TED University/ CMPE491

Senior Project

Final Report

GROUP MEMBERS

AYŞE NUR ŞAFAK 16271073704

KAYRA EGE ARDA 51508242928

BARAN AKIN 36194302446

EBRU KILIÇ 1444920268

SUPERVISOR

EMİN KUĞU

JURIES

VENERA ANDANOVA

TOLGA KURTULUŞ ÇAPIN

Date
25-05-2022

FINAL REPORT

Table of Contents

FINAL REPORT

Abstract

The use of the internet is growing in all parts of the world. Cybercrime and the increasing number of online users are both potential threats to data loss.
When an attacker's identity and actions are known, a system can be effectively protected.
Honeypot is a fake service that provides logical responses to assist in taking information about an attacker's whole shell interaction. Honeypot is a critical tool for monitoring cyberattacks and analyzing how we encounter attackers.
Cowrie is an interaction SSH honeypot, used for attacks. This report is about the results and our observations about the attackers' behavior from data logs. It took months to collect these data by monitoring.

Introduction

The rising vulnerabilities are one of the most regarding situations of the day. So, every day, billions of attacks on small, medium, and large networks appear around the world. Some detection tools have been developed to protect networks by gathering information from previous malicious attacks. A honeypot is one of these tools which traps attackers by using fake data that a hacker will attempt to steal from. In this project, we will design a low-interaction, SSH honeypot. It will be installed on a public server in order to investigate cyber-attack methods.

Problem

As the number of ways and strategies used to attack networks increases, the objective of protecting a network must similarly increase. While existing approaches and many other techniques can help to build a secure network, it is important to expect that vulnerabilities will always exist and will be exploited eventually. As a result, we must always develop new ways to combat threats, and one major technique is to install honeypots. While the concept is not recent, we believe that it will give valuable insight into what is vulnerable in our system, and how can we fix these vulnerabilities.

Aim

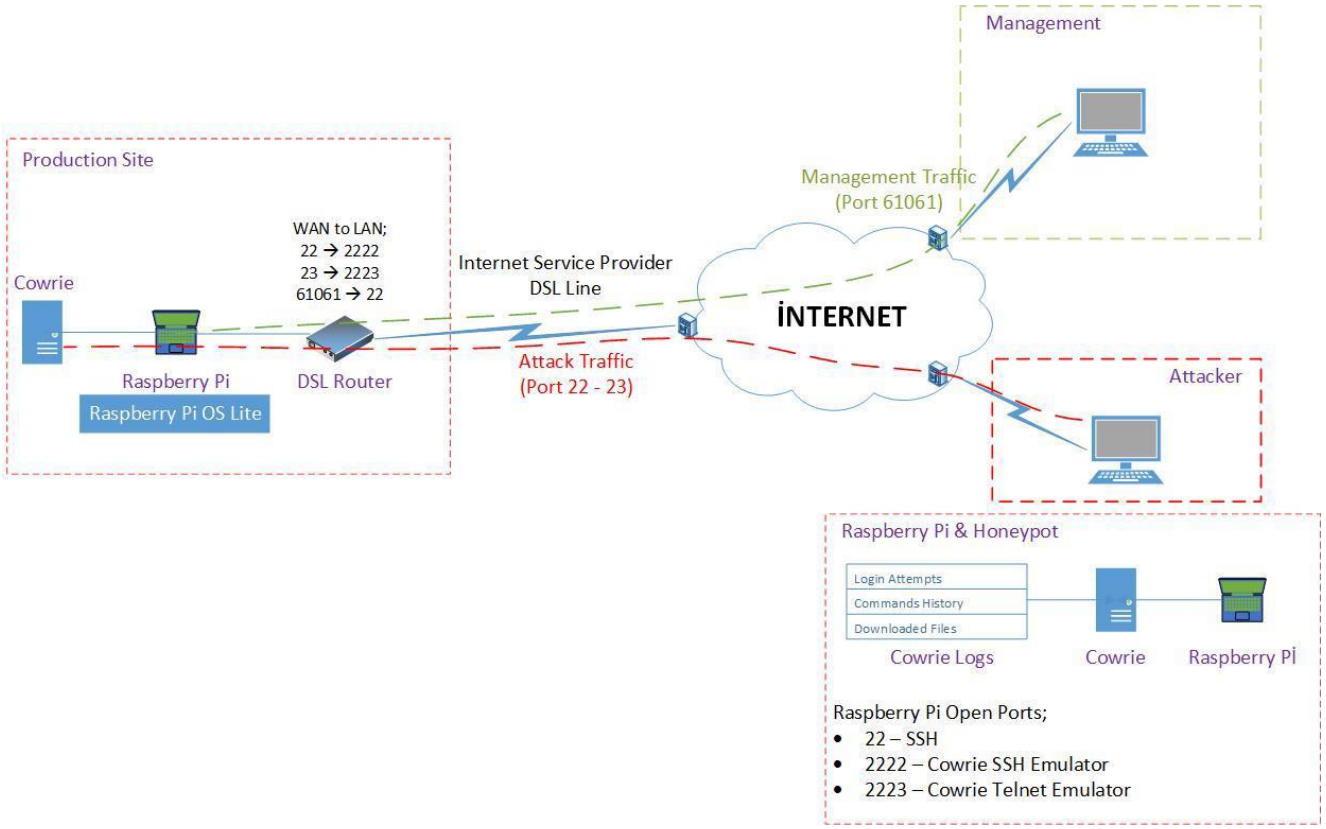
Our project's goal is to design a low-interaction, SSH honeypot. It will be installed on a public Linux server in order to investigate cyber-attack methods. The reason that we want to have our honeypot in the public is that we want to collect organic data about the attacks that may take place and further analyze this data. Our major reason for pursuing this project derives from our curiosity about the cybersecurity, computer networks and cyberattacks. Another one of our goals is that we would like to analyze and learn how the bad actors usually act, thus broadening our experience and views for both defending and pen testing in the future, in our careers. We also plan to use the data that we collect to assist in the manifestation or improvement of some IDS and IPS systems. Furthermore, this attacker data will enable us to predict and learn the behaviors and steps of the bad actors.

FINAL REPORT

System Configuration

Our system comprises glances, supervisord systemd and cowrie. Glances is used to track resource usage systemd is used to start these services automatically if the machine ever goes down supervisord is used to control and monitor the processes that are already running and our main application is the honeypot which is cowrie. We plan to add a static website as a bait application, our hopes are that when the hackers scan the machine that the static website is on they will see a vulnerability in the system which is the honeypot. These applications are all running on an server, our project doesn't require any client, even at the data collection action, we theoretically would be able to go next to the server, plug a cable and take the data off it, our usage of putty and connecting to the server using another port is not technically required nor it requires an client application even if we connect to the server, it is just a means of remotely accessing the server.

Network topology



Test Reports

Unit testing
Since this is open-source project, there is no need to do unit testing. As we point out in the "Test Plan Project", this is a research project in which we combine other codes into one project. But we found some bugs in the website codes and we fix them by unit testing.

API testing
Our project is based on security. Therefore, our APIs must be secure as possible. For testing, we have done a black box penetration test. We do not have any errors. The program worked well. Therefore, we do not need anything to fix it.

Integration Testing
We have checked if our project works well with all services. Since our project has not had many services, we

do not have to check too many combines. Our Honeypot data has parsed well, and we can see the data. After, we make these parsed data readable by codes. We checked if these codes do well. We look at the interaction of all the values with the database. If the values are correct and we can combine them or not.

System Testing

For our system testing, because everything has been built on top of an existing project we have little worries about the interactions between the few newly added components and the system itself, our main concern here is the throughput, which will be tested by a pseudo loading of the system, by swarming our system with “users”.

We have checked about integration handling many requests and the general throughput of the system. As a result, the system worked well, and we do not need anything to fix it.

Performance & Stress Testing

We had too many logs in a small time-space. Also, we had over 10 million logs in a couple of months. That means, our program is strong enough and passes the performance and stress testing.

Results

For our results, we have used another ready-made application to put our log files into an Sql-lite database, the software we have used for the parsing of the logs are as follows.(https://github.com/jasonmpittman/cowrie-log-analyzer)

From that database we have concluded some results. For each category we have decided to graph the mathematical amount of the top 15 results, the average time, what countries does the ip(s) originate from, their duration, commands list of IP (s) possible bot attacks and human attacks based on the time that the attacker spent on the system, and also the username password combinations that they have used. One interesting point was that there were 4 attempts at installing a crypto miner to the computer, based on how the commands were inputted and some spelling errors we concluded that this was a human attacker looking for systems to mine himself some crypto currencies. In total we recorded 2.154.361 attacks that made it into the system.

Figure 2 shows the graph of the attacks on honeypot from the top 15 locations. Clearly, USA produced the most attacks and Germany generated the least attacks. On the other hand, Because attackers may have utilized proxies or proxy chains for anonymity, it's very likely that the observed IP addresses aren't the attackers' true IP addresses.

Figure 2: The graph of top 15 location where attackers attack from

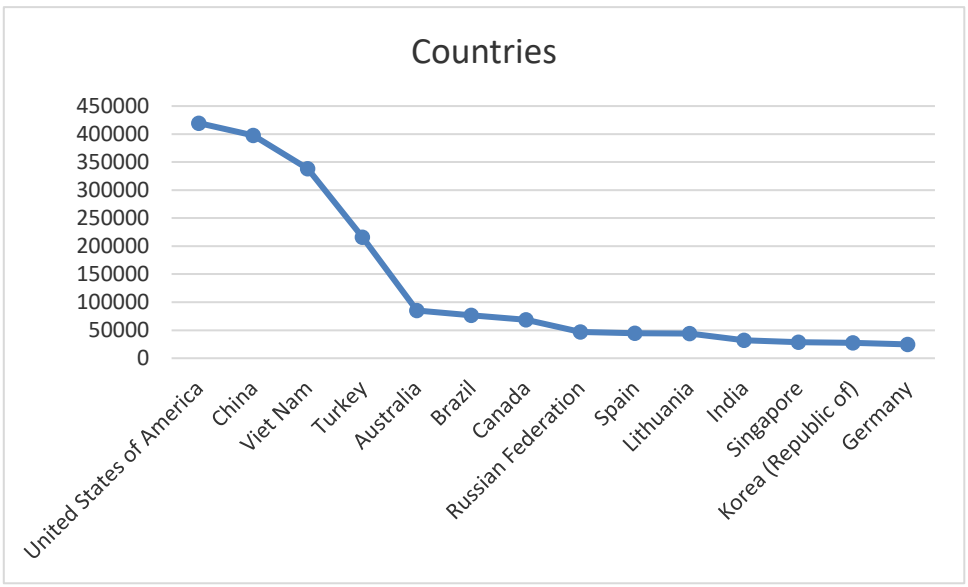


Figure 3 shows the graph of top 15 usernames which were used from attackers. Clearly, admin was used the mostly and ubnt was used leastly.

Figure 3: The graph of top 15 usernames which were used from attackers

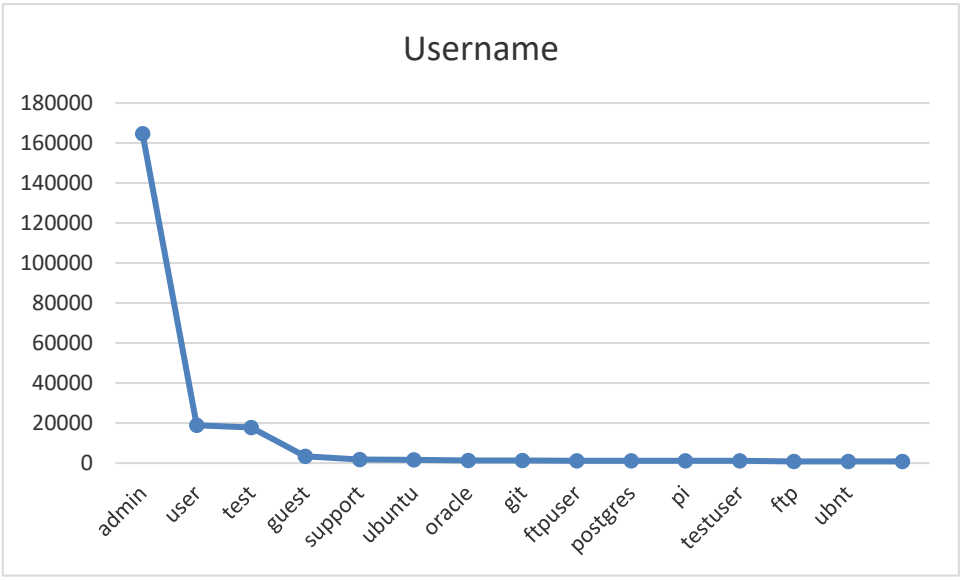
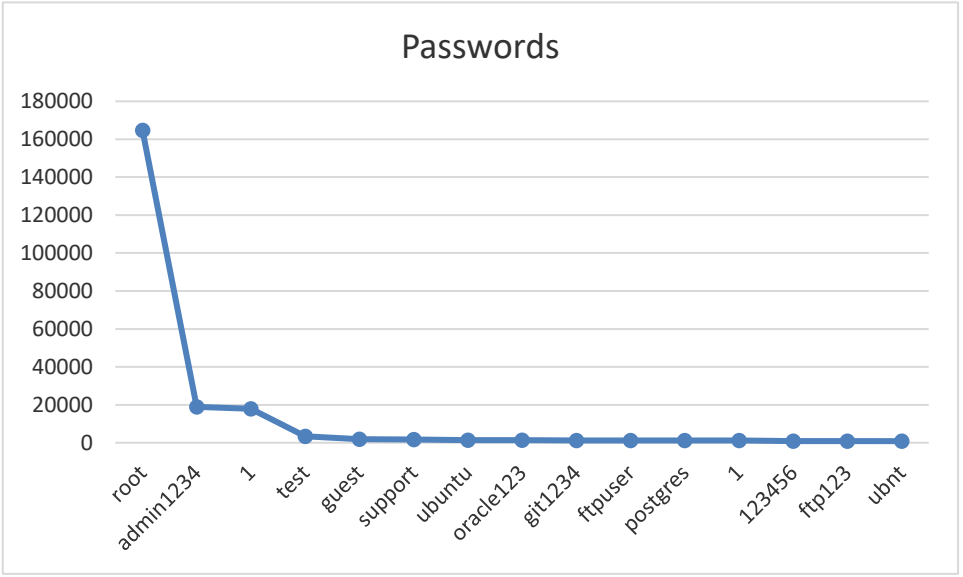


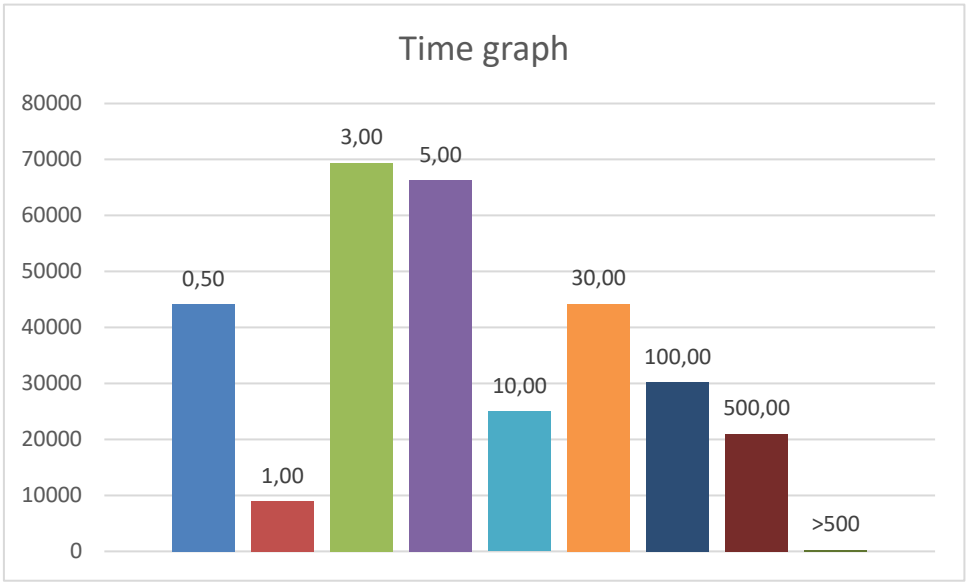
Figure 4 shows the graph of top 15 passwords which were used from attackers. Clearly, root was used the mostly and ubnt was used least. Thereofore, “root” is the weakest link of the chain.

Figure 4: The graph of top 15 passwords which were used from attackers.



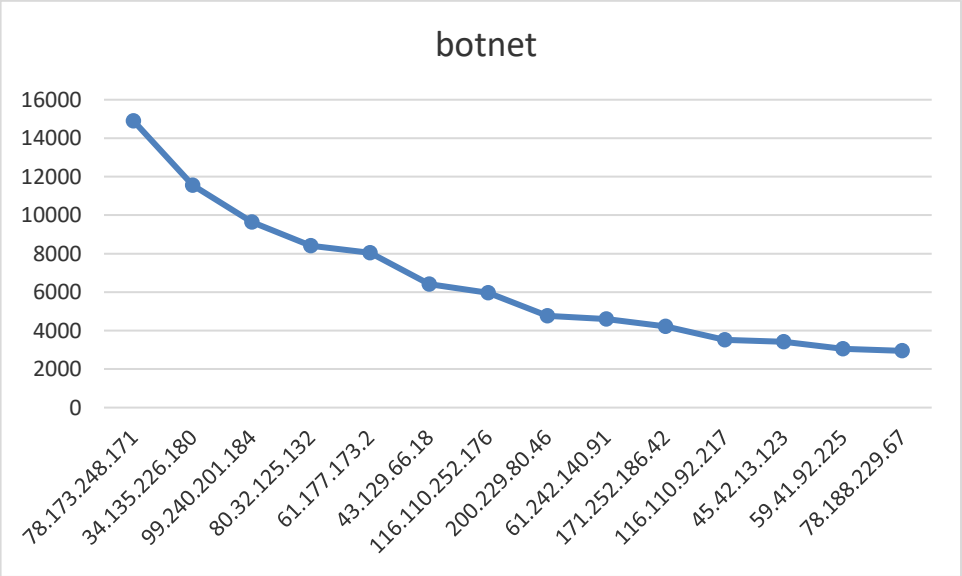
Here is the graph of time that is being spent in the system, the times on the bars are explaining the time spent in-between ex: (0.5 is equaled to 0-0.5, 1 equal to 0.5 – 1)

Figure 5: the graph of time that is being spent in the system



Here is the graph of the IP addresses that are suspected to be part of a botnet, this suspicion is based on the fact that these attacks have durations mostly that are under a single second, and they are at most 20 seconds for the whole duration of the attack.

Figure 6: The graph of the IP addresses that are suspected to be part of a botnet



Here are the data and the IP addresses of the suspected human attackers that took more than 50 seconds in the system.

Figure 7: the graph of the IP addresses that are suspected human attackers

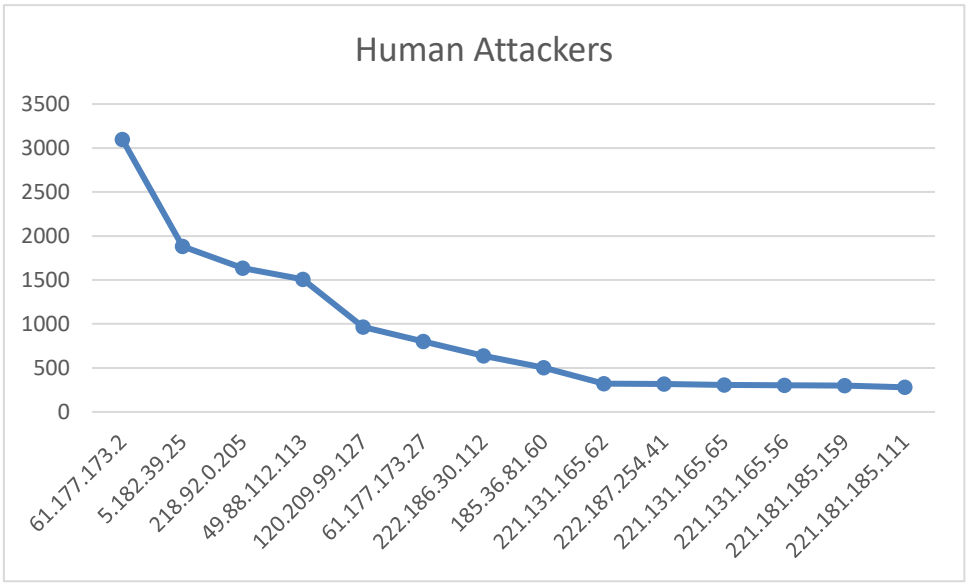


Figure 8: The graph of general IP list and the number of attacks that are launched from those IP addresses.

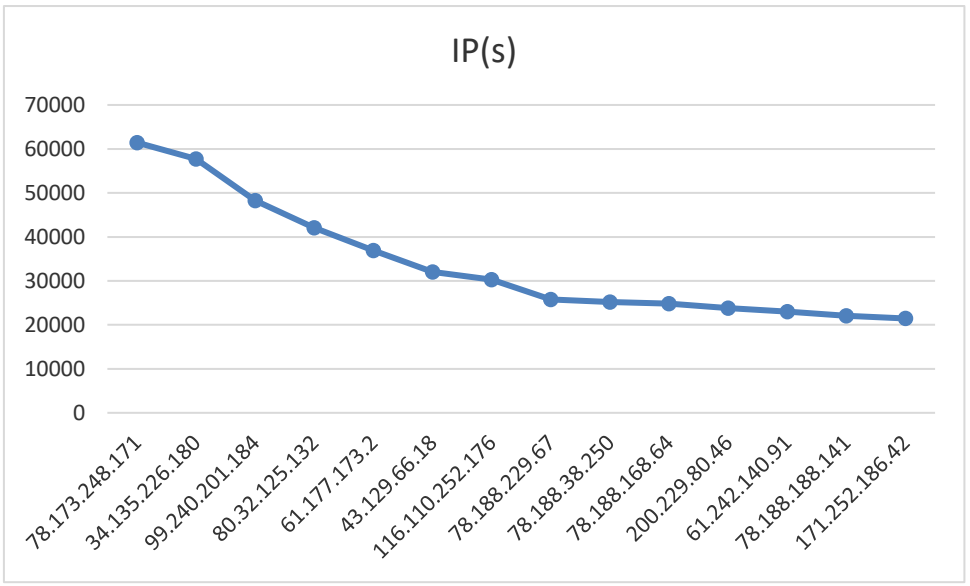
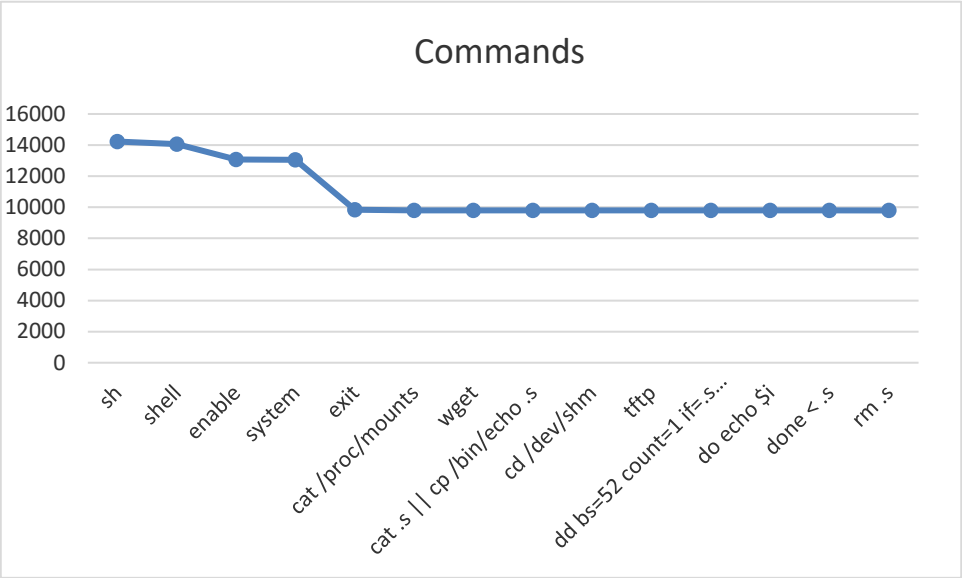


Figure 9 shows the graph of the top 15 commands that are used in these attacks. Clearly, sh was used the mostly and rm.s was used least.

Figure 9: the graph of the top 15 commands that are used in these attacks

FINAL REPORT



Because we have used a Database, and open-source software we can gather specific data for any required action with our project. It can be customized to fit needs. These are the results that we choose to extract and analyze as part of this project.

Conclusion

As for our conclusion we see that even an pointless system out in the wild (not protecting or hosting any valuable data) is subject to a lot of cyber-attacks, this means that even small companies or self-hosted anything on the internet is at an huge risk, that result was one of our hypothesis that we have proven. This research should open eyes about the volume and the inherent danger of cyber-attacks to the relevant parties. We see some interesting stuff especially about the usernames and the originating countries, note that none of these ip(s) were back traced so we do not know if they are behind vpn(s) or any other ip disguising methods, this fact was due to a limitation of resources. We expected the admin admin to be the most common username password combo but the results show differently, and some usernames are used much more frequently than others as the data shows. We also expected the attacks from turkey to be in the top 3 of this list. We are also surprised that we have received attacks from 152 countries in total. Another interesting fact that there was 4 human attempts of trying to install crypto miners to our devices. This data defying our expectations and our norms means that this will be a valuable insight in the developments of IDS and IPS systems. We have also shown that you might catch some interesting data by using a system like this and looking at the commands that people execute.

So with the data we have gathered and that data defying our expectations we have reached the primary goals of this project. To open eyes of the volume and the intensity of such attacks and to gather useful data to improve IDS and IPS systems.

Future of the project

For the next iterations of the project we might convert this to an IDS system, with some more logic and defined actions that the system will execute once it catches a certain behavior this can also be developed into an IPS system. But the best option for us would be to expand the logic of the honeypot, mirror an existing network topology at an institute and use this system as an last line of defense and an trap system, in such an environment if an attacker passes all the security measures in that institute but is caught at this system,

damage is prevented and we can gain some meaningful data by analyzing the path that they have taken to intrude in this system, and fix the security breaches in the rest of the system thus allowing us to constantly improve the security of the system while spending minimal resources once the system is deployed. As we have previously stated that such a system should be an last line of defense not something that should be in use constantly. But in an worst case situation this gives us the ability to harden so that situation never happens again and saves us with minimal damage. Also such an system would ease the job of the security researchers and cyber-sec engineers that are helping or actually working in that institution, because even in extreme cases there will be a lot of data of the path that has been followed so defensive members can move on to patch the system without spending a lot of time figuring out the how. A further bonus of such a system will allow the institute or the researchers discern the motive of the attacker by looking at what the attacker have done inside their systems.

References

1. [1] S. Russell and P. Norvig, Virtual honeypots: from botnet tracking to intrusion detection, 3rd ed. Addison-Wesley Professional, 2007.
2. ACM Code of Ethics and Professional Conduct.
3. The Software Engineering Code of Ethics, IEEE Computer Society.
4. IEEE Code of Ethics.
5. Computer and Information Ethics, Stanford Encyclopedia of Philosophy
6. https://github.com/cowrie/cowrie
7. https://cowrie.readthedocs.io/en/latest/index.html
8. https://github.com/jasonmpittman/cowrie-log-analyzer