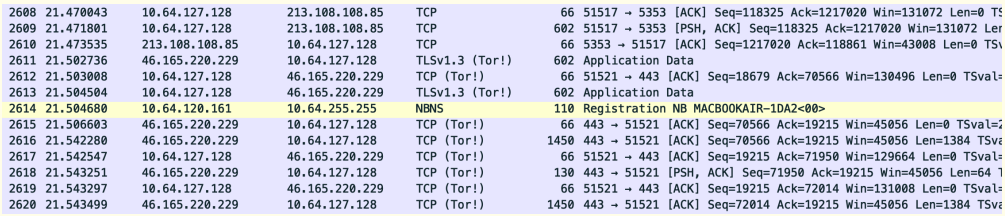

Figure 2: Tor display in Wireshark

Files Submitted

- `tor_detector.lua`
- `sample.pcap`
- `report.pdf`
- `README.txt`

References

- Little help from ChatGPT, Google.
- `https://www.wireshark.org/docs/wsdg_html_chunked/wsluarm_modules.html`: Wireshark Lua API Reference Manual