Network Security Report

**Task 1: Honeypot Traffic Analysis**

**Sub-task 1**

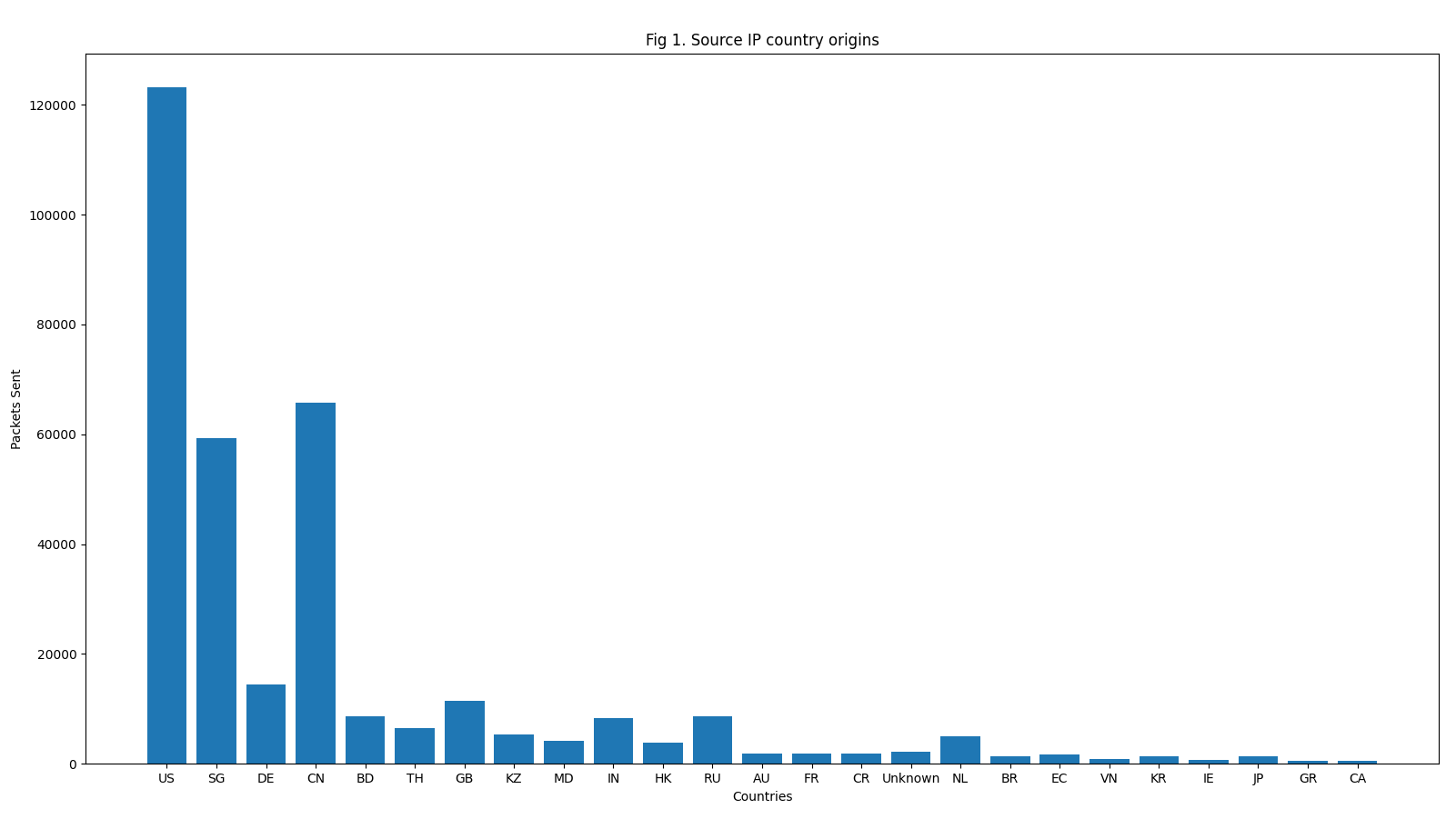


Fig 1 was created by the Pyplot module from the matplotlib python library which allowed me to create a bar graph. This data was retrieved first by looking at source IP addresses via wireshark statistics. Then, I scraped only the top 100 source IPs based on packets sent (ignoring the honeypot IP) and used a service named “ipinfo” which gave me the origin countries of the source IPs. This is all plotted onto the Fig 1 graph. This is all done using a python script.

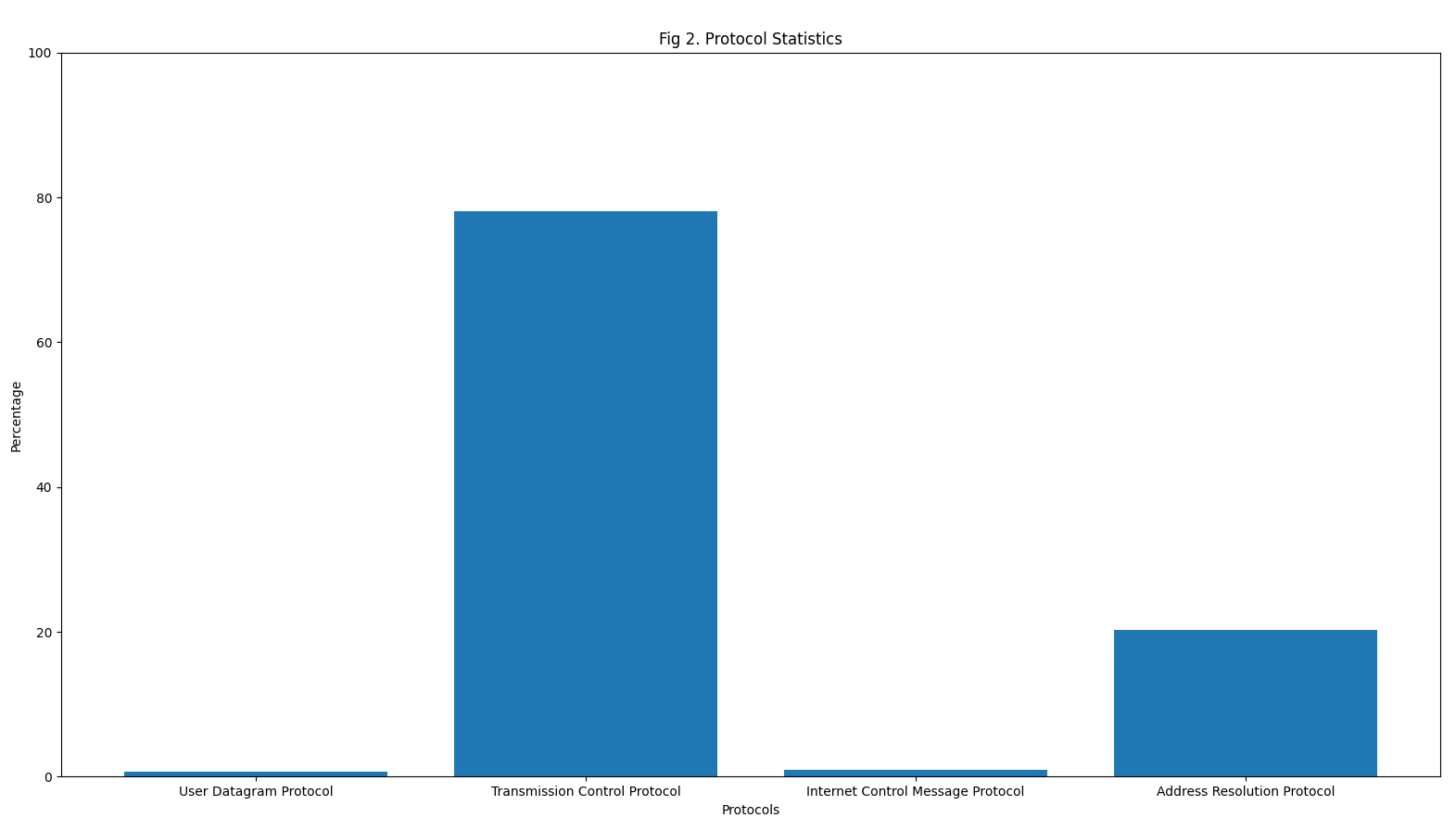


Fig 2 was also created by the same moule stated for Fig 1. This time, I looked at protocol hierarchy via wireshark statistics. Since there was no automatic hierarchy when saving as a csv file, I manually remove unnecessary data to get the relevant data in Fig 2. I simply ran a python script to obtain the bar graph above.

Table 1. Table of services

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Protocol | Percent Packets | Packets | Percent Bytes | Bytes |
| Network Time Protocol | 0.274227 | 2597 | 0.050808 | 124656 |
| Dynamic Host Configuration Protocol/UDP | 0.347615 | 3292 | 0.425332 | 1043530 |
| Domain Name System/UDP | 0.114464 | 1084 | 0.033482 | 82146 |
| Transport Layer Security | 0.127346 | 1206 | 0.368388 | 903821 |
| TPKT - ISO on TCP - RFC1006 | 0.041182 | 390 | 0.007129 | 17490 |
| SSH Protocol | 29.84972 | 282684 | 29.60813 | 72642021 |
| Session Initiation Protocol | 0.000845 | 8 | 0.000727 | 1784 |
| Remote Procedure Call | 0.001267 | 12 | 0.000215 | 528 |
| Remote Desktop Protocol | 0.000634 | 6 | 0.000165 | 406 |
| Modbus/TCP | 0.040442 | 383 | 0.00127 | 3115 |
| LANforge Traffic Generator | 0.001795 | 17 | 0.001081 | 2652 |
| IEC 60870-5-104 | 0.008448 | 80 | 0.001156 | 2836 |
| Hypertext Transfer Protocol | 0.083736 | 793 | 39.51675 | 96952320 |
| General Inter-ORB Protocol | 0.000739 | 7 | 0.000137 | 336 |
| EtherNet/IP (Industrial Protocol) | 0.002534 | 24 | 0.000264 | 648 |
| Distributed Network Protocol 3.0 | 0.000739 | 7 | 0.002774 | 6806 |
| Data/TCP | 0.20707 | 1961 | 0.022351 | 54836 |
| Internet Control Message Protocol | 0.915288 | 8668 | 0.101964 | 250163 |
| Dynamic Host Configuration Protocol/ICMP | 0.000211 | 2 | 0.000245 | 600 |
| Domain Name System/ICMP | 0.002851 | 27 | 0.000451 | 1106 |
| Data/ICMP | 0.003168 | 30 | 0.000418 | 1026 |

Method to obtain data same as from Fig 2. However, this time I simply placed the data on services in a table as making a graph would have not have made visualisation any better.

**Summary of findings**

The honeypot data was taken from the Singapore honeypot and was from the month of November 2017. From Fig 1, we can see that majority of the packets sent to the honeypot originated from 3 countries: United States, Singapore and China. There were many other countries as well but they do not generate significant requests to the honeypot. From Fig 2, TCP was the most used protocol and by a large margin compared to other IP/Transport layer protocols. From Table 1, a great majority of packets was from the SSH service and it is by a large margin compared to other services.

**Sub-task 2**

Table 2. Packet count based on packet size compared between USA and SG

|  |  |  |
| --- | --- | --- |
| Packet Size | Singapore Count | USA Count |
| 0-19 | 0 | 0 |
| 20-39 | 0 | 0 |
| 40-79 | 575082 | 574352 |
| 80-159 | 172126 | 185757 |
| 160-319 | 43430 | 69265 |
| 320-639 | 26325 | 30251 |

Number of source IPs that appear in both: 2229

**Summary of findings**

Both sets of honeypot traffic are from the same timeframe which is the month of November 2017. We are comparing between the honeypots located in USA and Singapore. From table 2, there are very similar number of packets in the packet size ranges given in wireshark which shows a strong correlation between the two honeypots. The number of source IPs that appear in both also come up to 2229. This is done by checking whether one source IP exists in the other honeypot’s source IPs. Since both honeypots received packets from around 4-5 thousand source IPs, having nearly 50% appearing in both Singapore and USA honeypots is interesting given how far apart Singapore and the US is in terms of geolocation and time zones.

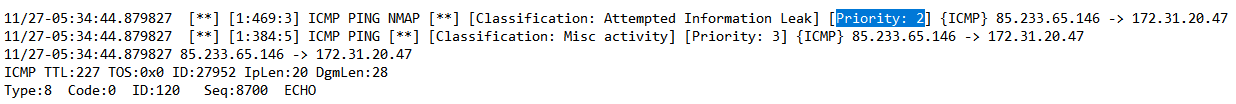
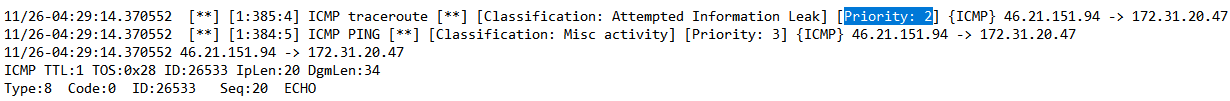
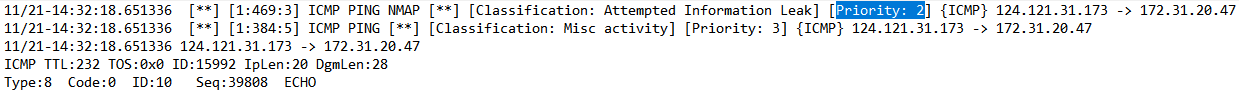
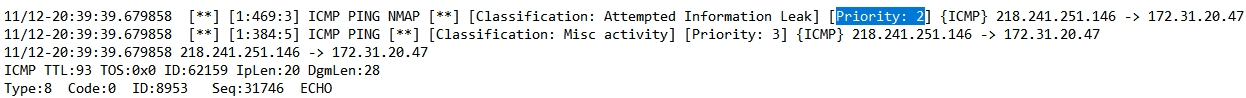
**Sub-task 3**

After installing snort, I ran this command:

“sudo snort -v -c /etc/snort/snort.conf -A console -r ../../Downloads/**<Pcap File>** > pcapIDS.txt”

in order to obtain some information on whether attack attempts occurred. This pcap file used was simply the merged pcap file for November in the Singapore honeypot and used a community ruleset to find these attempted attacks. From there I searched for the keyword “Priority” so see whether there was anything worth noting. Here are some of my results found:

Fig 3. Attempted Information Leak logs from Snort

These are just a few of the logs that were deemed to be attempted information leaks from the snort done on the pcap file. According the the Mitre framework, this should lie under Reconnaissance where the other party is trying to know more about the honeypot, in this case, running things like “Nmap” possible to learn about the honeypot’s port availability and type of device the honeypot is, which can then be used by malicious actors to attack if they find a vulnerability.

**Sub-task 4**

Analytics of honeypot traffic can be very useful in configuring cybersecurity tools. One such way is that It provides large amounts of data from the high traffic such that trends can be made to determine how malicious actors might work. These trends allow current cybersecurity tools to move from a reactionary approach where the tool might detect a possible malicious attack and the people must mitigate losses, to a more proactive approach where using these trends, the tools can already detect that this certain trend of traffic is malicious and block the traffic from performing any malicious deeds. Another way that the honeypot data can help is that with the large amounts of data and newly growing machine learning scene, cybersecurity tools can become automated to detect and prevent malicious traffic by learning from previous malicious attacks. The tools can then prevent the malicious traffic in the future as they are able to detect similar activity from the previous malicious attacks on the honeypot.

This allows current cybersecurity tools to become much stronger as it becomes more proactive in detecting and preventing malicious attacks, and that the large volume of traffic to these honeypots serve as data for data analysts to use machine learning to make detection of similar events in the future much easier.

**Task 2: APT Network Trace Analysis**

**Sub-task 1**

With the filter “ip.addr != 192.168.0.0/16” to remove local machine IPs, we can get the IP addresses that have communicated with the office network. Some information about the IP addresses are placed here. They are obtained by running the IP address through the whois command and obtaining their ASN name.

Table 3. IP Addresses communicated with and their respective ASN Names

|  |  |
| --- | --- |
| IP Address | ASN Name |
| 8.8.8.8 | GOOGLE, US |
| 74.6.143.26 | YAHOO-BF1, US |
| 74.6.143.25 | YAHOO-BF1, US |
| 74.208.236.166 | IONOS-AS This is the joint network for IONOS, Fasthosts, Arsys, 1&1 Mail and Media and 1&1 Telecom. Formerly known as 1&1 Internet SE., DE |
| 74.125.68.99 | GOOGLE, US |
| 74.125.68.95 | GOOGLE, US |
| 74.125.68.147 | GOOGLE, US |
| 74.125.68.139 | GOOGLE, US |
| 74.125.68.106 | GOOGLE, US |
| 74.125.68.105 | GOOGLE, US |
| 74.125.68.103 | GOOGLE, US |
| 74.125.24.94 | GOOGLE, US |
| 74.125.24.139 | GOOGLE, US |
| 74.125.24.138 | GOOGLE, US |
| 74.125.24.113 | GOOGLE, US |
| 74.125.24.102 | GOOGLE, US |
| 74.125.24.101 | GOOGLE, US |
| 74.125.24.100 | GOOGLE, US |
| 74.125.200.94 | GOOGLE, US |
| 74.125.200.139 | GOOGLE, US |
| 74.125.200.101 | GOOGLE, US |
| 74.125.200.100 | GOOGLE, US |
| 74.125.130.94 | GOOGLE, US |
| 74.125.130.102 | GOOGLE, US |
| 69.147.80.15 | YAHOO-SWB, US |
| 54.217.10.153 | AMAZON-02, US |
| 52.109.52.148 | MICROSOFT-CORP-MSN-AS-BLOCK, US |
| 40.79.150.120 | MICROSOFT-CORP-MSN-AS-BLOCK, US |
| 4.246.174.31 | MICROSOFT-CORP-MSN-AS-BLOCK, US |
| 34.160.122.198 | GOOGLE, US |
| 23.72.44.106 | AKAMAI-AS, US |
| 23.64.122.82 | AKAMAI-ASN1, NL |
| 23.15.147.56 | AKAMAI-AS, US |
| 216.239.32.55 | GOOGLE, US |
| 216.239.32.29 | GOOGLE, US |
| 212.166.100.13 | T-SYSTEMS-AT Rennweg 97-99, AT |
| 204.79.197.219 | MICROSOFT-CORP-MSN-AS-BLOCK, US |
| 204.79.197.203 | MICROSOFT-CORP-MSN-AS-BLOCK, US |
| 202.165.107.57 | YAHOO-SG3 internet content provider, SG |
| 202.165.107.50 | YAHOO-SG3 internet content provider, SG |
| 20.50.73.9 | MICROSOFT-CORP-MSN-AS-BLOCK, US |
| 20.205.243.166 | MICROSOFT-CORP-MSN-AS-BLOCK, US |
| 20.103.253.93 | MICROSOFT-CORP-MSN-AS-BLOCK, US |
| 185.199.109.133 | FASTLY, US |
| 185.125.190.28 | CANONICAL-AS, GB |
| 185.125.188.59 | CANONICAL-AS, GB |
| 185.125.188.58 | CANONICAL-AS, GB |
| 185.125.188.55 | CANONICAL-AS, GB |
| 172.67.7.19 | CLOUDFLARENET, US |
| 172.67.69.99 | CLOUDFLARENET, US |
| 172.253.118.94 | GOOGLE, US |
| 172.253.118.84 | GOOGLE, US |
| 172.253.118.106 | GOOGLE, US |
| 172.253.118.103 | GOOGLE, US |
| 172.253.118.101 | GOOGLE, US |
| 172.217.194.94 | GOOGLE, US |
| 172.217.194.83 | GOOGLE, US |
| 172.217.194.17 | GOOGLE, US |
| 172.217.194.138 | GOOGLE, US |
| 172.217.194.105 | GOOGLE, US |
| 172.217.194.103 | GOOGLE, US |
| 172.217.194.102 | GOOGLE, US |
| 172.217.194.100 | GOOGLE, US |
| 142.251.12.99 | GOOGLE, US |
| 142.251.12.94 | GOOGLE, US |
| 142.251.12.84 | GOOGLE, US |
| 142.251.12.147 | GOOGLE, US |
| 142.251.12.139 | GOOGLE, US |
| 142.251.12.113 | GOOGLE, US |
| 142.251.12.106 | GOOGLE, US |
| 142.251.12.105 | GOOGLE, US |
| 142.251.12.104 | GOOGLE, US |
| 142.251.12.103 | GOOGLE, US |
| 142.251.12.102 | GOOGLE, US |
| 142.251.12.101 | GOOGLE, US |
| 142.251.12.100 | GOOGLE, US |
| 142.251.10.94 | GOOGLE, US |
| 142.251.10.19 | GOOGLE, US |
| 142.251.10.139 | GOOGLE, US |
| 142.251.10.138 | GOOGLE, US |
| 142.251.10.113 | GOOGLE, US |
| 142.251.10.108 | GOOGLE, US |
| 142.251.10.102 | GOOGLE, US |
| 142.251.10.101 | GOOGLE, US |
| 142.251.10.100 | GOOGLE, US |
| 142.250.4.99 | GOOGLE, US |
| 142.250.4.94 | GOOGLE, US |
| 142.250.4.139 | GOOGLE, US |
| 142.250.4.138 | GOOGLE, US |
| 142.250.4.113 | GOOGLE, US |
| 142.250.4.109 | GOOGLE, US |
| 142.250.4.104 | GOOGLE, US |
| 142.250.4.102 | GOOGLE, US |
| 142.250.4.101 | GOOGLE, US |
| 142.250.4.100 | GOOGLE, US |
| 142.250.181.35 | GOOGLE, US |
| 13.107.4.50 | MICROSOFT-CORP-MSN-AS-BLOCK, US |
| 119.161.10.12 | YAHOO-HK2-AP internet content provider, HK |
| 119.161.10.11 | YAHOO-HK2-AP internet content provider, HK |
| 106.10.248.157 | YAHOO-SG3 internet content provider, SG |
| 106.10.236.40 | YAHOO-SG3 internet content provider, SG |
| 106.10.236.37 | YAHOO-SG3 internet content provider, SG |
| 106.10.236.141 | YAHOO-SG3 internet content provider, SG |
| 106.10.218.137 | YAHOO-SG3 internet content provider, SG |
| 104.69.44.74 | AKAMAI-AS, US |
| 104.69.42.238 | AKAMAI-AS, US |
| 104.69.38.163 | AKAMAI-AS, US |
| 104.26.9.198 | CLOUDFLARENET, US |
| 104.26.3.27 | CLOUDFLARENET, US |
| 104.26.2.27 | CLOUDFLARENET, US |
| 104.22.37.221 | CLOUDFLARENET, US |
| 104.22.36.221 | CLOUDFLARENET, US |

**Sub-task 2**

192.168.56.101 is the office machine that has been compromised by a malware. Why I believe so is the other 3 machines, when looking at their network activities, it contained a lot of normal activities such as browsing for football news, looking at websites like kruumui.com which is for learning the Thai language, using GitHub and even start.gg which is a website for looking for events relating to games. The 3 machines showed this kind of activity however for 192.168.56.101, it did not show much human activity and only repeatedly browsed the same few hosts while initiating connection consistently every minute without fail which makes me believe that this machine is infected with malware.

**Sub-task 3**

While researching on APTs and how what they do, one article relayed the fact that they often communicate with the command and control (C2) centres through TCP port 443 which sends encrypted data. Using this info, I filtered the packets as such:

tcp.dstport == 443 && ip.addr == 192.168.56.101

to see whether there is anything in the network trace that I can work with.

From there, I found something rather interesting:

Graphical user interface

Description automatically generated with medium confidenceText

Description automatically generated with low confidenceGraphical user interface, text

Description automatically generated with medium confidence

Every few seconds after the beginning of the minute, the machine will initiate a connection with this IP address which will always end in a RESET soon after. This happens every minute in the given network trace. This is always followed by connections to other servers but similar to the above pictures, they are always the same IP addresses. This leads me to believe that the machine is either relaying to its C2 centre that it is alive or that it is slowly exfiltrating data in small quantities to the C2 centre which is what the malware is doing after is has compromised the machine with IP address 192.168.56.101