Assignment 1: T-Pot: A Multi-Honeypot Platform

Abel Melinte
B00137882

Department of Informatics,
School of Informatics and Engineering,
Technological University Dublin,
Dublin 15

[Page Count: **Word Count:]**

Network Security Analytics
Digital Forensics and Cyber Security
09/11/2023

Department of Informatics,
School of Informatics and Engineering,
Technological University Dublin,
Dublin 15

Declaration On Plagiarism

I declare that the work we are submitting for assessment by the Institute examiner(s) is entirely our own work, except where the author or source has been duly referenced and attributed.

We confirm that this material has not been previously submitted for a degree or any other qualification at TUD or any at other institution.

We further confirm that we have read and understood the Institute policy on plagiarism in assignments and examinations (3AS08.doc) and that we are not, as far as we are aware, in breach of any of these regulations.

Names: Abel Melinte Student IDs: B00137882

Signed: Abel Melinte Date: 09/11/2023

Course: Digital Forensics and Cyber Security Module: Network Security Analytics

Abstract

A honeypot is a bait or fake production machine / virtual machine that is configured with vulnerable services which are placed outside or inside of the network.

In computer security terms, honeypots are basically a bait for hackers. It's a baited system that lurs all sort of attackers from around the world and make them perform their attacks against it. Honeypots are quite helpful in the computer security space as for some companies who decide to create honeypots, they can set them up and be able to protect their main machines as attackers will focus on their pots instead of where the sensitive data is being stored.

Table of Contents

1. Introduction

Honeypot is a security mechanism which is setup to detect any suspicious activities coming from attackers located all around the world, it is a way of gathering all sort of tactics which attackers can use, this can be highly beneficial as for a company, they can patch all these methods and be able to avoid all these attacks before the hackers reaches them.

In this assignment the task that was given out was to research and deploy a T-Pot within a cloud and leave it running and gather all attacker's data for an entire month.

1.1 Droplet (T-Pot) Setups

To launch the T-Pots, a cloud had to be chosen to go with, the cloud which was selected in throughout this assignment was DigitalOcean. DigitalOcean was selected as it was recommended in the brief, it also offered $200 free credits to use instead of paying out of our own pockets.

When the droplets were setup, they had to be selected to different zones around the world to see which area gets attacked more. The final decision was launching a T-Pot in Frankfurt, Bangalore and New York as they were extremely set apart from each other which will also lead the results to be more accurate.

1.1.1 Frankfurt T-Pot Droplet

A Frankfurt droplet was created, and T-Pot was manually installed on it, here are some images to show the details of the created droplet.

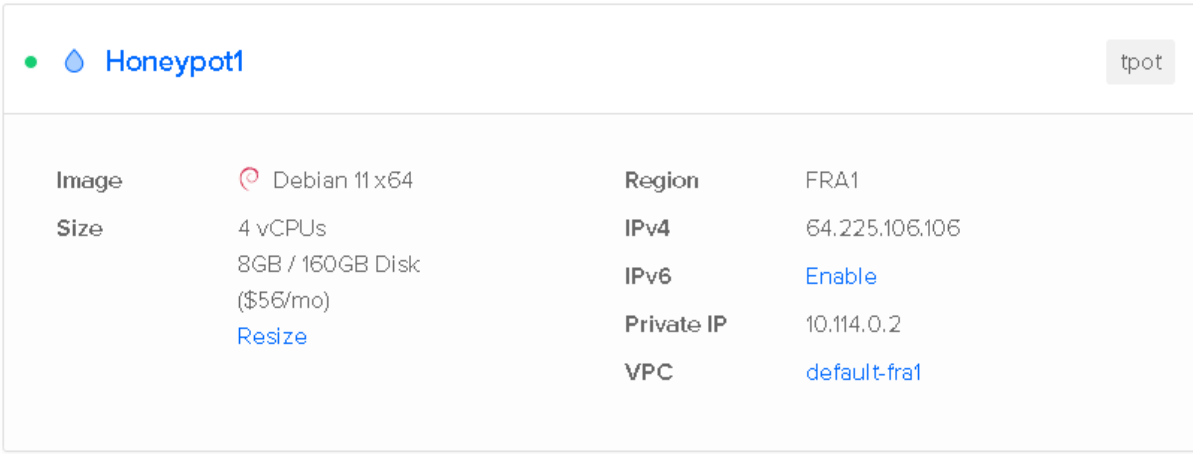


Figure 1 Honeypot1 Frankfurt

These were the specifications that were selected as it was recommended in the original T-Pot GitHub, if you decided to go with a lower spec system, the T-Pots will load insanely slow and won't function properly.

1.1.2 Bangalore T-Pot Droplet

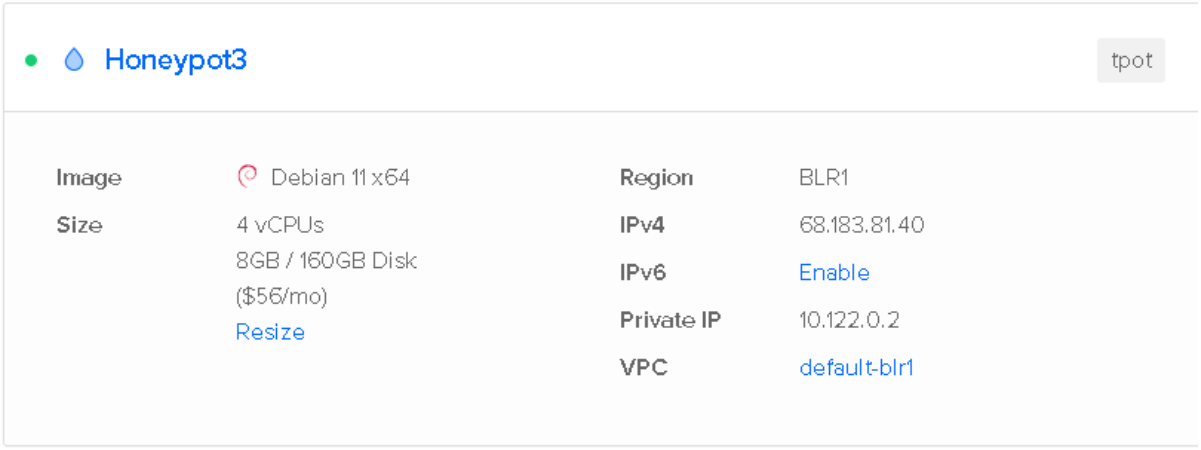


Figure 2 Honeypot3 Bangalore

It was also a requirement to launch the droplet with an Operating System of Debian 11 x64 as it was recommended on their GitHub. Debian 12 was tested and unfortunately it couldn't install as its missing framework.

Figure 3 Operating System Requirement

1.1.3 New York T-Pot Droplet

Finally, here is the information to the final droplet located in New York.

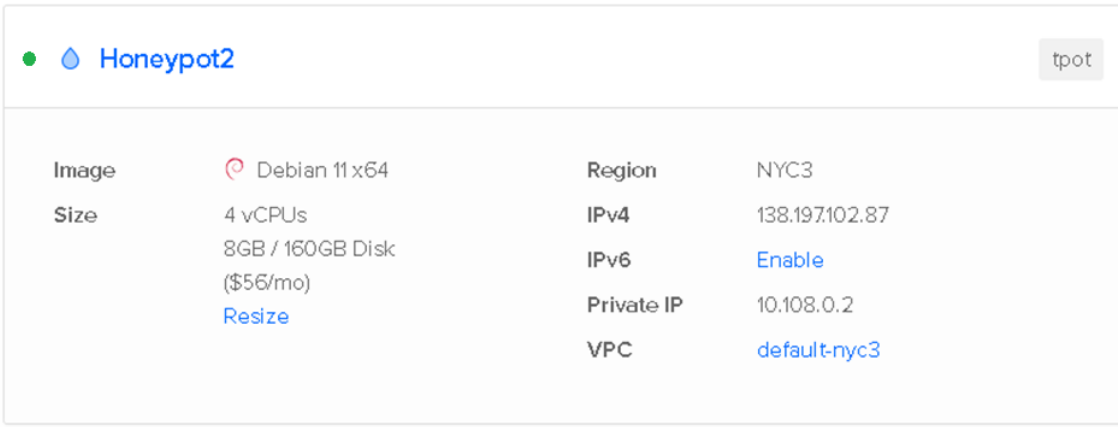


Figure 4 Honeypot2 New York

Droplet	Locations	IPV4
Honeypot1	Frankfurt (Germany)	64.225.106.106
Honeypot2	New York (United States)	138.197.102.87
Honeypot3	Bangalore (India)	68.183.81.40

2. Which Honeypots were used

T-Pot is an all-in-one multi–Honeypot Platform, so instead of installing individually each honeypot onto a machine, all you need is to download T-Pot from its GitHub page and it comes along with all honeypots included then you can manually edit the tpot.yml file and choose which honeypot services you want running and not.

2.1 Suricata Honeypot

Suricata is an open-source Network IDS (Intrusion Detection and Prevention System) and a Network Security Monitoring Engine. Its focus is to monitor network traffic for any suspicious activities and attacks, it provides real-time alerts and prevention options.

2.2 Cowrie Honeypot

Cowrie is a recommended honeypot to be used as its purpose is to track SSH and Telnet, it also logs brute force attacks and shell activities by attackers.

2.3 Dionaea Honeypot

Dionaea honeypot is an embedded python script that used libemu to be able to detect shellcodes, supporting ipv6 and tls.

2.4 Ddospot Honeypot

DDoSPot is a honeypot that can track and monitor DDOS (Distributed Denial of Service) attacks.

3. Analysis of gathered data

In this section, all the data that has been gathered over the space of one month will be published here, it will highlight the main data such as which countries attacked most, top attacking IP, different types of credentials used for the attempted logins better said brute force attacks, which honeypot performed best, the remaining data will be displayed further in this document.

3.1 Total Attacks towards each droplet

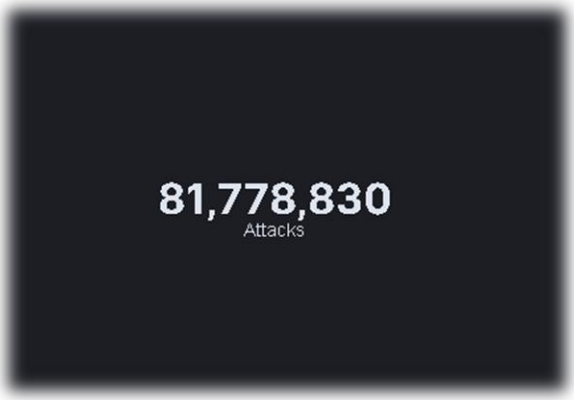


Figure 5 Total Attacks towards Bangalore

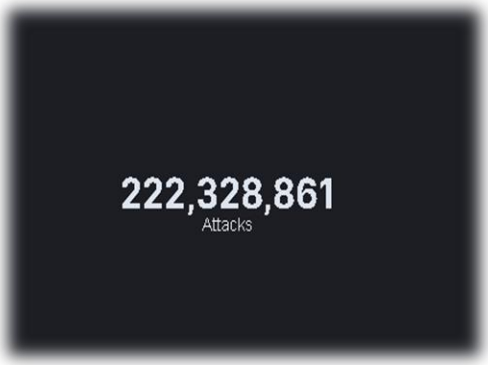


Figure 6 Total Attacks towards New York

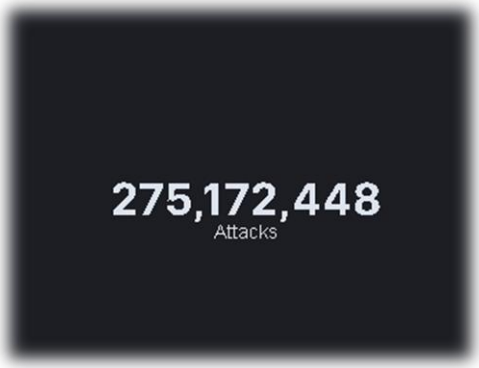


Figure 7 Total Attacks towards Frankfurt

After one month has passed, it shows that Frankfurt was the droplet which got attacked the most compared to the others, the reason for this is only in relation with the traffic patterns whereas some regions may experience higher levels of scanning and attempts of attacks due to the concentration of attackers or automated bots. Another reason could be that automated bots have some set IP ranges for vulnerabilities, so that means that it possibly detected more threats on the Frankfurt IP compared to Bangalore and New York.

3.2 Top Country Attacks

Bangalore

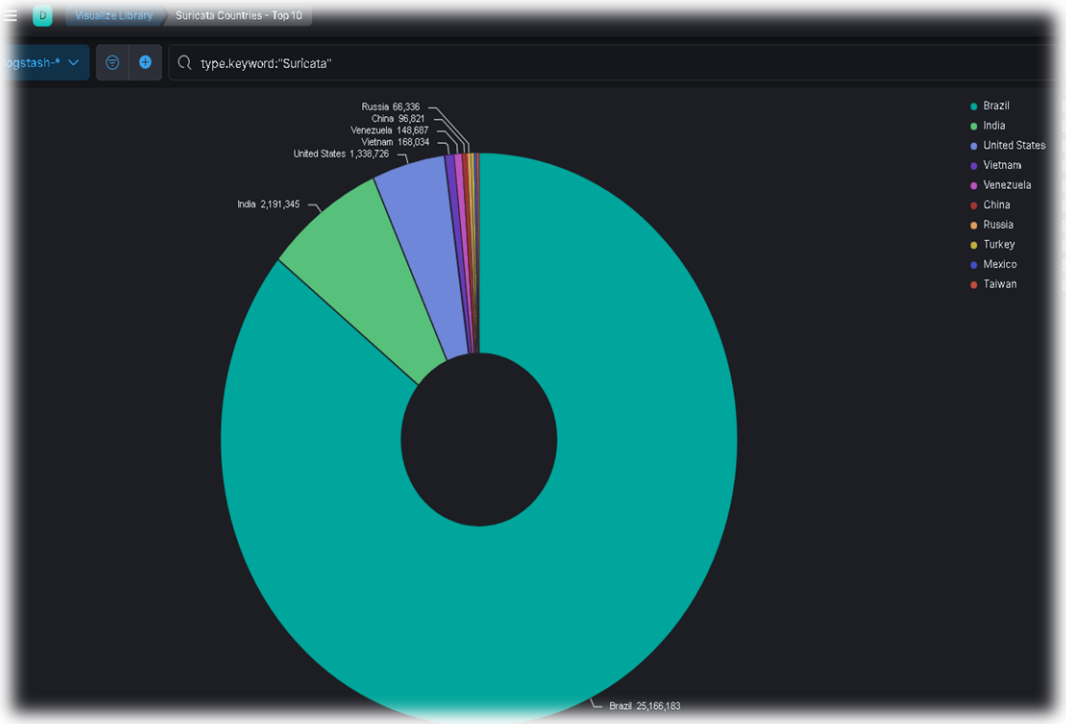


Figure 8 Top Country Attacks Bangalore

New York

Figure 9 Top Country Attacks New York

Frankfurt

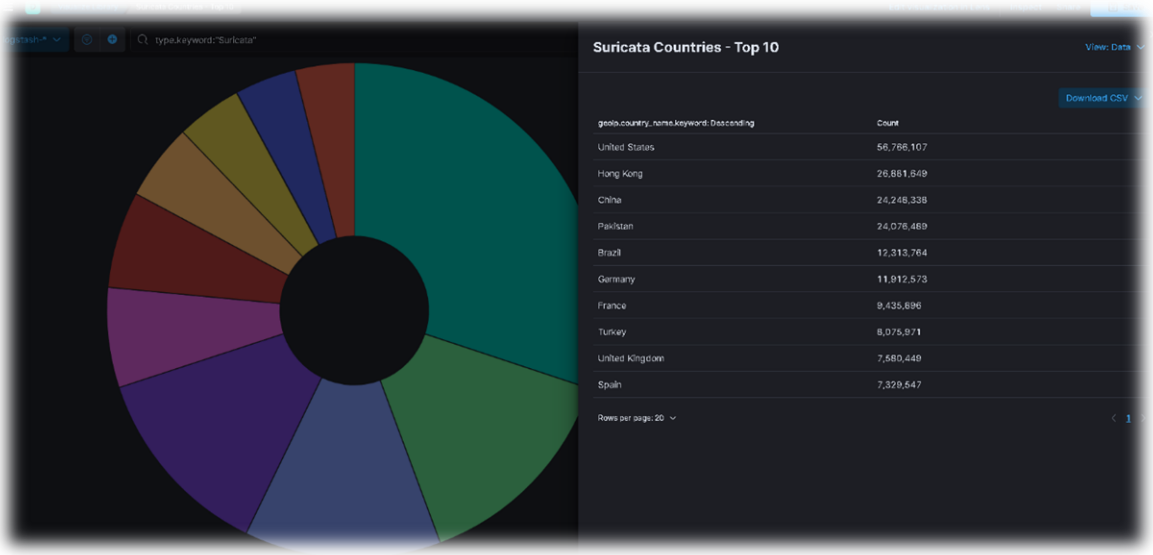


Figure 10 Top Country Attacks Frankfurt

Country	Number of Attacks
United States	97,640,604
Pakistan	65,148,072
Brazil	50,754,969
China	32,527,107
Hong Kong	26,816,649

The top country that has attacked the most was America; however, it does not mean that it was America itself as attackers could either have zombies located all around the world or even easier, the attacker could use a VPN on the attacking machines.

3.3 Cowrie Top URL Downloads

Frankfurt

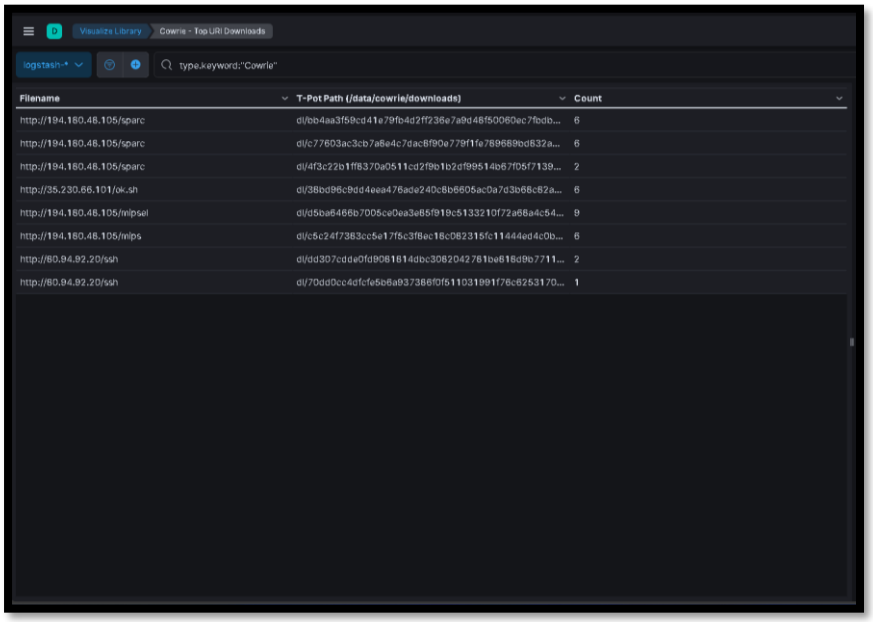


Figure 11 Frankfurt TOP URL

New York

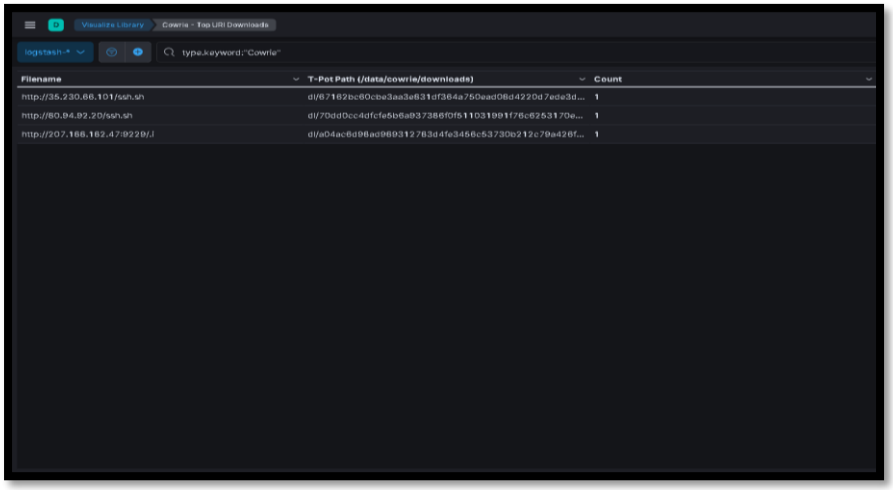


Figure 12 New York TOP URL

Bangalore

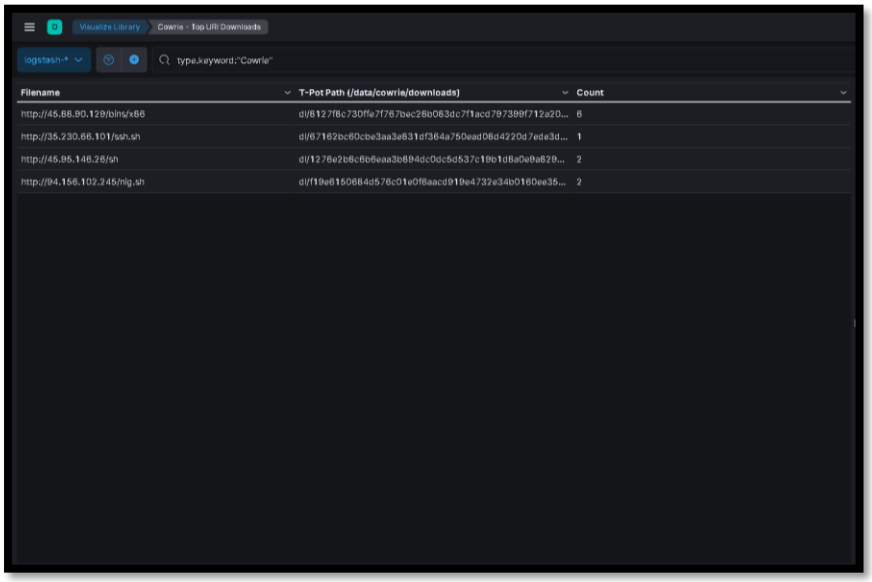


Figure 13 Bangalore TOP URL

The first URL that was taken into investigation with the use of VirusTotal was from Bangalore, the selected URL was ***http://45.88.90.129/bins/x86*** . This URL was selected as it had the highest count compared to the others.

Here are the results after the URL was entered and scan through VirusTotal:

Figure 14 VirusTotal Result

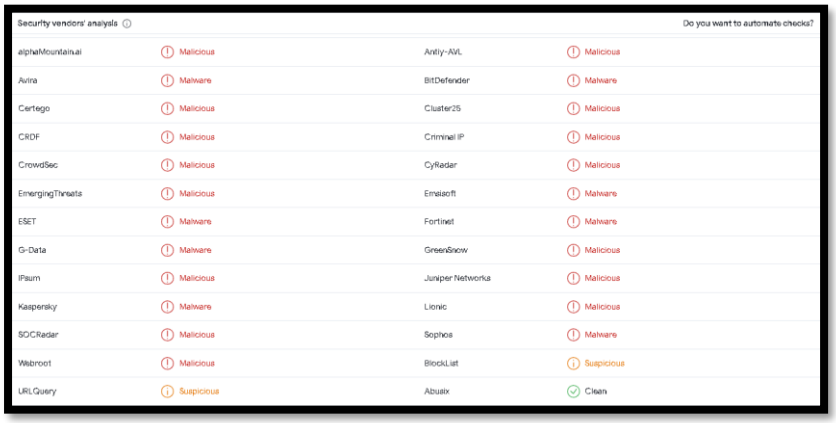


Figure 15 VirusTotal Contained Malware

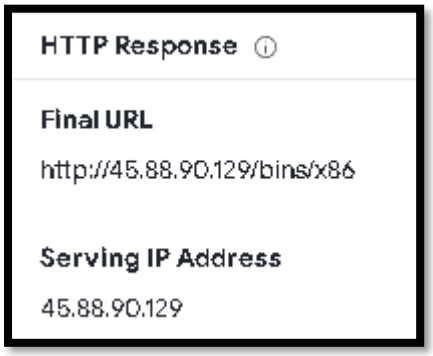


Figure 16 URL IP Track

After the IP was placed into AbuseIPDB, it showed that the IP was 100% Abuse and was report highly number of times by users. It shows that it comes from a Hosting Provider, this could mean that possibly a system within the Hosting Providers got infected and is now being used as an attacker.

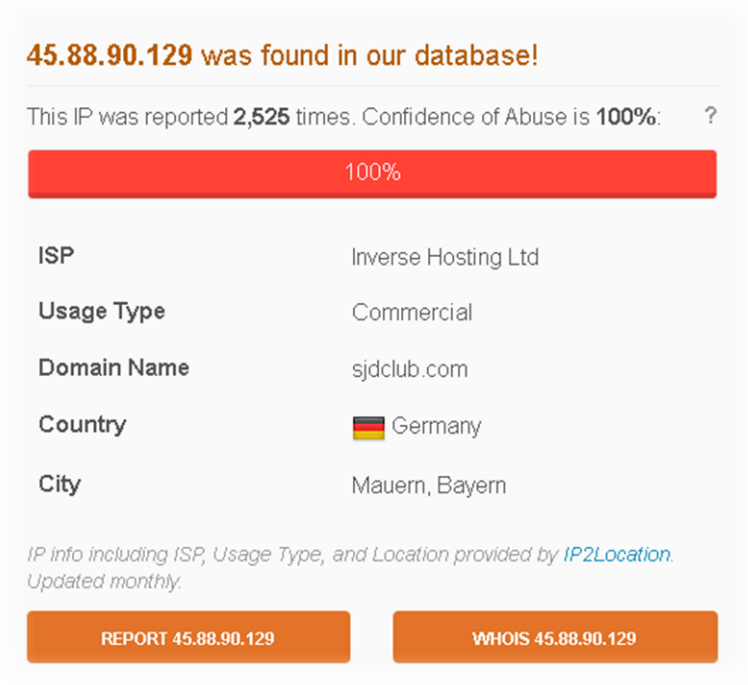


Figure 17 AbuseIPDB Result

Now moving onto the New York URL Download, it only shows 3 URLs and each of them have only a count of 1. The URL which was chosen to investigate through Virus Total was ***http://35.230.66.101/ssh.sh*** . This URL was chosen as we can see that the attacker included a SSH file in the URL, this means that if the machine runs the URL, it will install a malicious virus which can either allow the attacker to gain control to the machine or even just installing a malicious malware on it. The URL was noted and inserted into Virus Total and here are the results:

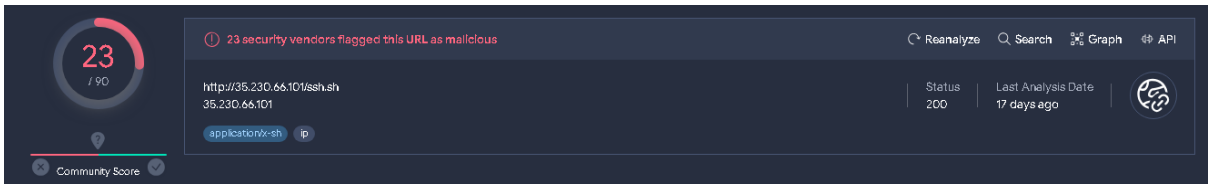


Figure 18 VirusTotal Result NY

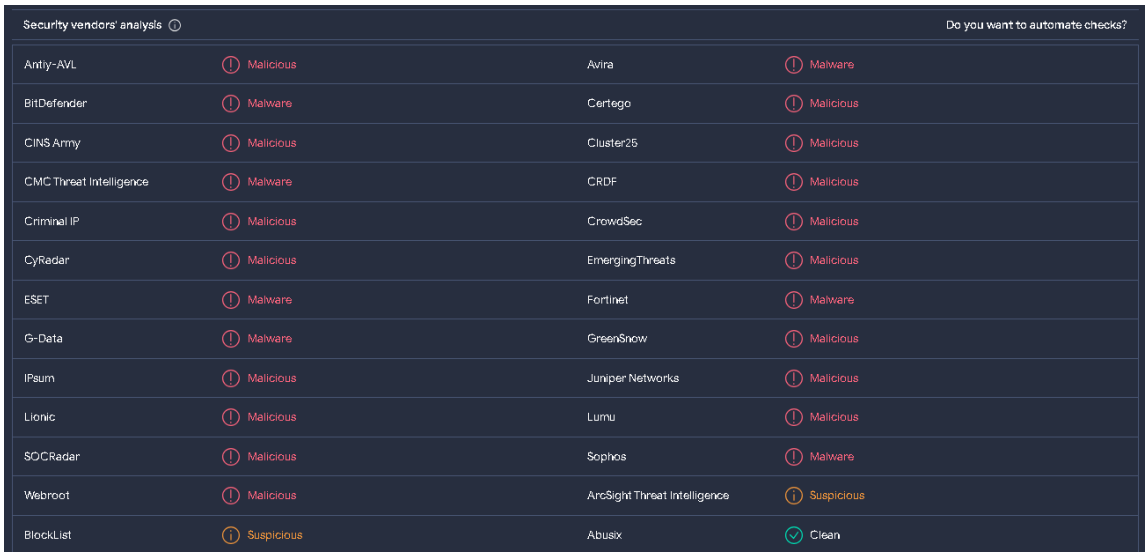


Figure 19 VirusTotal Malicious Detection

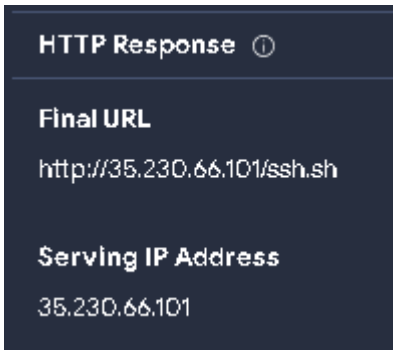


Figure 20 VirusTotal HTTP Response

After checking in the HTTP response, the IP was noted and was then placed into the AbuseIPDB to check if the IP has been reported and where it comes from. Here are the results:

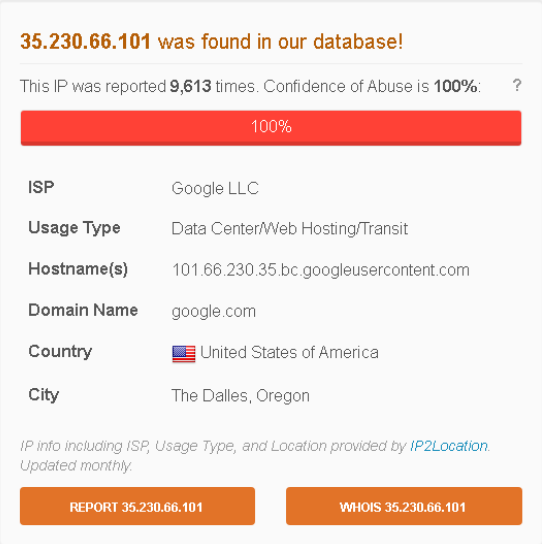


Figure 21 AbuseIPDB

After viewing the information from the IP, we can see that it has been reported 9,613 times from different users and it is currently sitting at 100% abuse, this means that the IP is highly infected and should be ignored. It also shows us where the requests are coming from, it comes from a Hosting Provider that is in Dalles, United States of America.

Now the last machine which will be investigated will be the one from Frankfurt, Germany. The URL that was picked and decided to take an in-depth look at was ***http://194.108.48.105/sparc*** . This URL was chosen as it had count of 14 which gave an alert that it could be suspicious compared to the others. After implementing the IP into Virus Total, surprisingly the result was completely different to what was expected, here are the results:

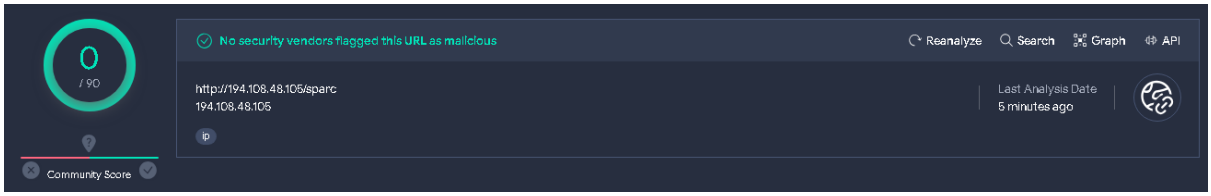


Figure 22 VirusTotal Results

The IP was placed into AbuseIPDB just to clarify that it was recorded by any user for suspicious activity, the results came back clear highlighting that no specific IP was found in their Database. It shows us that the request is coming from T-Mobile located in Czech Republic.

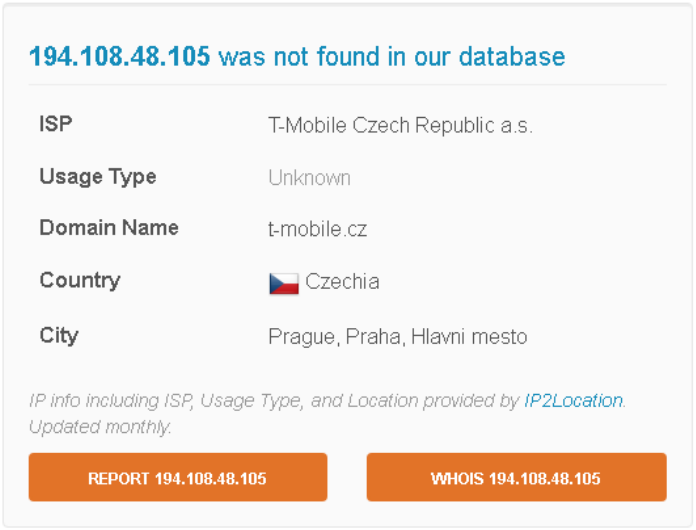


Figure 23 AbuseIPDB Result

3.4 Attacks by Ports

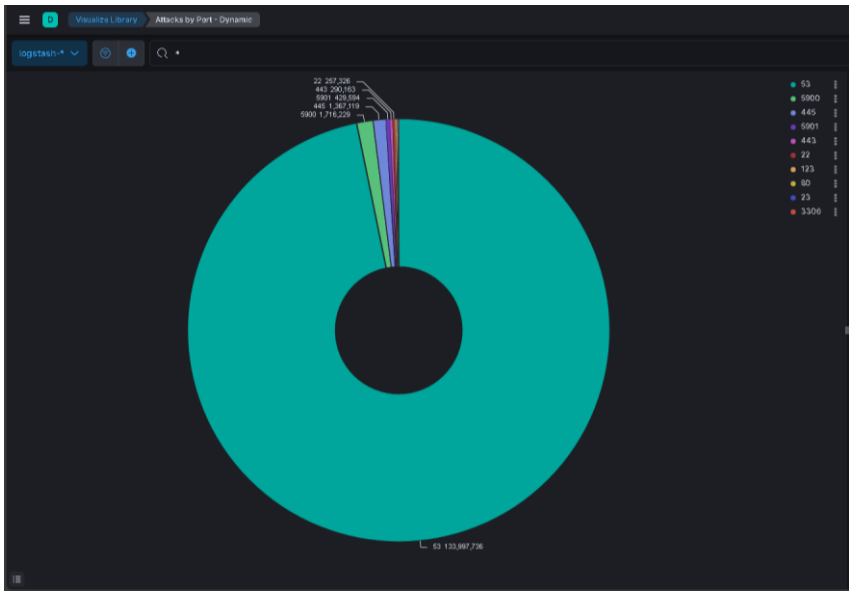


Figure 24 Attacks by Ports Frankfurt

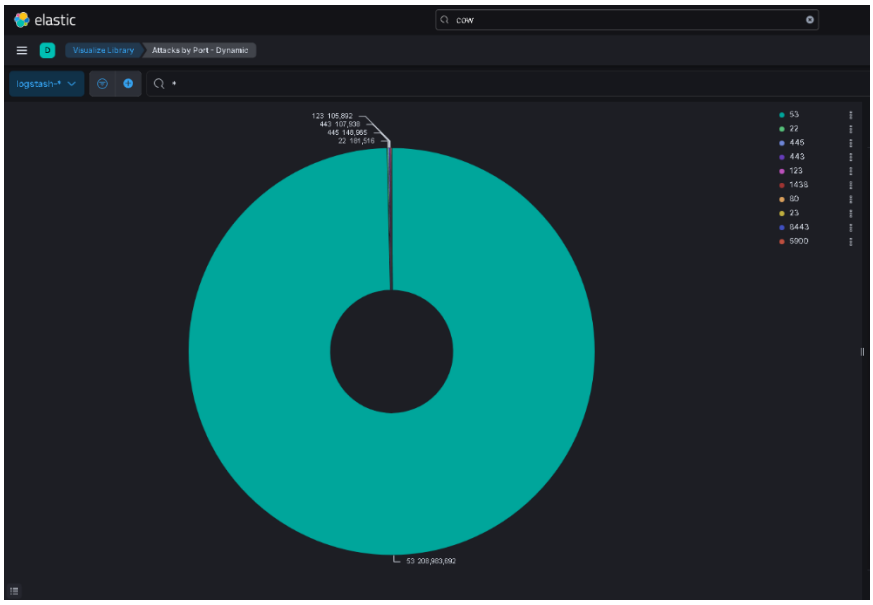


Figure 25 Attacks by Ports New York

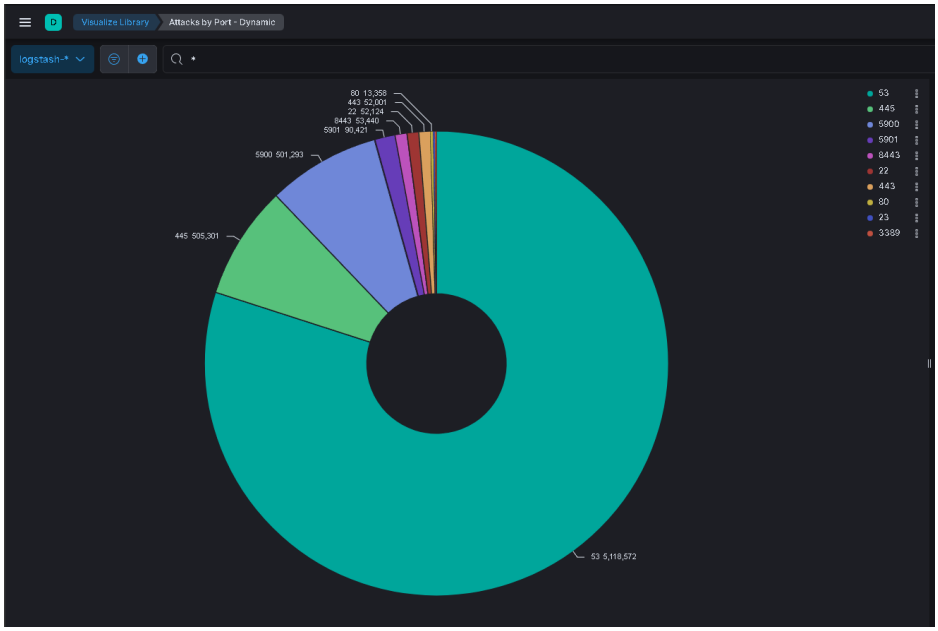


Figure 26 Attacks by Ports Bangalore

After looking at the data gathered for port attacks, it highlights which ports attackers thought might be open and went ahead with attacking them. Here is a quick table showing the top 5 ports that were attacked and what they are used for.

Ports	Used
53	Used for both UDP and TCP communication in DNS
445	Used for SMB (Server Message Block) over TCP

5900	Used for VNC (Virtual Network Computing)
443	Used for secure web browsing using HTTPS (Hypertext Transfer Protocol Secure) protocol
80	Encrypted HTTP ((Hypertext Transfer Protocol) traffic

The port which was attacked the most was port 53, the only reason for this is because the attacker was attempting to gain access through the port and have the ability to hijack the machine and use it as a zombie, it could also be that an attacker was planning to perform a DDOS Attack.

3.5 Username and Password Attempts

When it comes to Cowrie Honeypot, it was able to track each username and password that has been entered by the attackers in order to try gain access to the machine, this does not mean that the attackers manually entered these credentials, its more likely they attempted a brute force attack as it will complete the attack faster than entering details manually.

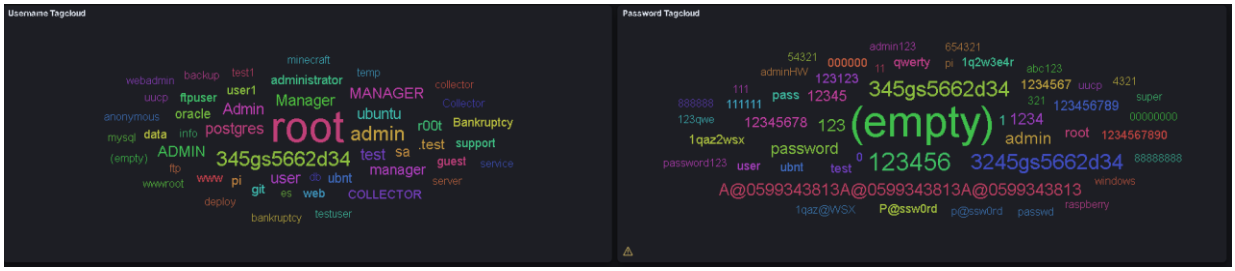


Figure 27 Frankfurt Username & Password

Figure 28 New York Username & Password

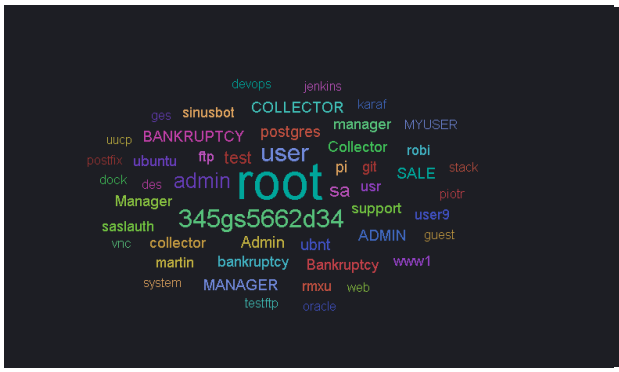


Figure 29 Bangalore U&P

Figure 30 Bangalore U&P

The left portion is the entered usernames, and the right portion is the attempted passwords from attackers. It instantly shows us that “Root” username has been used the most out all as with Cowrie the graphical view gives the information instead of having it counted in numbers of attempts, the same goes for the password, “(empty)” has been tried the most also.

3.6 Suricata CVE – Top 10

Using Suricata, the top CVEs were tracked and noted automatically. CVE stands for “Common Vulnerability and Exposures, it is a system that identifies, names and tracks vulnerabilities which are either in software or hardware systems. Each CVE which is tracked has its own unique identifier for example “CVE-YYYY-NNNN” whereas the “YYYY” indicates the year when the CVE was assigned and the “NNNN” is just the unique identifier. Here is a list of CVEs that were captured throughout the three machines.

Figure 31 Top 10 CVE Frankfurt

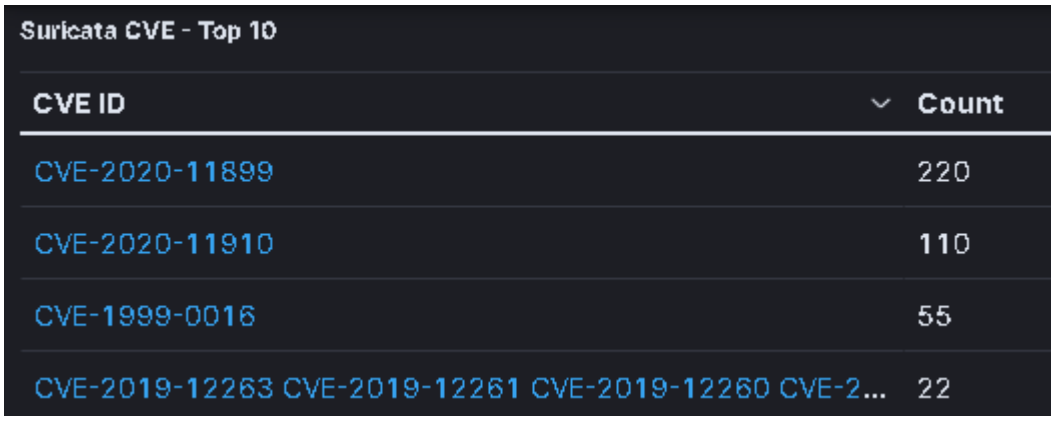


Figure 32 Top 10 CVE New York

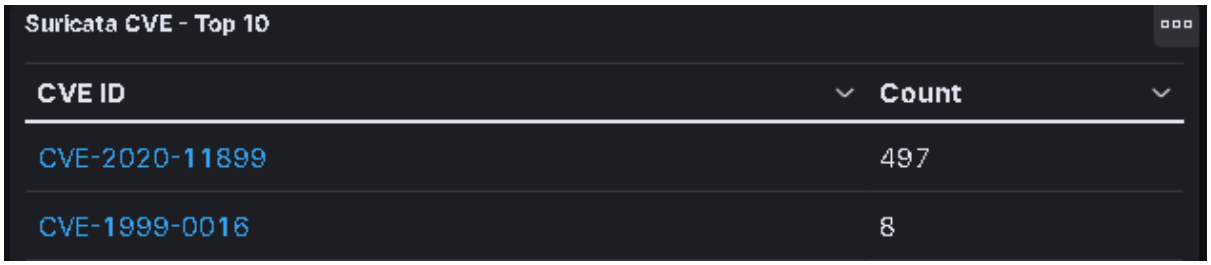


Figure 33 Top 10 CVE Bangalore

Each CVE was noted, the final decision was to take note of the highest counts CVE and get an in-depth information of them as if the count is higher, it means its more achievable than the others. The first CVE which was checked was from Frankfurt with a CVE identifier of “CVE-2020-11899”. To find in-depth information in relation to the CVE a website was needed for this, the website that was used was “CVEdetails.com” as it’s the most recommended out of all. After scanning the CVE on the website, this is the result:

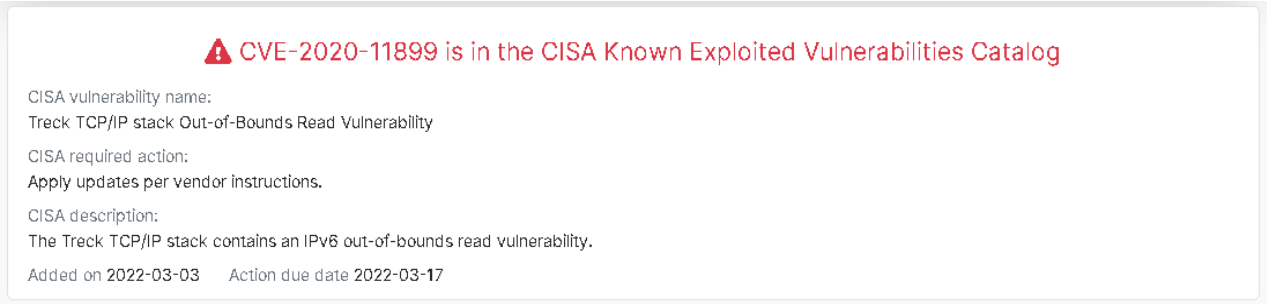


Figure 34 Top CVE Frankfurt

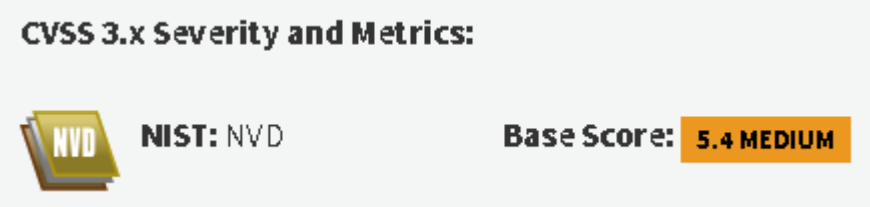


Figure 35 CVSS Score

The final CVSS score was a 5.4, this means that it is a "Medium Severity", that also means that it is a moderate-risk vulnerabilities that require attention and remediation.

Moving onto the New York machine, the top CVE was again "CVE-2020-11899" so the only option was to get information on the 2nd highest counted CVE which was "CVE-2020-11910", here are the results gathered from that.

Figure 36 Top CVE New York

Figure 37 CVSS Score

We can see that the CVE identifiers are different to each other, but they have similar issues, for the "CVE-2020-11899" the issue is that it has an out of bounds IPv6 whereas for "CVE-2020-11910" it has an out of bounds ICMPv4. It also has a CVSS score of 5.3 which will still be in the same medium severity as the first scanned CVE.

The final machine is the Bangalore machine, the same goes for this machine, it has the same top CVE as the other machines, so the option is to go ahead and take information from the CVE that hasn't been analysed yet. The CVE that will be chosen will be "CVE-1999-0016". Here are the following results for the recording of the data:

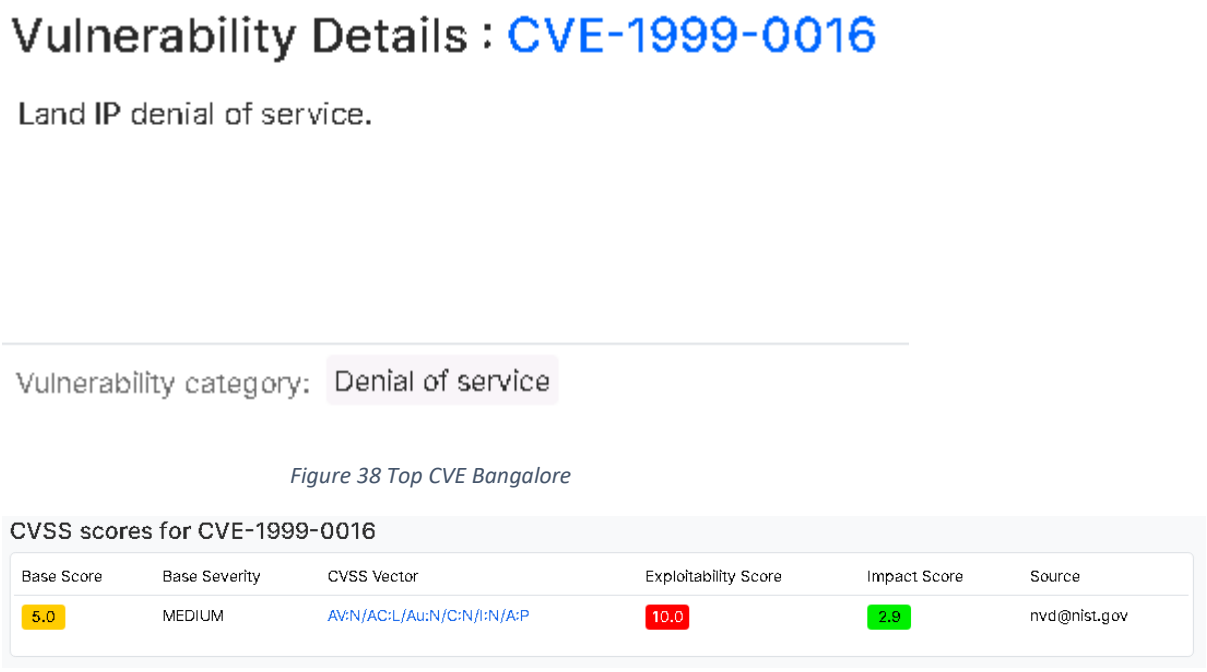


Figure 38 Top CVE Bangalore

Figure 39 CVSS Score

This CVE is different to the others, this time this website shows us that this specific CVE specifies as a denial of service, it also has a score of 5.0 which will place it in the medium severity.

3.7 Attacks by Honeypots

Here are some screenshots taken after one month of data gathering showing data of attacks by honeypots.

Here are the Bangalore Results:

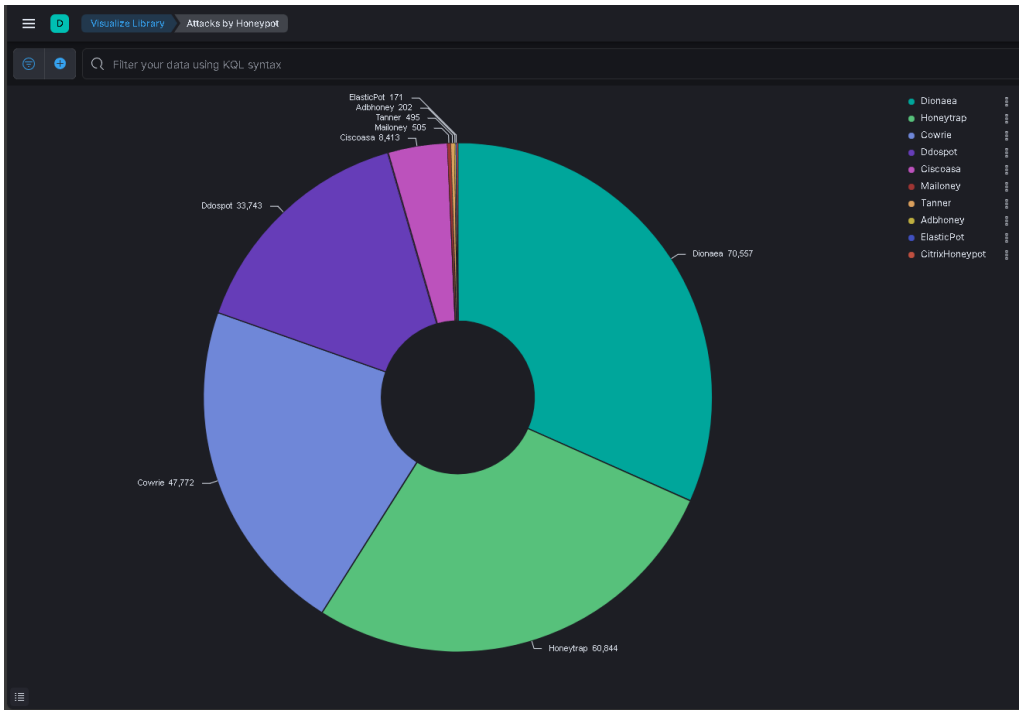


Figure 40 Bangalore Honeypots Attacks

Figure 41 Attacks by Honeypots Bangalore

In these images it shows that Cowrie was the honeypot that gathered the most data from attackers compared to the other honeypots.

Frankfurt Results:

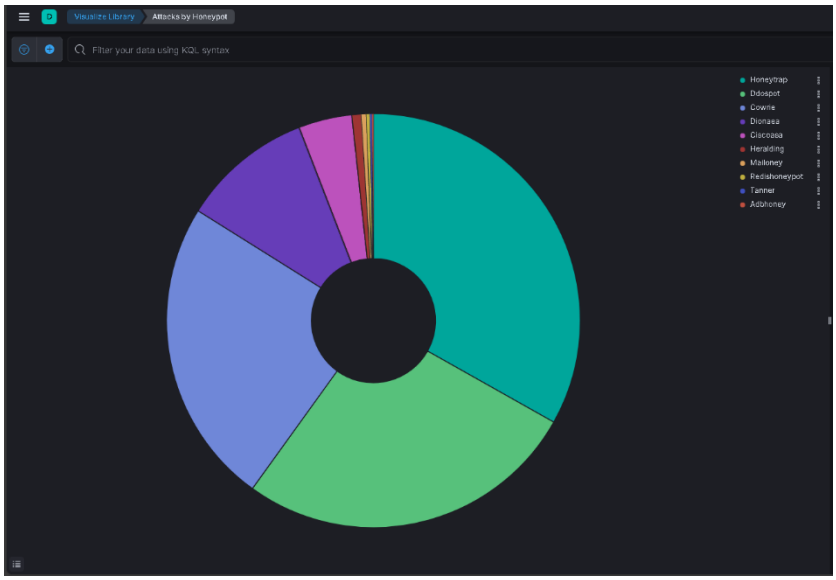


Figure 42 Frankfurt Honeypots Attacks

Figure 43 Attacks by Honeypots Frankfurt

For this machine it shows that Honeytrap was the honeypot that gathered most data from attackers compared to the other honeypots.

New York Results:

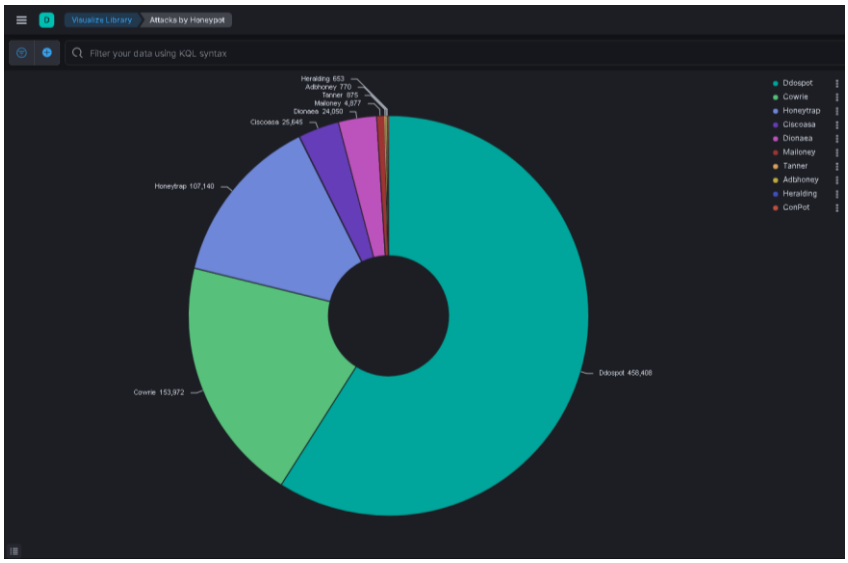


Figure 44 New York Honeypots Attacks

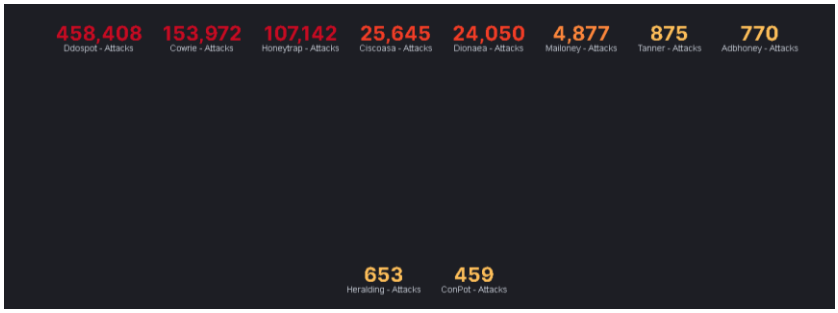
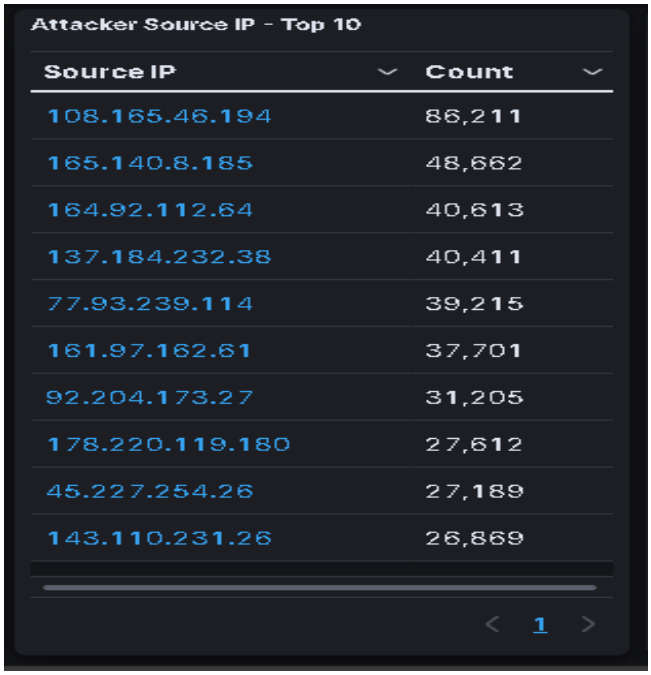


Figure 45 Attacks by Honeypots New York

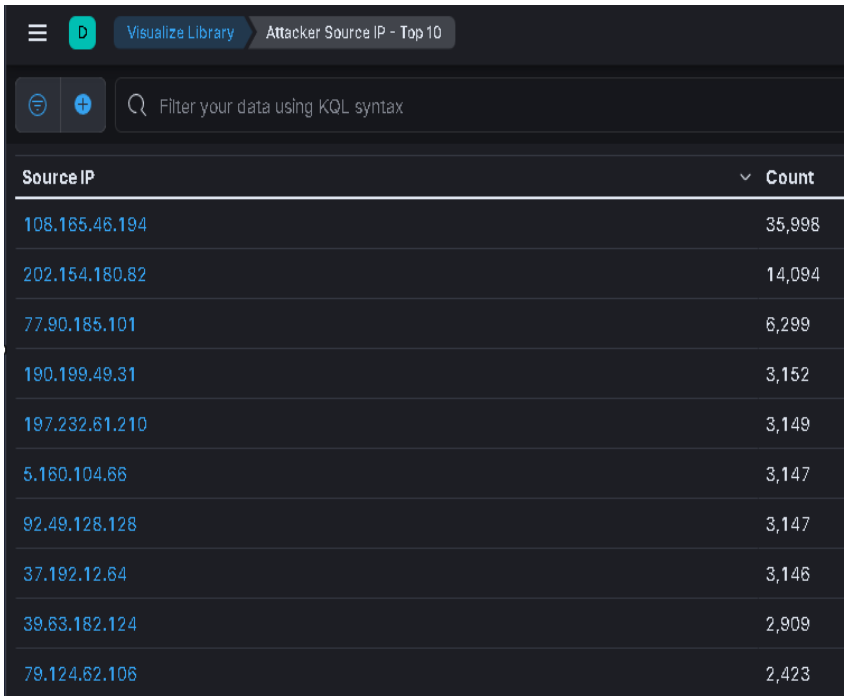
Finally in these images it highlights that Ddospot was the honeypot that managed to capture most data from incoming attacks compared to other machines.

3.8 Top IP Attacker



Frankfurt

Figure 46 Top IP Attacker Frankfurt



New York

Figure 47 Top IP Attacker New York

Bangalore

Figure 48 Top IP Attacker Bangalore

3.9 Attacker AS/N – Top 10

Throughout this section it shows displays the attackers from which field or organization the attack came from, here is some screenshots that show the number of times the attacker sent attacks towards the machine.

Bangalore:

Attacker AS/N - Top 10 - Dynamic

AS	ASN	Count
45899	VNPT Corp	51,863
58466	CHINANET Guangdong province network	43,002
8048	CANTV Servicios, Venezuela	41,729
4134	Chinanet	39,767
9829	National Internet Backbone	31,511
8151	UNINET	29,872
47331	Turk Telekom	28,663
7552	Viettel Group	27,395
10429	TELEFONICA BRASIL S.A	26,282
3462	Data Communication Business Group	20,989

Figure 49 AS/N Bangalore

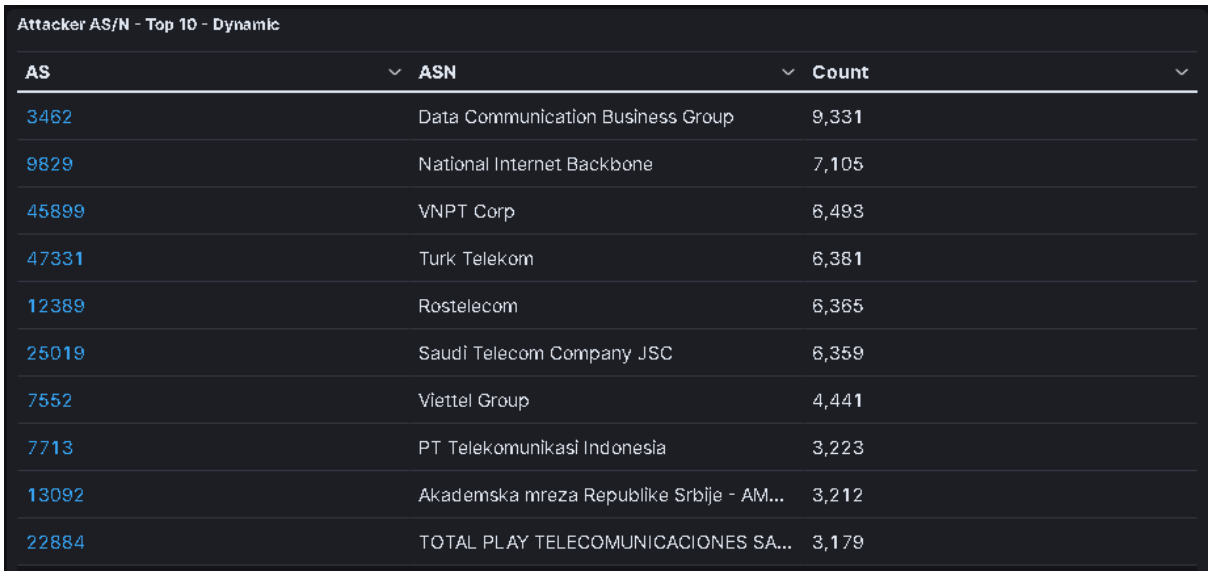
Frankfurt:

Attacker AS/N - Top 10 - Dynamic

AS	ASN	Count
45899	VNPT Corp	30,879
208091	Xhost Internet Solutions Lp	22,083
9299	Philippine Long Distance Telephone Comp...	20,251
12389	Rostelecom	9,497
8452	TE Data	8,141
20207	Gigared S.A.	8,041
24445	Henan Mobile Communications Co.,Ltd	7,683
9541	Cyber Internet Services Pvt Ltd.	7,665
58224	Iran Telecommunication Company PJS	7,465
4134	Chinanet	7,334

Figure 50 AS/N Frankfurt

New York:



Attacker AS/N - Top 10 - Dynamic

AS	ASN	Count
3462	Data Communication Business Group	9,331
9829	National Internet Backbone	7,105
45899	VNPT Corp	6,493
47331	Turk Telekom	6,381
12389	Rostelecom	6,365
25019	Saudi Telecom Company JSC	6,359
7552	Viettel Group	4,441
7713	PT Telekomunikasi Indonesia	3,223
13092	Akademska mreza Republike Srbije - AM...	3,212
22884	TOTAL PLAY TELECOMUNICACIONES SA...	3,179

Figure 51 AS/N New York

These screenshots highlight which field or organisations have the highest count of attacks towards the deployed machines.

4.0 Cowrie Command Line Input – Top 10

When it comes to attacking, attackers also perform CLI (Command Line Interface) attacks such as,

- Injection Attacks
- Cross-Site Scripting
- File Upload Vulnerabilities
- Remote Code Execution
- Butter Overflow
- Malicious Downloads
- Social Engineering
- Exploiting Default Credentials
- API and Web Service Attacks
- Brute Force Attacks

Here are some screenshots of what commands attackers have inputted in order to try retrieve data from the deployed machines.

Bangalore:

Cowrie Input - Top 10

Command Line Input	Count
cd ~; chattr -ia .ssh; lockr -ia .ssh	513
lockr -ia .ssh	513
cat /proc/cpuinfo | grep name | wc -l	486
cat /proc/cpuinfo | grep name | head -n 1 | awk '{print $4,$5,$6,$7,...	468
uname -a	468
free -m | grep Mem | awk '{print $2 ,$3, $4, $5, $6, $7}'	463
ls -lh $(which ls)	460
which ls	460
crontab -l	457
cat /proc/cpuinfo | grep model | grep name | wc -l	454

Figure 52 Command Line Bangalore

Frankfurt:

Cowrie Input - Top 10

Command Line Input	Count
cd ~; chattr -ia .ssh; lockr -ia .ssh	1,425
lockr -ia .ssh	1,425
cat /proc/cpuinfo | grep name | wc -l	1,296
free -m | grep Mem | awk '{print $2 ,$3, $4, $5, $6, $7}'	1,249
cat /proc/cpuinfo | grep name | head -n 1 | awk '{print $4,$5,$6,$7,...	1,248
ls -lh $(which ls)	1,233
which ls	1,233
crontab -l	1,223
w	1,213
uname -a	1,211

Figure 53 Command Line Frankfurt

New York:

Cowrie Input - Top 10

Command Line Input	Count
lockr -ia .ssh	588
cd ~; chattr -ia .ssh; lockr -ia .ssh	587
cat /proc/cpuinfo | grep name | wc -l	538
cat /proc/cpuinfo | grep name | head -n 1 | awk '{print $4,$5,$6,$7,...	517
free -m | grep Mem | awk '{print $2 ,$3, $4, $5, $6, $7}'	517
crontab -l	512
ls -lh $(which ls)	509
which ls	509
uname -a	506
w	505

Figure 54 Command Line New York

From the screenshots which are displayed, it shows that the command “lockr -ia.ssh” has been executed across all three droplets, this command mean s that when it is being executed it is trying to lock that specific file in the machine. The next command that has been used across all three droplets was “cd ~; chattr -ia .ssh; lockr -ia .ssh”. This command means that the attacker wants to redirect himself to the home directory and try run that same command again just in a different directory just in case the machine has it set somewhere else.

5. Improvements

Overall, the assignment went perfectly well, the data was gathered correctly and exactly over the space of one month. However, if I somehow must come across something like this one thing that I would change for sure is doing more research about T-Pot itself and the Elastic dashboard / menu itself. The main issue that has been faced was after the month data has been recorded, in the “Discover Options” there was a setting called “Index Lifecycle Policies”, this allowed you adjust the setting of the deleting phase. The default deleting phase was automatically set to thirty days where it was fine for the data gathering but after the one month, the data started to erase bit by bit meaning that the only fix for it was to navigate to the specified area and adjust it from 30 days to over 30 days such as 100 days for example. That would be the only change that would be taken if this assignment was seen again.

6. Conclusion

In summary, the month-long deployment and subsequent analysis of the TPOT honeypot provided a nuanced understanding of a complex cybersecurity environment. The comprehensive dataset captured provided a complete picture of malicious activity, from routine automated scans to more

advanced attack techniques. This first-hand understanding of potential vulnerabilities provides an opportunity to refine and strengthen your security strategy to ensure robust protection against evolving threats. The analysis process not only identified discernible patterns and trends, but also facilitated refinement of incident response protocols. In the future, insights gained from this work will help us continually optimize our cybersecurity practices and strengthen proactive and adaptive approaches to meeting emerging challenges. This experience highlights the need for continuous vigilance and dynamic defence, which are critical elements in addressing ever-changing cybersecurity threats.

7. Reference

DigitalOcean (2022) 'Simple Cloud Infrastructure for Developers', DigitalOcean, Available at: https://www.digitalocean.com/ (Accessed: 27 September 2023).

Telekom Security (2018) 'TPOT Community Edition', GitHub, Available at: https://github.com/telekom-security/tpotce (Accessed: 27 September 2023).

AbuseIPDB (2016) 'AbuseIPDB - Report IP Addresses Engaged in Malicious Activity', AbuseIPDB, Available at: https://www.abuseipdb.com/ (Accessed: 20 October 2023).

MITRE Corporation (1999) 'Common Vulnerabilities and Exposures (CVE) - Home', MITRE CVE, Available at: https://cve.mitre.org/ (Accessed: 30 October 2023).