Available online at
https://jcrinn.com/
https://crinn.conferencehunter.com/

Journal of Computing Research and Innovation 9(2) 2024

Journal of Computing Research and Innovation

Performance Analysis of Network Intrusion Detection Using T-Pot Honeypots

Mohd Faris Mohd Fuzi[1]*, Muhammad Fahimuddin Mazlan[2], Muhammad Nabil Fikri Jamaluddin[3], Iman Hazwam Abd Halim[4]

[1,2,3,4] *College of Computing, Informatics, and Mathematics, Universiti Teknologi MARA Perlis Branch, Arau Campus, 02600 Arau, Perlis, Malaysia*

ARTICLE INFO

Article history:
Received 1 July 2024
Revised 22 August 2024
Accepted 23 August 2024
Online first
Published 1 September 2024

Keywords:
Honeypot Detection
Performance Analysis
Network Intrusion
T-Pot

DOI:
10.24191/jcrinn.v9i2.477

ABSTRACT

Honeypots have become invaluable tools in the field of cybersecurity, allowing researchers to gain insights into attacker behaviour, collect data on malicious activities, and develop effective defence strategies. Traditionally, honeypots relied on rule-based approaches or signature-based detection to identify and categorise attacks. However, with the growing complexity and diversity of cyber threats, these methods often struggle to keep pace with evolving attack techniques. Modern honeypots, such as T-Pot, have become multi-faceted systems that provide researchers with a wealth of data. They could emulate different vulnerabilities and services, thus attracting a wide array of cyberattacks. This ability to simulate real-world systems and networks allowed for a detailed analysis of attack methodologies and helped to understand the evolving nature of cyber threats. As attacks became more sophisticated, so did the strategies to combat them. This included understanding the landscape of cyber threats, anticipating potential vulnerabilities, and staying ahead of the attackers. Thus, this project aims to implement a complex honeypot system with capabilities to detect and prevent cyberattacks. The project will involve designing the honeypot infrastructure, collecting data on attacks, integrating the model into the honeypot system for real-time analysis, generating reports and alerts based on the analysis, and continuously improving the system's defences. The tests revealed that honeypots can perform real cyberattacks, as well as detect and warn about threats. This project used Nmap, Hydra, and Hping3 to pretend to be attackers and show that the honeypot could fake network resources and attract them, which makes it a smart network intrusion detection system. There was a lot of experimental data on how well the honeypot could find things. Each test checked how well the honeypot could find threats on the network. In conclusion, these tests proved that the honeypot's methods for finding threats are correct, which means it can indeed find network breaches.

[1*] Corresponding author. *E-mail address*: farisfuzi@uitm.edu.my
https://doi.org/10.24191/jcrinn.v9i2.477

1. INTRODUCTION

In recent years, the use of honeypots has become increasingly popular as a means of detecting cyberattacks. According to Mudgal and Bhatia (2022), a honeypot is a system that attracts and detects unauthorised access attempts or malicious activities within a network or computer system. Regular production systems typically deploy honeypots alongside them, mimicking real services and applications. They appear to be legitimate targets for potential attackers, who may attempt to gain unauthorised access, exploit vulnerabilities, or deploy malware. An organisation deploys these honeypots within its production network to divert attackers from its real systems (Spyros et al., 2022). Honeypots can be valuable tools for cybersecurity professionals because they provide early detection, threat intelligence, diversion of attackers, and active defence (Patel et al., 2022). However, traditional honeypot systems suffered from limitations such as limited data volume and a lack of scalability. These limitations made it difficult to identify new and emerging attack patterns. The fact that typical honeypot systems depended on predetermined rules or signatures to identify attacks was a major problem. These criteria may have failed to recognize new or developing attack patterns, leading to erroneous classifications of legitimate traffic as attacks or the absence of any attacks altogether. Traditional honeypots failed to scale and may have been unable to handle much data. This made it harder for them to gather enough information to detect new or developing attack patterns. Traditional honeypots also faced visible limitations. Honeypots only tracked and logged activity when an attacker directly interacted with them.

Any attacks conducted on different areas of the system were not logged, only being recorded if the honeypot was also threatened (Tsochev et al., 2021). Furthermore, traditional honeypots often had issues with fingerprinting and discovery. This happened when an attacker figured out a honeypot's identity because of the characteristics they had. Something as minuscule as a spelling error in the emulation of a service could expose the fact that it was a honeypot (Matin, 2019). Finally, honeypots were a double-edged sword. They effectively lured attackers. This was due to the fact that attackers, once they gained access to a honeypot, could potentially find motivation to launch additional attacks. Attackers were typically relentless, and if they discovered that they had been duped, they were not likely to stop until they gained access to the real thing (Veena et al., 2023). The other disadvantage of using honeypots was that they just added to the complexity of a network's design. This meant that the additional resources incurred extra costs through maintenance. Honeypots also had to be kept up and running for them to work effectively.

The objective of this project was to develop an intelligence honeypot for network intrusion and evaluate honeypot detection performance. The research that looked at how well the T-Pot honeypot worked at stopping network intrusions focused on the design of the system and specifically looked at how honeypots can be used to stop network intrusions. This research allowed future system engineers to gain a clear vision of the honeypot issue.

The deployment of a T-Pot honeypot system was a critical step in strengthening an organization's cybersecurity defences. This project is significant because it enables network intrusion analysis through the T-Pot honeypot system, allowing for the preemptive identification of attack strategies and early detection of potential threats. With its extensive logging and monitoring capabilities, the system had been able to reveal the exploitation patterns of attackers, providing cybersecurity teams with an advantage in protecting their networks from breaches.

https://doi.org/10.24191/jcrinn.v9i2.477

2. RELATED WORKS

This chapter explores related research to the project. It presents an overview of previous studies and projects that have implemented honeypot. The section discusses the advancements, methodologies, and outcomes of these research efforts, providing valuable insights and context for the current project.

In one of the previous study, Mehta et al. (2021) used machine learning to predict directory traverser patterns during attacks using cowrie honeypot data. The authors recommend examining cowrie honeypot data to identify the directory traverser pattern, a common malicious actor approach. The cowrie honeypot simulates vulnerable services to attract and track attackers. Researchers use cowrie honeypot data to detect directory traversal threats. Security vulnerability directory traversal allows unauthorised access to files and folders outside the intended range. Data analysis predicts directory traverser tendencies, which helps improve defences against these attacks. The authors forecast the directory traverser pattern using Support Vector Machine (SVM), a machine learning algorithm. The SVM model is trained with pre-processed data, where features indicate attack characteristics. SVM can anticipate new directory traversal threats by recognising patterns and features.

Meanwhile, Baçer et al. (2021) use honeypots to target Secure Socket Shell (SSH) and Telnet protocols. The study examines SSH and Telnet honeypot data. The authors propose mimicking SSH and Telnet services with a tailored honeypot. Honeypots entice and capture intruders so researchers can examine their strategies. Researchers study SSH and Telnet honeypot data to understand attacks. The researchers evaluate the data to understand assault trends, strategies, and attributes to create effective responses. The SSH and Telnet honeypot reveals threats, vectors, and sources targeting these protocols. Researchers discover brute-force login attempts, password guessing, and vulnerability attacks in the data. Data analysis improves network security, according to the paper. Administrators can prevent assaults by learning their methods and being proactive. SSH and Telnet honeypot data can disclose protocol vulnerabilities and weaknesses, enabling better secure configurations and defence techniques. The study uses a honeypot to analyse SSH and Telnet assaults. The SSH and Telnet honeypot tracks assaults on these protocols. We analyse the data for attack trends and weaknesses. The analysis results can improve network security and help develop effective SSH and Telnet defences.

Another researcher, Zymberi, I. (2021) study examines honeypots and machine learning for malware detection. The study deployed and analysed DShield and T-Pot honeypots. Setting up honeypots in a network environment and gathering data to understand threat actors' behaviours was the goal. The study showed how T-Pot, a multi-honeypot platform, uses the Elasticsearch, Logstash, and Kibana (ELK) Stack to analyse and visualise data. This integration collected extensive external threat data, improving network security. Honeypots like T-Pot are important in cybersecurity for various reasons, as shown in the thesis. They effectively monitor and log suspicious actions, reveal attack strategies, and assist construct strong defences. T-Pot data revealed common weaknesses used by attackers and the most popular attacks, such as SSH brute force attacks. This study stressed the importance of honeypots in cybersecurity research and their function in network protection against cyber threats. Analysing honeypot efficacy against network intrusions shows how honeypots like T-Pot may help understand and mitigate cybersecurity threats.

The following studies is from Matin et al. (2019) describes a method for collecting and analysing Portable Executable (PE) malware. The research focuses on Windows, which is vulnerable to virus attacks. The authors propose using the Modern Honey Network honeypot and machine learning methods to collect and analyse PE malware. The Modern Honey Network efficiently collects malware samples. However, it cannot determine the malware file format, which is crucial for Windows-based malware research. Researchers use the Modern Honey Network to collect malware samples. Researchers extract features from these samples to determine the PE file format. Virus Total is used to classify malware. The analysis

https://doi.org/10.24191/jcrinn.v9i2.477

identified 1,222 malware samples, of which 945 were PE malware. In terms of the SVM module, the cowrie honeypot differs from the Modern Honey Network structure in the study. Both honeypots attract and catch criminals. The researchers may have classified malware using SVM. Often used for classification, SVM can accurately distinguish malware by examining extracted properties. The research suggests using the Modern Honey Network honeypot to collect malware samples and SVM to categorise an analyse PE malware.

Lastly, the study by Nursetyo et al. (2019) examines the effectiveness of honeypots against network intrusions, particularly the T-Pot honeypot. Honeypots on servers might establish a shadow server and prevent attackers from accessing the genuine server, according to the study. This strategy is essential for configuring and using honeypots like T-Pot to improve network security. The study discusses Medusa for brute-force assaults and Kippo for attacker logs. Network security analysis and intrusion detection use honeypots, which offer crucial insights into attacker behavior and aid in the creation of robust defense strategies against various cyber threats. Modern cybersecurity methods use honeypots to strengthen networks against brute-force attacks, according to the report.

3. METHODOLOGY

3.1 Design and development phase

During the design and development phase, the performance analysis of honeypot detection against network intrusion using TPOT-honeypot was discussed. Initially, we designed the TPOT-honeypot, incorporating all the necessary equipment and mechanisms for project development. Attackers attempted to breach the honeypot using three types of attacks: Nmap, Brute Force, and Distributed Denial-of-Service (DDoS), utilising the Parrot operating system for its convenience in accessing required tools, such as Nmap in aggressive mode. The attackers chose Parrot operating system, known for its straightforward user interface, for its ease of controlling the operating system and ran it on a DHCP server. Fig. 1 show the design of T-POT honeypot attack.

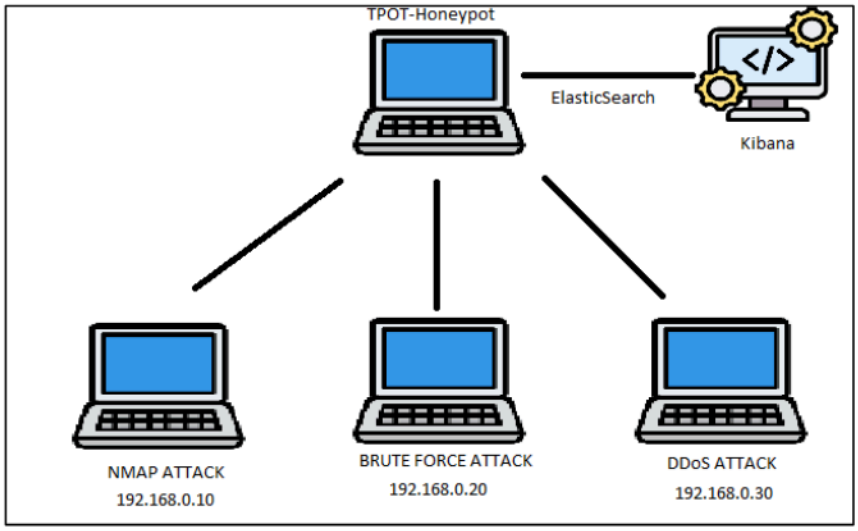

Fig. 1. Design of T-POT honeypot attack

https://doi.org/10.24191/jcrinn.v9i2.477

In this research, Debian 11 was used to operate the TPOT-honeypot. Because the attacker needed to handle multiple DDoS attacks, the TPOT-honeypot was deployed in a Kali Linux environment. Kali Linux was selected for its ability to protect privacy and leave no evidence of use that could result in an information breach. For the study, the TPOT-honeypot collected attacker data such as IP address, SSH port, internet protocol, and timestamp. The Elasticsearch module was then employed to translate the attacks into logs. Elasticsearch was chosen for its wide range of features, including powerful built-in functions like data rollups and index lifecycle management, efficient data storage and search capabilities, and integration with Beats and Logstash for data processing. Additionally, Kibana provided real-time visualisation of Elasticsearch data and quick access to application performance monitoring, logs, and infrastructure metrics data. Fig. 2 show the T-POT environment after it successfully been installed and setup.

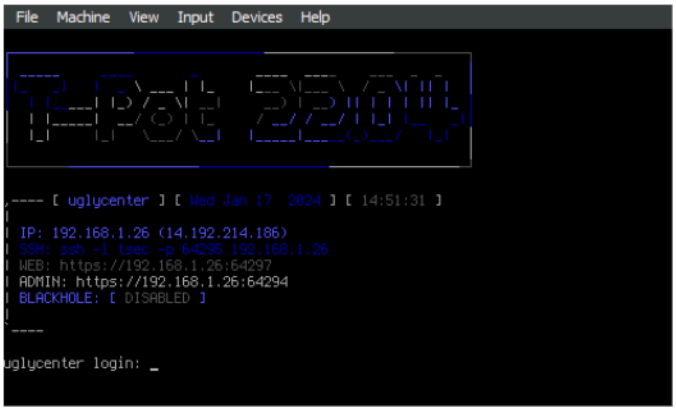

Fig. 2. T-Pot environment

3.2 Experimentation phase

In this project, several experiments were conducted during the performance analysis phase of honeypot detection against network intrusion using TPOT-honeypot. The format for the experiments consisted of two main types: common attack ports and honeypot alerts. This phase was meticulously designed to simulate real-world scenarios and attack patterns, offering valuable insights into the system's performance under varied and potentially challenging conditions. The focus extended beyond merely testing a system; it was an exploration of the frontiers of attack patterns in cybersecurity, delving into how attacks could be executed and breach security measures.

A critical component, the test phase, evaluated a network's security posture by simulating a variety of cyber-attack scenarios. This phase unfolded through a series of carefully structured experiments, each tailored to probe different aspects of network security. The initial experiment harnessed the capabilities of Nmap, a robust network exploration tool, to conduct reconnaissance of the network's open ports. This scenario, known as "Common Attack Port," simulated the initial stage of an intrusion, where potential vulnerabilities were mapped out. Subsequently, the "Alert from Honeypot" scenario was designed to test the network's detection capabilities and its ability to alert administrators of attempted breaches.

The second set of tests utilised Hydra, a powerful brute-force tool, to simulate attack attempts on common network ports. This mirrored the actions of an adversary attempting to gain unauthorised access by guessing credentials. The accompanying scenario was again checked for the honeypot's effectiveness in identifying and reporting the intrusion attempts. Lastly, the third experiment involved a DDoS attack

https://doi.org/10.24191/jcrinn.v9i2.477

simulation conducted with hping3, pushing the network to its limits by flooding it with traffic from random sources. This test aimed to mimic a widespread attack, challenging the network's ability to maintain service continuity under heavy-load conditions. In both scenarios of this experiment, the honeypot's response to such high-volume threats was also observed.

3.2.1 Experiment one: Test with Nmap attack.

To execute our scenario, we require certain tools to run the script. Nmap, short for Network Mapper, is a powerful and widely used network discovery and security auditing tool. Network administrators, security professionals, and enthusiasts use it for various tasks related to network mapping, management, and security analysis. The sudo nmap -A 172.20.10.11 command focuses on conducting an in-depth network scan to evaluate the security and configuration of a specific network target, identified by the IP address 172.20.10.11. Fig. 3 shows the Nmap attack commands.

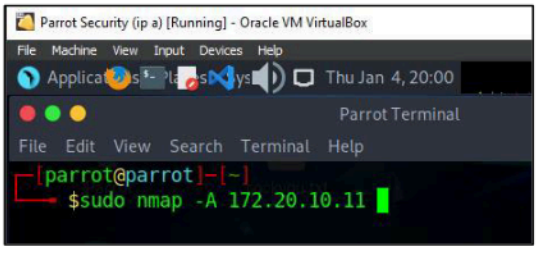

Fig. 3. Nmap command

By leveraging Nmap, a powerful and versatile tool in cybersecurity, this project aims to uncover critical insights into network security. The -A flag in the command signifies an aggressive scan, combining several advanced features. This includes Operating System (OS) detection, which identifies the target's operating system; version detection, which determines the versions of services operating on open ports; script scanning, which executes various scripts to gather detailed information and potentially identify vulnerabilities; and traceroute, which maps the path data takes to reach the target. This comprehensive approach provides a holistic view of the network target's security posture.

3.2.2 Experiment two: Test attack with Brute force attack.

In the second experimental setup of the investigation, the project implemented a brute-force attack simulation. It utilised the Hydra tool, targeting a server with the IP address 192.168.1.109. The purpose of this simulation was to mimic an authentic attack scenario where the attacker does not have prior knowledge of valid user credentials. For this simulation, the rockyou.txt file, which is well-known for its extensive compilation of real-world passwords previously exposed in various data breaches, was employed as a source for both usernames and passwords. This approach is reflective of a typical method attackers use to identify weak credentials. By employing the rockyou.txt file as a dictionary in their brute-force attack, the researchers used the rockyou.txt file as a dictionary to replicate an attacker's technique to exploit vulnerable credentials and assess the server's security measures against such intrusions. Fig. 4 shows the Brute Force commands.

Fig. 4. Brute force command

https://doi.org/10.24191/jcrinn.v9i2.477

3.2.3 Experiment three: Test attack honeypot using DDoS Attack.

In this section of the study, this project employed a tool known as hping3, which can generate and dispatch a multitude of network packets. They used this tool with specific parameters, namely --flood and --rand-source, to simulate an overwhelming flow of messages directed at the IP address 192.168.1.109. It is important to note that this IP was configured to receive Dynamic Host Configuration Protocol (DHCP) assignments, allowing it to alter rapidly. The --flood option ensured a continuous stream of messages, providing an opportunity to evaluate the network's resilience to sustained high traffic volumes. Meanwhile, the --rand-source parameter randomized the origin of each message, effectively creating the illusion of a distributed source. This simulation was critical in assessing the network's capability to handle a multitude of requests that seemingly originate from a diverse array of locations. Such an approach was instrumental in testing the network's defences against distributed denial of service (DDoS) attacks, wherein the network is bombarded with requests that appear to be coming from numerous distinct sources. Fig. 5 shows the DDoS attack command.

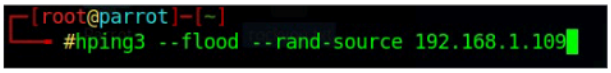

Fig. 5. DDoS attack command

3.3 Analysis phase

The TPOT-honeypot project followed a systematic approach during the analysis phase to process and extract insights from the gathered data (Kristyanto et al., 2022). Initially, the collected data, which included logs and recorded attacker interactions, was stored in a structured format like Comma-Separated Values (CSV) files, databases, or Elasticsearch indexes to facilitate easy access and manipulation for further analysis. Then Kibana was used to display the stored data, which improved the efficiency of data analysis and manipulation. Various pre-processing and feature engineering techniques have been applied to the loaded data throughout this phase. This process included cleaning the data, addressing missing values, converting categorical variables, and selecting pertinent features for analysis.

4. RESULT AND DISCUSSION

This section examines the gathered data using various methods to uncover trends, connections, and key observations. These results, which emerge from the analysis, are critical for forming the final conclusions and recommendations.

4.1 Comparison for Common Attack Port

Comparing common attack ports across various attack methodologies, such as Nmap scanning, brute force attacks, and DDoS attacks, provides insightful perspectives into potential vulnerabilities. Each attack type tends to target different sets of ports based on their objectives. For instance, people often use Nmap, a network mapping tool, to find open ports and services. This tool scans a wide range of ports to identify which ones are listening, making it a comprehensive approach for identifying potential entry points into a network.

Brute force attacks, on the other hand, often focus on specific ports associated with services that require authentication, such as SSH (port 22) and FTP (port 21). Brute force scenarios frequently target

https://doi.org/10.24191/jcrinn.v9i2.477

these ports as the attackers repeatedly try different username and password combinations to gain unauthorised access.

DDoS attacks, which overwhelm a target with excessive traffic, typically focus on ports associated with high-profile and heavily used services. For example, HTTP (port 80) and HTTPS (port 443) are common targets as they are standard ports for web traffic. The goal of DDoS attacks is often to disrupt services rather than gain unauthorised access, hence the focus on these widely used ports. Comparing the commonly attacked ports across these methods highlights the varying intentions behind different types of network attacks. Fig. 6 shows the comparison of the targeted port.

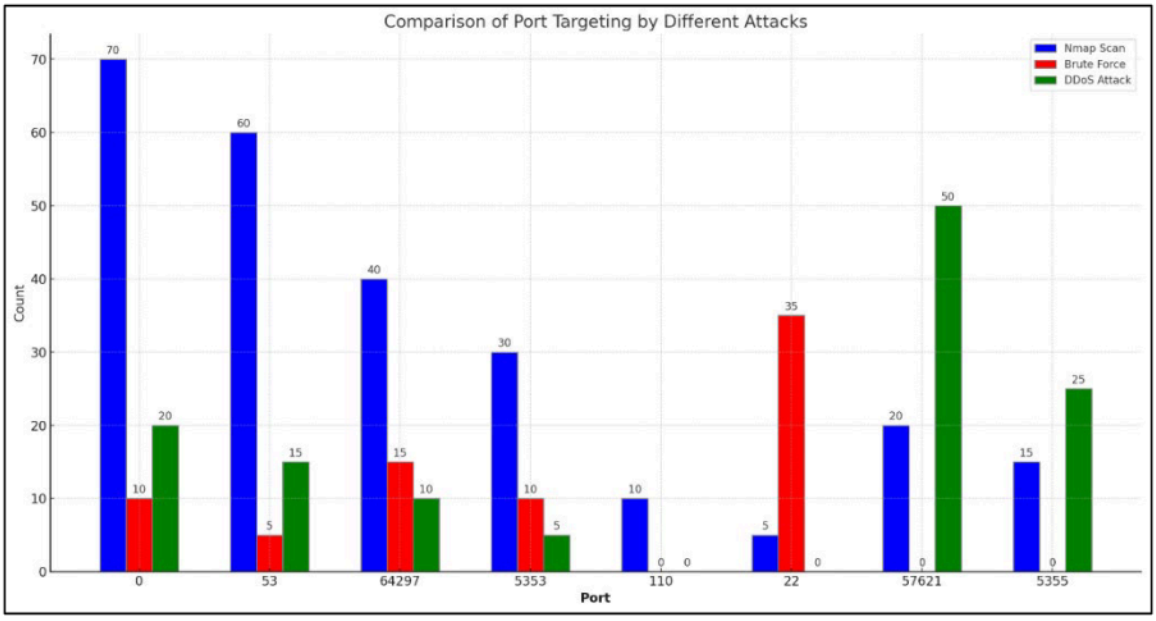

Fig. 6. Comparison of targeted port

Port 0 is significantly more targeted by Nmap scans compared to brute force attacks, a pattern that warrants a closer examination. The prominence of port 0 in Nmap scan activities could be attributed to various factors. Primarily, it is considered a part of enumeration processes, where Nmap is utilized to probe systems for responses to unconventional queries, given that port 0 is officially a reserved port and not standardly assigned to any service. Attackers might include port 0 in their scans to investigate system behaviours under atypical conditions or to detect systems with non-standard configurations. Additionally, the inherent behaviour of certain versions of Nmap could automatically involve port 0 in scans, either as part of default settings or due to specific user configurations. Conversely, the minimal focus on port 0 by brute force attacks is explicable given the nature of such attacks, which typically concentrate on ports associated with active services vulnerable to exploitation.

Nmap is often used to probe port 53 to gather information about the DNS services of a target system. This is because DNS services can reveal significant information about a network, such as the domain names and associated IP addresses, which can be valuable for both legitimate network administrators and malicious actors. Scanning port 53 can help identify DNS servers, which can be a vector for DNS amplification attacks, a type of DDoS attack. Brute force attacks, on the other hand, are less likely to target port 53. Brute force typically involves attempting to guess the credentials to gain unauthorized access to a

https://doi.org/10.24191/jcrinn.v9i2.477

system. Since DNS services do not generally require authentication in the same way that, for instance, an SSH or FTP service would, they are not commonly targeted for brute force credential attacks.

Port 64297 is being scanned a lot by Nmap but hardly ever by brute force attacks. This difference demonstrates that scanning and trying to break into systems are done in different ways. Nmap is a tool used to check networks for safety reasons, and it looks closely at port 64297. This may be because this port is rarely used and has unusual services that need to be checked for safety or strange behavior. On the other hand, brute force attacks, which try to guess passwords to get into systems, don't happen much on port 64297. Attackers usually go after well-known ports that have important services like SSH on port 22 or RDP on port 3389, where they have a better chance of getting in.

Port 5353 appears to be predominantly targeted by Nmap scans, whereas DDoS attacks on this port are minimal. This discrepancy is likely rooted in the distinctive purposes served by Nmap and the strategic objectives underlying DDoS attacks. Port 5353 is typically associated with multicast DNS (mDNS) services, which are utilised for service discovery within local networks. The increased attention from Nmap scans could suggest an effort by network administrators or security analysts to identify devices leveraging mDNS, a protocol integral to the automatic detection of services on a local network. Such scanning might be part of a broader security assessment to pinpoint potential vulnerabilities within devices that employ mDNS or as a segment of an extensive network mapping operation. Lack of DDoS attacks on port 5353 can be explained by the inherent characteristics of mDNS. It is a protocol largely confined to local network use and does not generally require external internet access. Consequently, disrupting mDNS would not typically yield the level of impact that DDoS attackers seek, as their goal is often to impair services with a significant online presence or operational criticality. DDoS campaigns tend to concentrate on ports that, when compromised, can cause extensive service disruption or notable economic harm, such as the common web service ports 80 and 443, or the global DNS service port 53.

Port 110 stands out as it is frequently scanned by Nmap but not targeted by any brute force or DDoS attacks. This port is commonly associated with the Post Office Protocol version 3 (POP3), used for fetching emails from a server. The reason Nmap scans this port often could be because network professionals are looking to identify and check email servers. They might want to make sure these servers don’t have any weak spots that could be taken advantage of, or they could be checking that the email service is working right and isn’t using any old, less secure methods. On the other side, the data doesn’t show brute force or DDoS attacks on port 110, which might be because many email services now use more secure ways to get emails that don’t involve port 110. Attackers might also prefer to go after parts of the network that can give them more valuable information or control, like web services or databases. Plus, trying to guess email passwords through brute force isn’t as easy now because of security steps like locking accounts after too many wrong tries. As for DDoS attacks, which are meant to shut down services, hitting port 110 might not cause enough trouble compared to other services that are more critical to online activities. It’s also possible that such attacks just weren't happening during the time the data was collected or that they weren't noticed.

Port 22 is primarily targeted by brute force attacks, with comparatively fewer scans from Nmap and no recorded Distributed Denial of Service (DDoS) attacks. Port 22 is commonly used for Secure Shell (SSH) connections, which are encrypted links meant for secure communication and managing servers remotely. Attackers often use brute force methods on port 22 to try to guess login details and gain unauthorised access to systems, making this port a frequent target for such attacks. Nmap, a tool used for mapping network services, appears to scan port 22 less often in this data. This might mean that in the network being observed, there's less need to check SSH services or maybe a greater focus on finding other services. As for DDoS attacks, these aim to overload services and make them unavailable. However, port 22 isn't typically used to serve large numbers of users directly, like a web service does, so shutting it down

https://doi.org/10.24191/jcrinn.v9i2.477

might not have a wide-reaching impact. This could explain why there are no DDoS attacks on port 22 in this dataset.

Port 57621 is most attacked by DDoS attacks, with very few Nmap scans and no brute force attacks observed. This pattern could suggest that port 57621 is important for the network's operation, making it a likely target for DDoS attacks, which aim to shut down services by overwhelming them with traffic. The goal is to disrupt the service, affecting the users who rely on it. The reason behind the low number of Nmap scans might be that this port is not usually used for services that need regular checking for weak spots or isn't seen as a high-risk area on the network. Nmap scans are typically done to find services that could be vulnerable, so a lower number of scans on port 57621 might mean it's not a common place for such services or it's not considered a big security concern. The complete absence of brute force attacks could mean that port 57621 doesn't have a service that needs a username and password to get in. Brute force attacks try to guess these login details to break into systems, so if there's nothing to log into on this port, attackers wouldn't bother with brute force methods here.

Port 5355 is predominantly targeted by DDoS attacks while receiving minimal attention from Nmap scans and no recorded brute force attacks. Port 5355 is typically used by the Link-Local Multicast Name Resolution (LLMNR) protocol, which can be used to resolve the names of networked devices in scenarios where DNS is not available. The high frequency of DDoS attacks on port 5355 might be because disrupting LLMNR could cause significant network disruptions, particularly in local networks where devices rely on this protocol for name resolution. DDoS attacks generally aim to flood a service with excessive traffic to make it unavailable and targeting port 5355 could be part of a strategy to disrupt network communication within a targeted environment. The relatively low number of Nmap scans on port 5355 may indicate that in the given network, this port is not commonly used for running services that are crucial to the network's infrastructure or that it is not a common vector for vulnerabilities, thus requiring less frequent security checks. The absence of brute force attacks on port 5355 supports the idea that this port is not typically associated with services that require authentication, such as a login interface.

4.2 Comparison for Alert from Honeypot

In an Nmap scan, which is typically a network mapping and security auditing process, the alerts are relatively lower in count across all categories. Similar to a routine check-up, this scan evaluates the network for open ports and vulnerabilities, much like a maintenance inspection. It's not inherently aggressive, which explains the lower frequency of alerts compared to actual attacks.

When it comes to brute force attacks, there is a noticeable increase in certain types of alerts, especially those indicating attempts to gain unauthorised administrative privileges. These attacks are the digital equivalent of trying every possible key on a lock until one fit; they are persistent and focused, which is why the graph shows a higher concentration of these specific alerts. The goal is to gain high-level access, and the corresponding alerts reflect this concentrated effort.

The DDoS attack, on the other hand, shows a significant rise in alerts that signify potentially hazardous traffic. Similar to a flood, this type of attack overwhelms the network with a high volume of requests, disrupting normal operations. The surge in such alerts is characteristic of the DDoS attack's attempt to render network services unavailable. Fig. 7 shows the comparison of alert categories.

https://doi.org/10.24191/jcrinn.v9i2.477

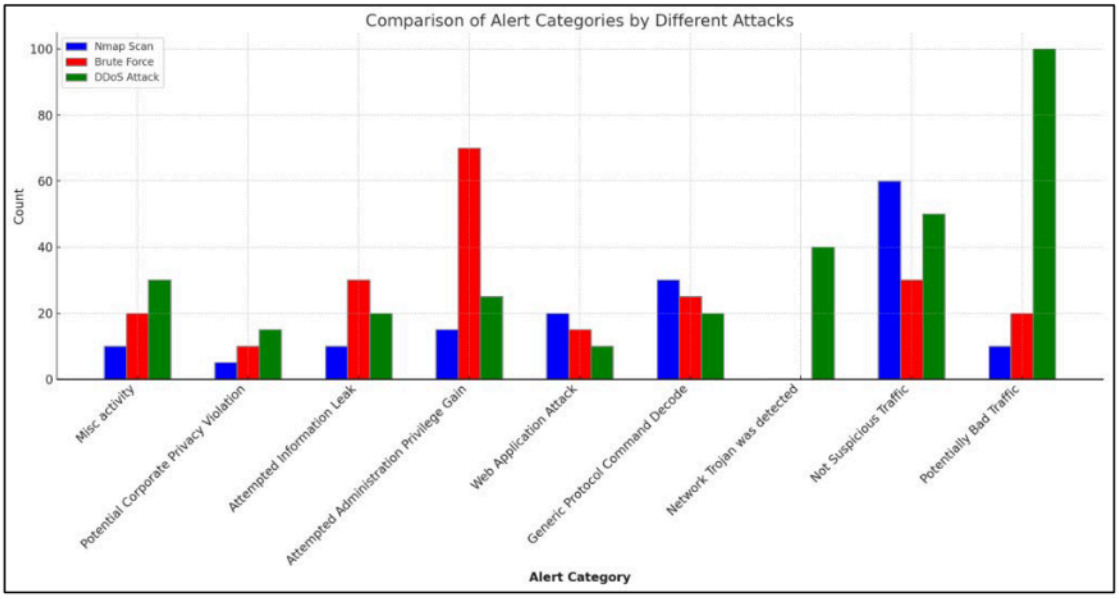

Fig. 7. Comparison of alert categories

5. CONCLUSION AND RECOMMENDATIONS

This project reflects on the comprehensive assessment of network security through honeypot engagement. The experiments conducted provided invaluable insights into the effectiveness of honeypots in simulating real-world cyber-attack scenarios, as well as their ability to detect and alert to threats. This project has successfully met its intended goals. The first objective was to develop an intelligent honeypot for network intrusion. Using tools like Nmap, Hydra, and Hping3 to simulate different attack scenarios, it was shown that the honeypot can effectively look like network resources and attract potential intruders, making it an intelligent system for finding network intrusions. The second objective focused on evaluating the accuracy of honeypot detection. The experiments conducted provided a substantial amount of data regarding the honeypot's detection capabilities. Each test was designed to challenge the honeypot's ability to detect and differentiate between different types of network threats. The results from these experiments have shown a high degree of accuracy in the honeypot's detection mechanisms, affirming its potential as a reliable tool for identifying network intrusions.

Throughout the project's development, we encountered several limitations that impacted the research process. Additionally, the complexity of the Kibana dashboard within T-Pot presented a steep learning curve. Its intricate features, while powerful, required a level of technical expertise that was challenging to achieve within the project's timeframe. This complexity sometimes hindered efficient monitoring and analysis of the honeypot system's collected data. Lastly, T-Pot's inability to directly identify the types of attacks it simulated was a notable limitation. While T-Pot excelled at collecting data on network intrusions, the lack of immediate identification meant that a more in-depth analysis was required to classify the nature of each attack, adding another layer of complexity to the research process.

In conclusion, we suggest several key upgrades to make this project more effective and comprehensive in the future. Making the most of T-Pot's features, enabling it to handle more complex attack scenarios

https://doi.org/10.24191/jcrinn.v9i2.477

currently limited by the existing computer's lower power, is necessary. The Kibana dashboard, which is part of T-Pot, is quite detailed and complex. A better understanding of Kibana would aid in monitoring and analysing the honeypot's data more efficiently. Adding advanced analysis tools or machine learning to T-Pot is another area to explore. This could allow the system to identify the types of cyberattacks as they happen, making T-Pot better at detecting threats. With these improvements, the project setup would be much stronger, leading to more insightful cybersecurity research. These steps would also create a solid base for any future work in this field, expanding the possibilities for protecting networks against cyberattacks.

6. ACKNOWLEDGEMENTS/FUNDING

The author hereby acknowledges the financial support from Universiti Teknologi MARA (UiTM) under 600-RMC 5/3/GPM (058/2022) and College of Computing, Informatics and Mathematics UiTM Cawangan Perlis for providing the facilities for this research.

7. CONFLICT OF INTEREST STATEMENT

The authors agree that this research was conducted in the absence of any self-benefits, commercial or financial conflicts and declare the absence of conflicting interests with the funders.

8. AUTHORS CONTRIBUTION

Author 1 carried out the research, supervised the research progress, and wrote and revised the article. Author 2 designed and developed the honeypot project. Authors 3 and 4 reviewed the experiment and discussed the results and analysis. All authors contributed to the final version of the manuscript.

9. REFERENCES

Baçer, M., Güven, E. Y., & Aydin, M. A. (2021). SSH and Telnet protocols attack analysis using honeypot technique. In *Proceedings - 6th International Conference on Computer Science and Engineering, UBMK 2021* (pp. 806–811). https://doi.org/10.1109/UBMK52708.2021.9558948

Kristyanto, M. A., Krisnahati, I., Rawung, F., Dzhalila, D., Nurwibawa, B. D., Murti, W., Adi Pratomo, B., & Shiddiqi, A. M. (2022). SSH bruteforce attack classification using machine learning. In *2022 10th International Conference on Information and Communication Technology (ICoICT 2022)* (pp. 116–119). IEEE Xplore. https://doi.org/10.1109/ICoICT55009.2022.9914864

Matin, I. M. M. & Rahardjo, B. (2019). Malware detection using honeypot and machine learning. In *7th International Conference on Cyber and IT Service Management (CITSM)* (pp. 1-4). IEEE Xplore. https://doi.org/10.1109/CITSM47753.2019.8965419

Mehta, S., Pawade, D., Nayyar, Y., Siddavatam, I., Tiwart, A., & Dalvi, A. (2021). Cowrie honeypot data analysis and predicting the directory traverser pattern during the attack. In *Proceedings of the 2021 IEEE International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES 2021)* (1-4). IEEE Xplore. https://doi.org/10.1109/ICSES52305.2021.9633881

Mudgal, A., & Bhatia, S. (2022). A step towards improvement in classical honeypot security system. In

2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON 2022) (pp. 720–725). IEEE Xplore. https://doi.org/10.1109/COM-IT-CON54601.2022.9850502

Nursetyo, A., Setiadi, D. R. I. M., Rachmawanto, E. & Sari, A. (2019). Website and network security techniques against brute force attacks using honeypot. In *2019 Fourth International Conference on Informatics and Computing (ICIC)* (pp. 1-6). IEEE Xplore. https://doi.org/10.1109/ICIC47613.2019.8985686

Patel, P., Dalvi, A., & Sidddavatam, I. (2022). Exploiting honeypot for cryptojacking: The other side of the story of honeypot deployment. In 2022 6th International Conference on Computing, Communication, Control and Automation (ICCUBEA 2022) (pp. 1-5). IEEE Xplore. https://doi.org/10.1109/ICCUBEA54992.2022.10010904

Spyros, A., Papoutsis, A., Koritsas, I., Mengidis, N., Iliou, C., Kavallieros, D., Tsikrika, T., Vrochidis, S., & Kompatsiaris, I. (2022). Towards continuous enrichment of cyber threat intelligence: A Study on a honeypot dataset. In *Proceedings of the 2022 IEEE International Conference on Cyber Security and Resilience (CSR 2022)* (pp. 267–272). IEEE Xplore. https://doi.org/10.1109/CSR54599.2022.9850295

Tsochev, G., Sharabov, M., & Georgiev, A. (2021). Using machine learning reacted with honeypot systems for securing network. In *Proceedings International Conference Automatics and Informatics (ICAI 2021)* (pp. 425–428). IEEE Xplore. https://doi.org/10.1109/ICAI52893.2021.9639590

Veena, K., Meena, K., M, M. T., C, H., & Rajalakshmi, D. (2023). An advanced intrusion detection solution for networks based on honeypot servers. In *2023 International Conference on Inventive Computation Technologies (ICICT)* (pp. 1217–1222). IEEE Xplore. https://doi.org/10.1109/ICICT57646.2023.10134511

Zymberi, I. (2021). *Honeypots: A means of sensitizing awareness of cybersecurity concerns* [Bachelor thesis - Information and Communication Technology, Metropolia University of Applied Sciences]. https://urn.fi/URN:NBN:fi:amk-202105016540

https://doi.org/10.24191/jcrinn.v9i2.477