Graham Dungan
February 9, 2025
UIN: 332001764

CSCE 413: Software Security
POC 9: Honeypots

Create a Socket Application

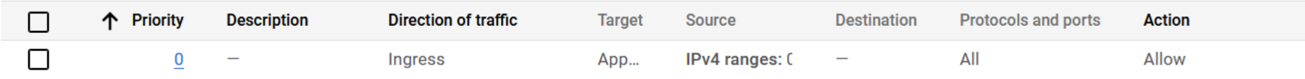
For this assignment, I created a program named `honeypot.py`. This program initializes a set number of sockets on different ports. The application opens popular TCP and UDP ports, many of which are collected from this website of frequently used TCP/UDP ports. The program can be run by;

1. Entering the PoC9 directory.
   `cd PoC9`
2. Running the `honeypot.py` Python script. `python3 ./honeypot.py`

The program will log the opened ports and any connection attempts in the console and `honeypot.log`. The program works by instantiating a thread for each listed port. TCP ports and UDP ports required threads running different functions: `tcp_listener` and `udp_listener`, respectively. The UDP threads simply listen for any requests, meanwhile, TCP ports accept requests and send ”Service Unavailable” responses.

Instantiate a Free Cloud VM

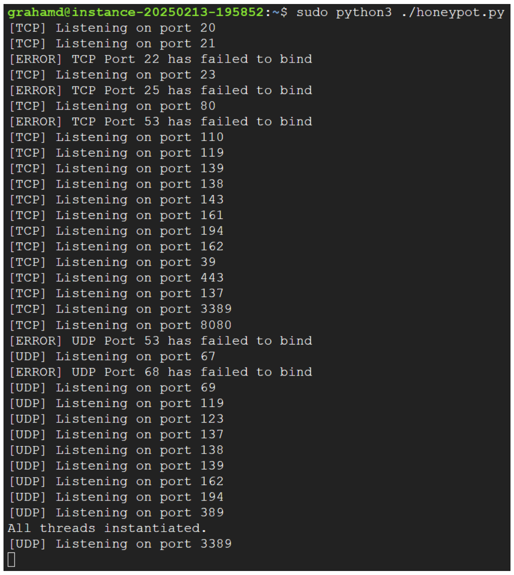
I chose to host `honeypot.py` on a simple Google Cloud VM. I edited firewall rules to allow all ingress traffic and disabled any protections against known malicious IP’s.



□	↑ Priority	Description	Direction of traffic	Target	Source	Destination	Protocols and ports	Action
□	0	—	Ingress	App...	IPv4 ranges: (	—	All	Allow

Run the App

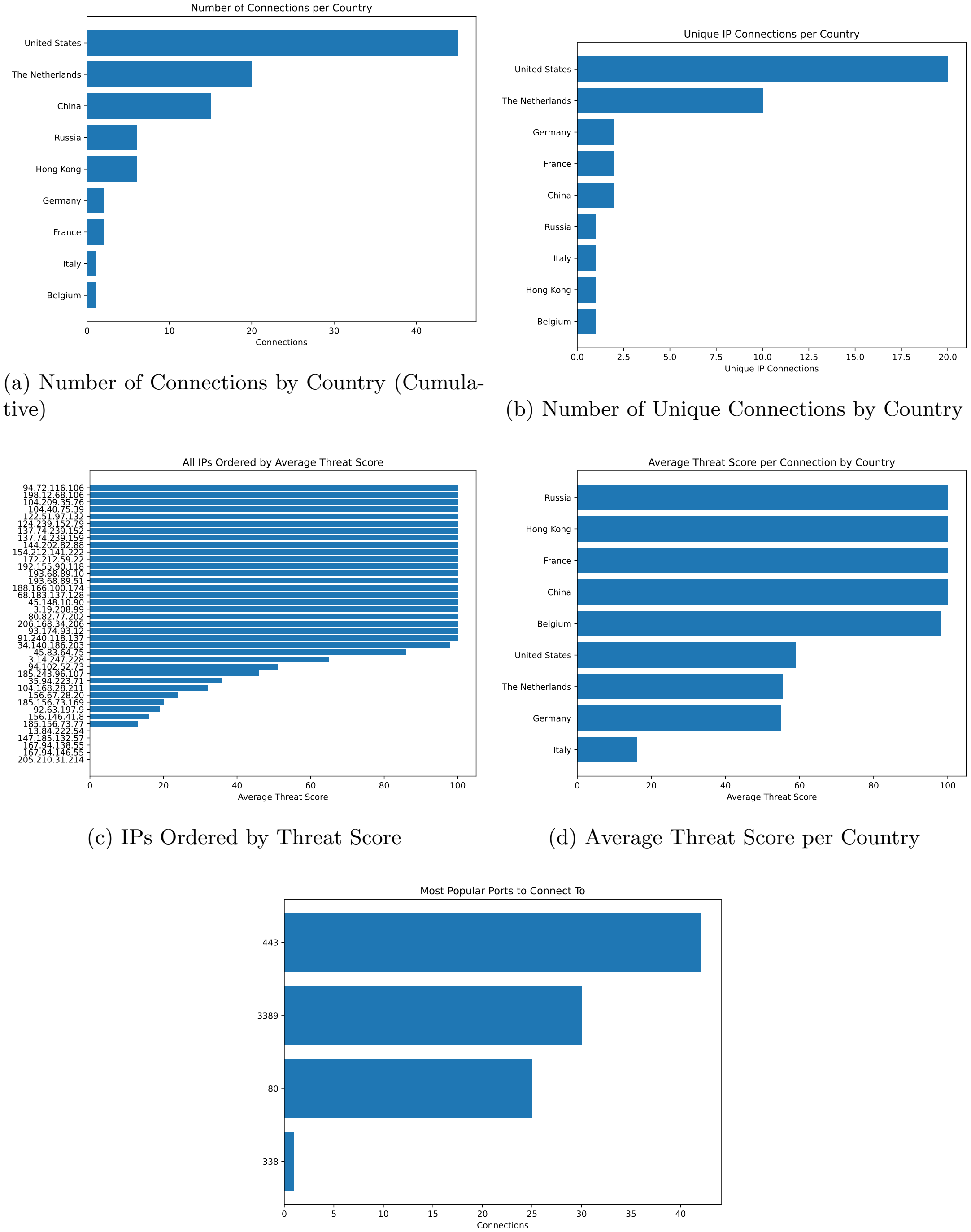
I ran the honeypot for three hours instead of one. The choice to run the honeypot for three hours was a consequence of the fact that I forgot to turn it off, however, this mistake resulted in a larger data sample. The app was run in a simple tmux session and scanned over most listed ports.



```
grahamd@instance-20250213-195852:~$ sudo python3 ./honeypot.py
[TCP] Listening on port 20
[TCP] Listening on port 21
[ERROR] TCP Port 22 has failed to bind
[TCP] Listening on port 23
[ERROR] TCP Port 25 has failed to bind
[TCP] Listening on port 80
[ERROR] TCP Port 53 has failed to bind
[TCP] Listening on port 110
[TCP] Listening on port 119
[TCP] Listening on port 139
[TCP] Listening on port 138
[TCP] Listening on port 143
[TCP] Listening on port 161
[TCP] Listening on port 194
[TCP] Listening on port 162
[TCP] Listening on port 39
[TCP] Listening on port 443
[TCP] Listening on port 137
[TCP] Listening on port 3389
[TCP] Listening on port 8080
[ERROR] UDP Port 53 has failed to bind
[UDP] Listening on port 67
[ERROR] UDP Port 68 has failed to bind
[UDP] Listening on port 69
[UDP] Listening on port 119
[UDP] Listening on port 123
[UDP] Listening on port 137
[UDP] Listening on port 138
[UDP] Listening on port 139
[UDP] Listening on port 162
[UDP] Listening on port 194
[UDP] Listening on port 389
All threads instantiated.
[UDP] Listening on port 3389
```

Results

After collecting the resulting `honetpot.log` file, I decided to do a deeper analysis of the data. I created a Python script that used IP-Api and AbuseDB to find the location and threat percentage of each IP, respectively. After compiling this dataset, I made several graphs to make inferences on the type of connections. This dataset can be found in `honeypot_data.csv`.



(a) Number of Connections by Country (Cumulative)

(b) Number of Unique Connections by Country

(c) IPs Ordered by Threat Score

(d) Average Threat Score per Country

(e) Most Popular Ports

Figure 1: A collection of graphs detailing honeypot connections.

It is seen that a majority of the connections (both cumulative and unique) came from the United States. All of Russia's, Hong Kong's, France's, and China's connections were given an abuse confidence score of 100%. The main ports accessed were 443, 3389, 80, and 338. Ports 443 and 80 are standard for HTTP connections, and port 3389 is standard for remote desktop connections. Since most of the connections had a high abuse confidence rating, as evident by (1c), it would make sense that many of the ports accessed were ones typically vulnerable to attacks.

Selecting the notably frequent IP `122.51.97.132`, it is seen that the IP is owned by Shenzhen Tencent Computer Systems Company Limited in Shanghai, China, and has attempted to connect to ports 80 and 443 multiple times. On AbuseDB, IP `122.51.97.132` has an abuse confidence level of 100% and is frequently listed with the categories Port Scan, Web App Attack, and Brute-Force. A simple google search for the IP reveals that it is on many block lists.

Another somewhat frequent IP is `167.94.138.55`. This IP is related to a service called Censys which performs network mapping to "discover, monitor, and analyze devices that are accessible from the Internet". The IP is whitelisted on AbuseDB as being safe, and stands out among the other more malicious IP connections.

Overall, it is seen that a majority of attempted connections, if not for educational or internet-mapping purposes, are notoriously malicious IP's attempting to access sensitive ports on the machine.