Implementation of Honeynet and Honeypot in Network Infrastructure in Production Network

Nawshad Ahmed Evan
School of Computing and Creative Technologies
University of the West of England
Bristol BS16 1QY, United Kingdom
nawshadevan@gmail.com

Md Raihan Uddin
Department of Electrical and Computer Engineering
The University of Alabama in Huntsville
Huntsville, AL 35899, USA
mu0016@uah.edu

***Abstract*—Network infrastructure in a production environment is increasingly targeted by attackers every day. Many resources and services now rely on the internet, making network infrastructure one of the most critical parts to protect, as it hosts numerous company resources and services. Several solutions have already been proposed to prevent attacks, minimize damage, and divert hackers and intruders. Among these, the honeypot stands out as a highly effective tool; it is designed to mimic both a scanner and an attacker, diverting and misleading them within a simulated, production-level environment. This paper will demonstrate the use of a honeynet where a honeypot acts like a real resource to deceive the attacker and analyze their behavior.**

***Index Terms*—Honeynet, honeypot, network security, production network.**

I. Introduction

Security is a top priority in both social and corporate life. To maintain strong security, it is essential to understand data protection techniques and perform regular security checks [1]. The world is becoming more challenging in terms of security, making a safe environment a primary focus for individuals and companies. As technology advances, each new day brings fresh security challenges [2].

In a production environment, network security is a significant challenge. Protecting the network is one of the most important factors, whether it's operating over the internet, a LAN, or other methods, and this is true for any size of business. While no system is completely immune to attacks, a strong and effective security system is essential for safeguarding customer data. A good security system helps businesses reduce the risk of data theft and sabotage [3]. Digitally connected devices and applications are becoming a part of every aspect of our lives—homes, offices, cars, and even our bodies. All of these gadgets are becoming "smarter" to take advantage of being connected to the internet.

The Internet of Things (IoT) is growing rapidly, which has expanded the attack surface far beyond traditional enterprise IT infrastructures. It is important to understand the security risks of IoT before discussing its features [4]. This study explains how Honeypots and Honeynets can be used to increase the security of a network's infrastructure.

Honeypots are designed to improve security by detecting unauthorized attempts to access a data system. Modern network security often includes firewalls, intrusion detection systems, and encryption. However, today's environment requires more proactive methods to identify, redirect, and contain unwanted access. Honeypots offer a proactive approach to these network security concerns [5]. Honeypots are typically hosted on servers that simulate different environments to appear like a real network. An attacker can enter the honeynet from any machine, which allows the system to monitor the attacker's behavior and divert threats away from the real network. A honeynet simulates a genuine network more accurately than a single honeypot, making it suitable for large or complex environments where attackers might be drawn to what seems like an authentic and appealing target [6].

All network infrastructure resources—including hardware, software, systems, and devices—work together to enable connectivity, control, and communication. This ranges from servers to Wi-Fi routers. This infrastructure also includes the network protocols that allow users and systems to communicate and interact [7].

This study will propose a secure, hybrid-designed model for network infrastructure security to provide trusted communication and reliable service. The main challenge is to integrate honeypots and honeynets with firewalls and analyze hacker footprints to improve security. We will use two types of honeypots: one will be placed in front of the honeynet's production router to act as a fake web server, tricking attackers into spending their time there. The other will be placed behind the router, where it can analyze attacker behavior in a controlled environment. The honeynet is set up before the main network infrastructure to ensure security and continuous service in a production environment.

II. Related Works

New methods and technologies are needed for network security and forensics. Honeypots are widely used for network security, as these tools can track intruders and monitor hacker activity. Modern communication has connected the world and made the internet accessible to everyone. To maintain security, it is crucial to continuously develop new strategies and upgrade firewalls and intrusion detection systems [8]. Honeypot security solutions are constantly being improved and used in new ways. Even traditional decoy honeypots have

many uses for security engineers and can sometimes help predict security vulnerabilities.

Cloud services and their user bases are growing rapidly. Companies are now more comfortable hosting their servers in the cloud to serve their clients. While internet and app services have made our lives easier, this expansion also increases vulnerability. To protect these services, Honeypot and Honeynet technologies must be improved by closely monitoring and analyzing attacker activity [9].

This paper will implement a technique to create a Honeynet and Honeypot. This method is designed to deceive hackers with fake content, diverting them from the real network and potentially protecting it. However, if a hacker carefully examines the material and detects the deception, they might erase their traces before we can collect them and could even plan a larger attack. Poorly maintained or misused honeypots could also increase risks from hackers [10].

Understanding hacker attack patterns is essential for network engineers and researchers. Engineers regularly review security reports and logs to measure these patterns, which is a key part of developing and maintaining security. By comparing an attacker's habits and attack stages, researchers have noticed that attacks on weekdays can have different outcomes than those on weekends. Although some days stand out, attack patterns can vary. By studying these patterns, experts can better understand an attacker's behavior and adjust the network's security architecture to defend against it. Hackers can attack from outside the network or by compromising an internal user's computer. Honeypots and Honeynets help security analysts by recording both external and internal threats and vulnerabilities [11].

An employee's lack of security awareness can make a company vulnerable. Employees with low awareness might accidentally reveal company secrets. Therefore, security awareness training is essential for network security. Many companies are now training their employees on security to help them manage risks and vulnerabilities. This training often focuses on basic communication, technical skills, and how to recognize threats like phishing and ransomware [12].

Hackers attack individuals and corporations to steal data, often aiming to steal money or hold data for ransom in exchange for money or cryptocurrencies. Financial firms and organizations that hold public data are major targets, but this does not mean small businesses are immune. Hackers will also attack any network that appears insecure. As a result, security must be a top priority for any organization, regardless of its size.

In security research, a honeypot is a valuable tool for carefully examining hacker activity. It allows security experts to see how an attacker penetrates a system, elevates their privileges, and moves through the network. Security organizations, research institutions, and government agencies are all assessing these threats to find solutions. With a honeypot, an attacker's actions can be examined in a secure environment, either physical or virtual, preventing a breach of the actual network [13].

III. Proposed Approach

This section explains our methodology. We will introduce the topic, present the data we collected, analyze it, and then validate our findings to show the project's reliability. The goal of this project is to implement a solution for real-time threats to network infrastructure. The analysis and project discussion will be illustrated with a diagram to clarify our method and structure.

This paper will create a Honeypot within a Honeynet. A honeynet is a network where all devices are decoys but are designed to function like real ones. A honeypot is a single fake server or workstation that imitates a real resource [14]. Active protection is a relatively new concept in information technology. While some have tried to define this term, many definitions are incomplete or miss key features of this security approach. Some have blended active and offensive security strategies.

This study examines the features of active protection techniques like Honeypots and Honeynets, including their benefits and limitations [15]. Honeypot systems have been around for over a decade and have improved and adapted over time, thanks to programs like The Honeynet Project and Project Honeypot.

Despite their benefits, honeypots are not commonly used in businesses. This may be due to the challenges of installing and managing a honeypot, as well as a lack of understanding of its advantages. Compared to traditional firewalls, honeypots and honeynets offer advanced security features and provide valuable support for security engineers [16].

A. Network Security Tactics and Types

The foundation of network security management is the CIA triad: Confidentiality, Integrity, and Availability. Many network security tools are ineffective without a good strategy and proper administration. To manage a network without interruption and meet user needs, it's important to handle fault management, configuration, and performance analysis of the security infrastructure [17].

Confidentiality protects sensitive data from unauthorized users and attackers. Access to this data should be limited to authorized individuals. **Integrity** ensures that data is not altered by unauthorized people, maintaining its accuracy for providers. **Availability** ensures that authorized users can access the data when they need it. Any interruption that prevents users from accessing the service is considered a security violation. The service must be secure and consistently accessible to the right users [18].

Security methods can be categorized based on the component level: hardware, software, and cloud. **Hardware** is the physical part of a network or connected device. It is necessary to safeguard hardware from physical and digital attacks, as a network attack can damage hardware by overusing its resources and shortening its lifespan. **Software** is the graphical interface that provides services. A direct software infection can damage the hardware or the entire network, so software security must always be kept up-to-date. **Cloud**

services are hosted on the internet and can be used from anywhere. It's crucial to protect this blended hardware and software environment in data centers using the CIA approach [19].

B. Conceptual Framework

Network security engineers are always focused on preventive measures to protect their network. Security surveillance is needed to prevent network breakdowns, unauthorized access, and malicious users. Securing a commercial network often requires expensive hardware and software. A security framework for a network is typically based on its core design [20].

Since this project uses open-source components, its security approach was designed to be adaptable to any network topology. Engineers choose hardware based on their network design. This article discusses our specific network design and compares it to standard and hybrid security models.

We will use the GNS3 open-source simulator to build a decoy network topology where some devices will behave like real ones. We will configure network protocols and connect them to the internet with a real routing engine to make the simulation more realistic. This network will be placed next to the main production network's border router. A Windows-based Honeypot server will be configured behind the Honeynet router to analyze attacker activities and store logs. This honeypot will have defense and traceback systems to capture attacker footprints, trace them back to their source, and analyze them.

Tracing the attacker's IP will help reveal their location, hop count, and route path. The connection state of each hop will be presented in milliseconds (ms) for measurement. We will also check the source IP's domain information for identification [21].

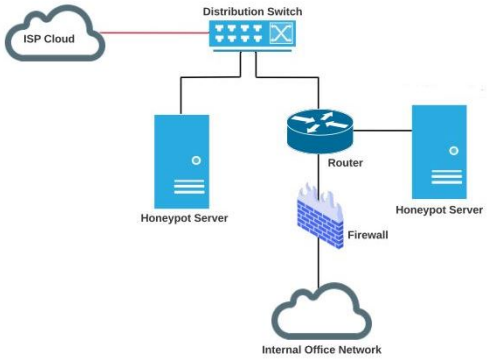


Fig. 1. Hybrid network security infrastructure.

An attacker will first try to find and enter the production network. To counter this, another honeypot will be set up in front of the Honeynet network router to deceive the attacker. It will present a fake message to make it seem like a legitimate web server. The goal is to make the attacker think they are in a real service environment, which will distract them on the fake web server. This decoy device is placed before the Honeynet router to entice hackers. However, if poorly designed, attracting hackers into a fake network can be dangerous. A comprehensive Honeynet is therefore designed to prevent hackers from gaining access to more devices if they become more aggressive [10].

Next, an attacker machine will be placed in the network topology to generate random attacks using Kali Linux tools to evaluate the system. The logs will then be analyzed to understand the attacker's activity. This project offers a design that traps attackers, making it difficult for them to distinguish between real and fake networks. By searching the honeypot for resources, the attacker wastes time and leaves behind valuable footprints and logs, which helps us understand their attack patterns. This can reduce vulnerabilities and protect the real network infrastructure. The second honeypot will capture the attacker's source to determine their network origin and system details, identifying any other potential vulnerabilities.

C. Comparison Between Regular and Hybrid Design

Cyber attacks are one of the biggest criminal risks today, and security engineers are constantly taking steps to prevent them. Security breaches often happen due to a lack of awareness or insufficient precautions. Security measures rely on hardware, software, and cloud components to be effective [22].

Different network security models use different tools. Most models include a firewall or a similar security device. Many security engineers believe a firewall is the most cost-effective solution for network security. A firewall offers services like intrusion detection and prevention, threat analysis, security logs, and more. The main goal of a firewall is to protect a network and provide uninterrupted service [23].

However, a firewall alone cannot protect a network with critical services. Attackers today can bypass firewalls using methods like SQL injection, social engineering, application vulnerabilities, and exploiting IoT devices or employee ignorance. Once a firewall is breached, hackers can quickly infiltrate networks and access resources [24].

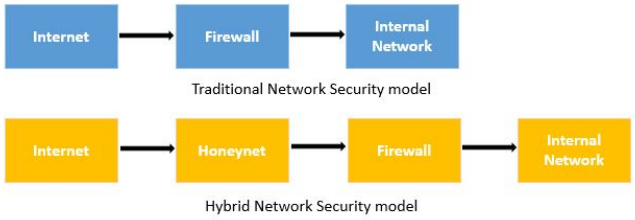


Fig. 2. Comparison of traditional and hybrid secure network designs.

In addition to a firewall, other security control methods are needed. While a firewall is a crucial component, this traditional approach is well-known to everyone, including hackers.

The network model developed here uses a different security approach. As shown in the diagram, it features a three-layer security system: a decoy device that misleads the attacker, a fake network that manipulates them, and another decoy device that analyzes and tracks them. These three layers are placed before the firewall that protects the production network. This

unique approach provides security engineers with extra reports on hackers trying to target network devices, giving them time to study logs, analyze threats, and prepare for an attack.

D. Instrumentation and Simulation

In this paper, the goal is to develop a secure network topology with two different Honeypots, each serving a different purpose. We used the GNS3 simulator for our lab to visualize the project in a virtual environment [25].

The Graphical Network Simulator-3 (GNS3) is a free, open-source network simulation platform. It provides a graphical interface where complex network labs can be created by importing various network equipment like routers, switches, firewalls, and servers. It works by emulating the images of the devices that do the actual job. Using this emulator, anyone can design a high-quality, complex network topology, including simulations of Ethernet, frame relay, and ATM switches. Virtual machines from VMWare or VirtualBox can be imported using the GNS3 VM. GNS3 has a built-in feature for packet capture using Wireshark. The main drawback of GNS3 is that while the platform itself doesn't use many base machine CPU resources, the lab machines (routers, switches, firewalls) installed in GNS3 do consume CPU and RAM according to their allocated memory [26].

Our lab was built with seven components, including a cloud, a distribution switch, and another switch from the GNS3 simulator. We also used a Cisco IOS router, a Pentbox Honeypot, a KFSensor Honeypot, and a Firewall in our topology. The Pentbox and KFSensor honeypots were run in VMWare, and the virtual machines were imported into the GNS3 environment as VMWare templates.

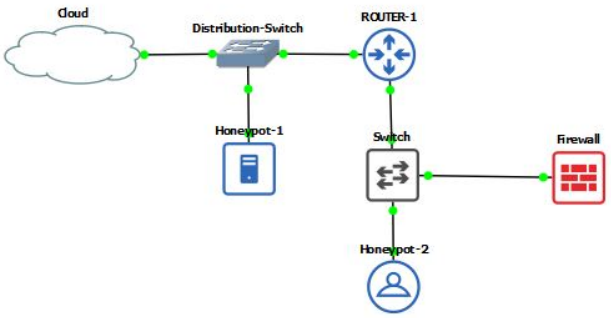


Fig. 3. GNS3 Topology diagram.

The entire lab was run in the GNS VM, with both GNS3 and GNS VM at version 2.2.5. The GNS VM was allocated 10 GB of RAM and 2 dual-core processors. Using the GNS VM is better than the standalone GNS3, especially when running Cisco IOS or Unix devices. Once installed, the GNS VM is simple to use as a plug-and-play OS. It can also be accessed remotely as a cloud by setting up internet connectivity [27].

We ran two of our honeypots in VMWare and imported them as templates into the GNS3 VM environment. One honeypot was installed on Ubuntu, and the other on a Windows operating system.

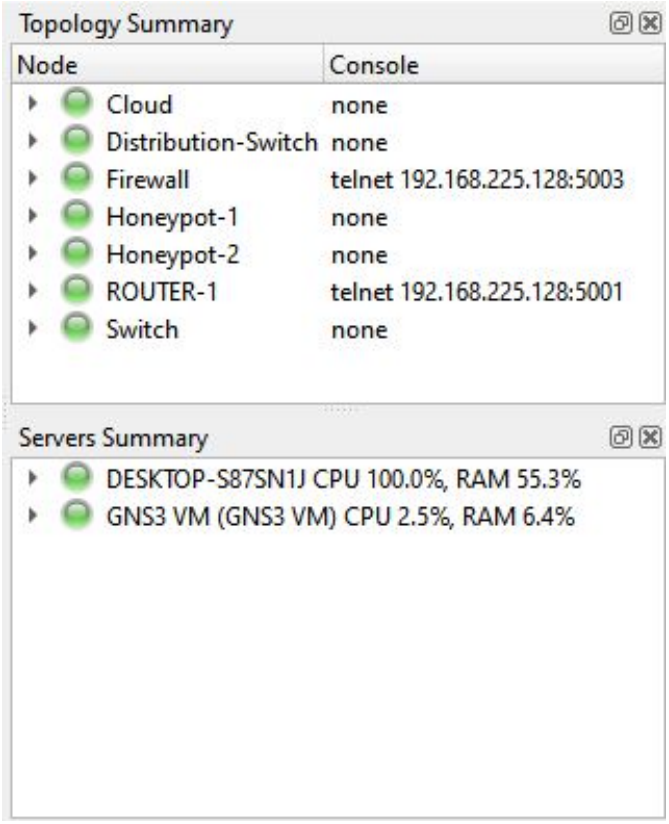


Fig. 4. GNS3 topology and servers summary.

IV. Data Collection and Analysis

The Ubuntu-based Pentbox honeypot was used to mimic an attacker's target. We lured the attacker with a fake banner on a web server using this honeypot, causing them to leave behind a footprint. This data is useful for forensic analysis of the attacker. Pentbox is a penetration testing tool with several features for penetration testing, network analysis, and monitoring [28].

The Pentbox was set up as a web server with a fake message, and port 8080 was opened to fool the attacker. This honeypot serves as the first decoy in front of the network. This open-source program can be cloned from GitHub for learning purposes [29]. This honeypot is a good network decoy that can gather an attacker's footprints before they can assault the main network.

The router connects and distributes the internet to all connected devices, routing IP packets between them. It translates network addresses and sends local network traffic to the wide area network and vice versa. Routers also have security features like access control lists, port filtering, and IP filtering. Although routers do not provide deep inspection services like firewalls, some can offer content filtering and bandwidth control [30].

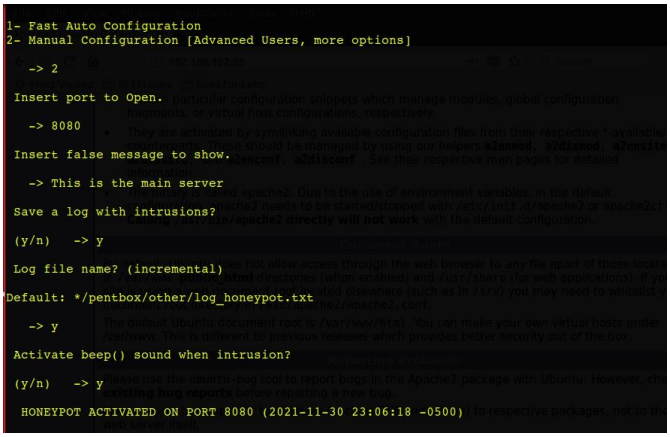
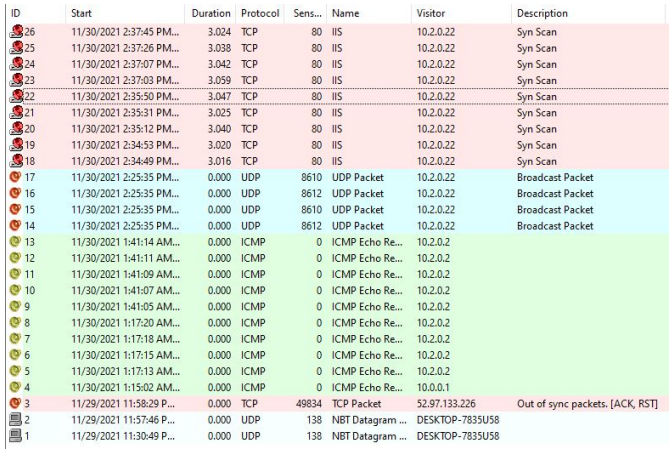


Fig. 5. Pentbox configuration data as a Honeypot.



ID	Start	Duration	Protocol	Sens...	Name	Visitor	Description
26	11/30/2021 2:37:45 PM...	3.024	TCP	80	IIS	10.2.0.22	Syn Scan
25	11/30/2021 2:37:26 PM...	3.038	TCP	80	IIS	10.2.0.22	Syn Scan
24	11/30/2021 2:37:07 PM...	3.042	TCP	80	IIS	10.2.0.22	Syn Scan
23	11/30/2021 2:37:03 PM...	3.059	TCP	80	IIS	10.2.0.22	Syn Scan
22	11/30/2021 2:35:50 PM...	3.047	TCP	80	IIS	10.2.0.22	Syn Scan
21	11/30/2021 2:35:31 PM...	3.025	TCP	80	IIS	10.2.0.22	Syn Scan
20	11/30/2021 2:35:12 PM...	3.040	TCP	80	IIS	10.2.0.22	Syn Scan
19	11/30/2021 2:34:53 PM...	3.020	TCP	80	IIS	10.2.0.22	Syn Scan
18	11/30/2021 2:34:49 PM...	3.016	TCP	80	IIS	10.2.0.22	Syn Scan
17	11/30/2021 2:25:35 PM...	0.000	UDP	8610	UDP Packet	10.2.0.22	Broadcast Packet
16	11/30/2021 2:25:35 PM...	0.000	UDP	8612	UDP Packet	10.2.0.22	Broadcast Packet
15	11/30/2021 2:25:35 PM...	0.000	UDP	8610	UDP Packet	10.2.0.22	Broadcast Packet
14	11/30/2021 2:25:35 PM...	0.000	UDP	8612	UDP Packet	10.2.0.22	Broadcast Packet
13	11/30/2021 1:41:14 AM...	0.000	ICMP	0	ICMP Echo Re...	10.2.0.2	
12	11/30/2021 1:41:11 AM...	0.000	ICMP	0	ICMP Echo Re...	10.2.0.2	
11	11/30/2021 1:41:09 AM...	0.000	ICMP	0	ICMP Echo Re...	10.2.0.2	
10	11/30/2021 1:41:07 AM...	0.000	ICMP	0	ICMP Echo Re...	10.2.0.2	
9	11/30/2021 1:41:05 AM...	0.000	ICMP	0	ICMP Echo Re...	10.2.0.2	
8	11/30/2021 1:17:20 AM...	0.000	ICMP	0	ICMP Echo Re...	10.2.0.2	
7	11/30/2021 1:17:18 AM...	0.000	ICMP	0	ICMP Echo Re...	10.2.0.2	
6	11/30/2021 1:17:15 AM...	0.000	ICMP	0	ICMP Echo Re...	10.2.0.2	
5	11/30/2021 1:17:13 AM...	0.000	ICMP	0	ICMP Echo Re...	10.2.0.2	
4	11/30/2021 1:15:02 AM...	0.000	ICMP	0	ICMP Echo Re...	10.0.0.1	
3	11/29/2021 11:58:29 P...	0.000	TCP	49834	TCP Packet	52.97.133.226	Out of sync packets. [ACK, RST]
2	11/29/2021 11:57:46 P...	0.000	UDP	138	NBT Datagram ...	DESKTOP-7835U58	
1	11/29/2021 11:30:49 P...	0.000	UDP	138	NBT Datagram ...	DESKTOP-7835U58	

Fig. 7. Analyzing intruder behavior using KFSensor.

Our routing device handled LAN-to-WAN routing and acted as the network's border gateway. Border routers manage Border Gateway Protocol (BGP) and internet traffic. With our CISCO IOS 7200 border router, we can filter traffic from specific sources and destinations. We can also set port restrictions for security. The border router can filter IP addresses, access control lists, and ports to prevent network attacks, allowing us to block any suspicious or vulnerable addresses. We used the CISCO 7200 border router because it provides these features and serves as a reliable routing system with security surveillance capabilities [31], [32].

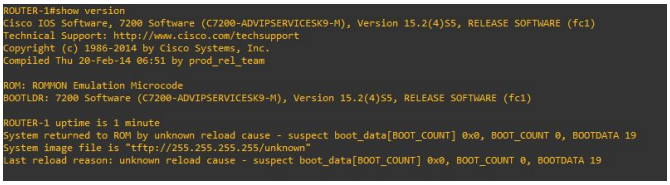


Fig. 6. CISCO IOS Router version.

KFSensor is an intrusion detection system for Windows that can be used as a honeypot in a network security model. It is known as an active intrusion detection tool and can identify attackers moving on a network and store logs to analyze their motives. KFSensor provides an interface where anyone can view a real-time threat report with detailed information [32].

In this research, we used KFSensor as a honeypot to analyze attacker behavior and attack patterns. KFSensor provides information about the attacker's source, protocol, traffic, packet type, and more [33]. I generated threats from a testing computer and observed the results on its real-time threat interface. This interface showed TCP, UDP, and ICMP threats from other machines. TCP and UDP packets came from the testing attacker's workstation, while ICMP packets came from the router. We found broadcast UDP packets and a 'sync scan' TCP, which is caused by a Distributed Denial of Service (DDoS) attack from another machine on the network.

In this lab, we generated a DDoS attack from an attacker workstation to analyze the data collected by KFSensor. We used Slowloris in Kali Linux for the DDoS attack. Slowloris is an open-source tool available on GitHub that generates many HTTP session requests to a target location for an HTTP DDoS attack. The targeted system becomes unable to respond to other users due to the excessive HTTP requests, causing service interruptions [**?**]. Slowloris generated a DDoS attack by sending keep-alive header packets.

We sent a DDoS attack to the honeypot to test how we could collect data for analyzing the attacker's patterns. This DDoS attack drains service resources such as bandwidth, CPU, and memory, causing service interruptions [34]. The Slowloris tool sent a large number of HTTP requests to our honeypot to verify that it captures the attacker's data in real time. We measured the attack's start and end times in the event description. The figure shows that the attack was a form of sniffing, which we measured. After reviewing multiple reports on the time frame and action type, we can determine the attacker's timing and patterns, including when they choose to launch a new attack.

When more information about the attacker is needed, we can use the KFSensor Nmap feature to trace them. We tracked the path from the attacker's IP information using Nmap and scanned it to get information about the operating system, system version, connectivity status, and packet travel time through a trace report [33]. With all this information, we measured the distance between our system and the attacker's system and gathered details about the materials used to generate the attack. Knowing the operating system and version helps us understand how dangerous the next attack could be.

To find the attacker's location on a global map, we used the Nmap topology to get the targeted IP address. The Nmap tool built into KFSensor can locate the vulnerable IP address by following the trace route step-by-step and provides graphical feedback about the attacker's hop location [33]. At this point, it is clear that we have successfully back-traced the attacker using the KFSensor Honeypot. We used KFSensor Nmap to scan and capture information about the attacker's system. With Nmap in KFSensor, we scanned every active IP address, performed a full network scan, identified network vulnerabilities,

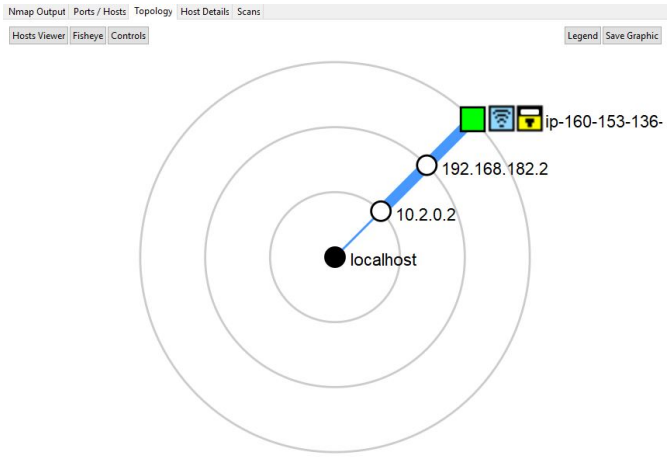


Fig. 8. KFSensor Nmap topology finder.

and developed a visual map of the attacker's location, open ports, and other key information [33].

V. Result and Performance Analysis

The designed network security diagram set a benchmark, and we tested its performance. The goal was to reduce network vulnerability and challenge attackers by analyzing and observing the collected reports. Our first approach was to create a fake environment to trap the attacker. The first honeypot we set up was designed to mimic a fake web server on port 8080 to trick the intruder. The server would display a fake message upon entry.

The second honeypot was designed to attract the intruder to get their footprints and analyze scan reports. This device provides a brief report that allows us to identify and observe the intruder's attitude and attack patterns.

The performance was analyzed in a real-time attack scenario. We connected all devices to the internet to make them reachable globally and observed how attackers were more likely to find and attack our devices.

The result was that the first honeypot received the highest number of vulnerability scanning reports. This is because this honeypot is at the front of the network, and its information can be easily found by scanning the network surface. We analyzed this report using the built-in Wireshark on the Ubuntu operating system of that honeypot. Wireshark is a network vulnerability analysis tool available on most popular operating systems, and it can capture packets and traffic patterns going in and out of a specific machine.

We observed that Honeypot 2, which was used for analyzing intruders, received fewer hits from footprints and vulnerability scans from the internet. This was due to the Honeynet router, which has its own defense system against flooding and Denial of Service attacks to keep the routing engine running smoothly. Even though we did not set up any access control lists or port filtering, the router has a built-in protection system to prevent attacks from taking down its core operations.

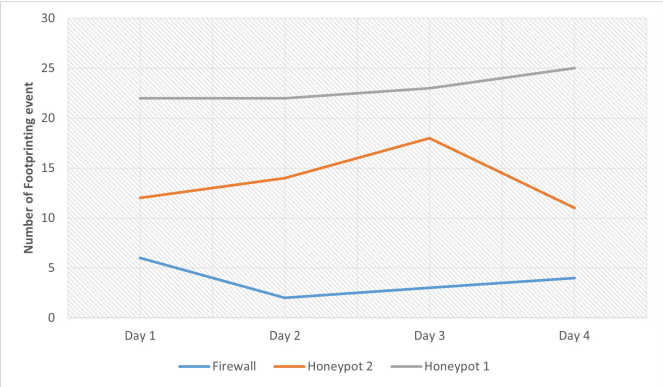


Fig. 9. Footprint events on Honeynet devices and Firewall.

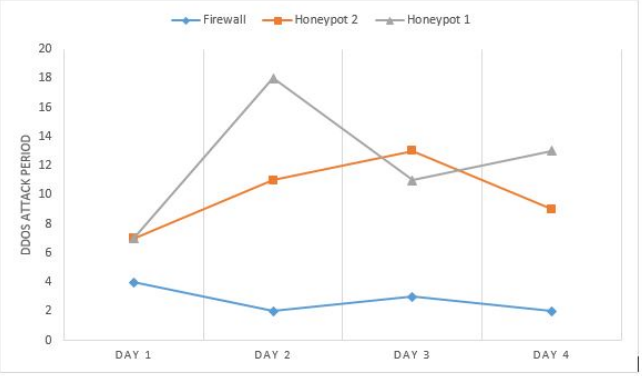


Fig. 10. DDoS events on Honeynet devices and Firewall.

While connected to the global internet, we discovered another major attack on our devices: a Distributed Denial of Service (DDoS) attack. Similar to the footprinting events, we found that Honeynet-1 received most of the attack events, while the Honeynet and the Firewall received the fewest. This attack caused higher bandwidth consumption and CPU usage on each device. The firewall has its own mechanism to protect against DoS attacks. In general, DDoS attacks are carried out by botnets that send a huge amount of traffic or service requests to the target, causing service interruptions by overwhelming the system's processes.

TABLE I
Result of Foot Printing on Each Device from WAN.

	Firewall	Honeypot 2	Honeypot 1
Day 1	6	12	22
Day 2	2	14	22
Day 3	3	18	23
Day 4	4	11	25

By observing the scan and footprint data, it is clear that the first honeypot attracted the most attention from attackers, while the second honeypot received less, as it was inside the Honeynet. The Firewall attracted the least attention because attackers had already found access to two other devices before reaching it.

TABLE II
REPORT OF DISTRIBUTED DENIAL OF SERVICE ATTACKS ON EACH DEVICE FROM WAN.

	Honeypot 1	Honeypot 2	Firewall
Day 1	7	7	4
Day 2	18	11	2
Day 3	11	13	3
Day 4	13	9	2

The DDoS attack results were similar, suggesting that attackers performed some footprinting on the network before launching the attack. Generally, attackers gather information before an attack, searching for opportunities and weak points. This is the first phase of hacking any system.

It is likely that attackers spent more time on the honeypots, which is why they invested less time on the firewall. This gave security engineers enough time to analyze the risk by studying the data from the honeypots. The next step would be to apply filters or create limitations on the firewall. Keeping an eye on the attacker's source and the ports they attack can also help security engineers prepare.

The goal of this work was to mislead attackers with a decoy and analyze their behavior before an attack. The results demonstrated the project's success, as most of the major attacks were directed at the honeypots, and hackers remained on the Honeynet for a long time instead of the real infrastructure. Additionally, we received a detailed report on their attack methods, which helped us decide on the next stage of security. The hybrid security model showed that these decoys gave security engineers enough time to prepare before any security breach or service interruption. By tracing these reports, any security engineer can easily determine the next steps for improving defense. Tracing the attacker's steps is highly recommended for protecting a network before experiencing a cyber attack.

VI. FUTURE WORK

Honeypots and Honeynets are among the most powerful tools for security research, and this study describes some aspects of a security project using them. However, the author intends to continue researching network security and a hybrid network architecture model to improve network infrastructure and security. Further study on this paper may include the following topics. The honeypots should be enhanced to resemble a real server with several services running in different VMs. The Honeynet will use a clustered, load-balanced network design with multiple routers for different internet service providers, merging internet traffic effectively. We will move beyond DDoS attacks to integrate intrusion protection into the honeypot along with an intrusion detection system. The Honeynet infrastructure will be automated to integrate with the main network infrastructure. This study discusses Honeynet and Honeypot solutions with practical examples. The author plans to add more sophisticated network scenarios and honeypot servers in future work.

VII. CONCLUSION

Network security plays a vital role in protecting confidential data and important services from theft and can save any company from cyber robbery. By focusing on this subject, this paper provided a practical concept with a real-life attack scenario on network security. The solution showed how a honeypot can be a strong security tool by providing different services. When we need to misguide a hacker or collect their footprints from a safe system, a honeypot is undoubtedly one of the best solutions. The Honeynet was presented as one of the most effective tools to protect a network, supplementing firewall security by using honeypot devices. Overall, this project discussed the importance of network security with a technique for implementing Honeypot and Honeynet services in a production network.

REFERENCES

[1] S. Rathore, P. K. Sharma, V. Loia, Y.-S. Jeong, and J. H. Park, "Social network security: Issues, challenges, threats, and solutions," *Information sciences*, vol. 421, pp. 43–69, 2017.

[2] H. Lin, Z. Yan, Y. Chen, and L. Zhang, "A survey on network security-related data collection technologies," *IEEe Access*, vol. 6, pp. 18 345–18 365, 2018.

[3] Y. E. Sagduyu, Y. Shi, and T. Erpek, "Iot network security from the perspective of adversarial deep learning," in *2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 2019, pp. 1–9.

[4] O. Mozhaev, H. Kuchuk, N. Kuchuk, M. Mozhaev, and M. Lohvynenko, "Multiservice network security metric," in *2017 2nd International Conference on Advanced Information and Communication Technologies (AICT)*. IEEE, 2017, pp. 133–136.

[5] I. M. M. Matin and B. Rahardjo, "Malware detection using honeypot and machine learning," in *2019 7th international conference on cyber and IT service management (CITSM)*, vol. 7. IEEE, 2019, pp. 1–4.

[6] J. Ren, C. Zhang, and Q. Hao, "A theoretical method to evaluate honeynet potency," *Future Generation Computer Systems*, vol. 116, pp. 76–85, 2021.

[7] J. Fox, A. Donnellan, and L. Doumen, "The deployment of an iot network infrastructure, as a localised regional service," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*. IEEE, 2019, pp. 319–324.

[8] A. F. Sheikh and A. F. Sheikh, "Network fundamentals and infrastructure security," *CompTIA Security+ Certification Study Guide: Network Security Essentials*, pp. 9–34, 2020.

[9] M. Baykara and R. Das, "A novel honeypot based security approach for real-time intrusion detection and prevention systems," *Journal of Information Security and Applications*, vol. 41, pp. 103–116, 2018.

[10] S. Krishnaveni, S. Prabakaran, and S. Sivamohan, "A survey on honeypot and honeynet systems for intrusion detection in cloud environment," *Journal of Computational and Theoretical Nanoscience*, vol. 15, no. 9-10, pp. 2949–2953, 2018.

[11] G. Leech, S. Garfinkel, M. Yagudin, A. Briand, and A. Zhuravlev, "Ten hard problems in artificial intelligence we must get right," *arXiv preprint arXiv:2402.04464*, 2024.

[12] F. Malecki, "Best practices for preventing and recovering from a ransomware attack," *Computer Fraud & Security*, vol. 2019, no. 3, pp. 8–10, 2019.

[13] A. Kyriakou and N. Sklavos, "Container-based honeypot deployment for the analysis of malicious activity," in *2018 Global Information Infrastructure and Networking Symposium (GIIS)*. IEEE, 2018, pp. 1–4.

[14] W. Fan, Z. Du, M. Smith-Creasey, and D. Fernandez, "Honeydoc: an efficient honeypot architecture enabling all-round design," *IEEE journal on selected areas in communications*, vol. 37, no. 3, pp. 683–697, 2019.

[15] C. Wu, L. Wu, J. Liu, and Z.-P. Jiang, "Active defense-based resilient sliding mode control under denial-of-service attacks," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 237–249, 2019.

[16] S. Dowling, M. Schukat, and H. Melvin, "A zigbee honeypot to assess iot cyberattack behaviour," in *2017 28th Irish signals and systems conference (ISSC)*. IEEE, 2017, pp. 1–6.

[17] M. Aminzade, "Confidentiality, integrity and availability–finding a balanced it framework," *Network Security*, vol. 2018, no. 5, pp. 9–11, 2018.

[18] L. Yin, B. Fang, Y. Guo, Z. Sun, and Z. Tian, "Hierarchically defining internet of things security: From cia to caca," *International Journal of Distributed Sensor Networks*, vol. 16, no. 1, p. 1550147719899374, 2020.

[19] A. T. Atieh, "Assuring the optimum security level for network, physical and cloud infrastructure," *ScienceOpen Preprints*, 2021.

[20] N. Saigushev, U. Mikhailova, O. Vedeneeva, and A. Tsaran, "Information systems at enterprise. design of secure network of enterprise," in *Journal of Physics: Conference Series*, vol. 1015, no. 4. IOP Publishing, 2018, p. 042054.

[21] A. Y. Nur and M. E. Tozal, "Record route ip traceback: Combating dos attacks and the variants," *Computers & Security*, vol. 72, pp. 13–25, 2018.

[22] S. Tirumala, M. R. Valluri, and G. Babu, "A survey on cybersecurity awareness concerns, practices and conceptual measures," in *2019 International Conference on Computer Communication and Informatics (ICCCI)*. IEEE, 2019, pp. 1–6.

[23] S. Jingyao, S. Chandel, Y. Yunnan, Z. Jingji, and Z. Zhipeng, "Securing a network: how effective using firewalls and vpns are?" in *Future of Information and Communication Conference*. Springer, 2019, pp. 1050–1068.

[24] R. Wildenauer, K. Leidl, and M. Schramm, "Hacking an optics manufacturing machine: You don't see it coming?!" in *Sixth European Seminar on Precision Optics Manufacturing*, vol. 11171. SPIE, 2019, pp. 35–40.

[25] D. E. Kurniawan, H. Arif, N. Nelmiawati, A. H. Tohari, and M. Fani, "Implementation and analysis ipsec-vpn on cisco asa firewall using gns3 network simulator," in *Journal of Physics: Conference Series*, vol. 1175, no. 1. IOP Publishing, 2019, p. 012031.

[26] J.-I. Castillo-Velázquez and A. Delgado-Villegas, "Gns3 limitations when emulating connectivity and management for backbone networks: a case study of canarie," in *2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, 2020, pp. 1–4.

[27] V. Kazak, D. Shevchuk, L. Panchuk, and V. Shulevka, "Methods and tools for evaluating the accuracy of the air navigation using gns," in *2018 IEEE 5th International Conference on Methods and Systems of Navigation and Motion Control (MSNMC)*. IEEE, 2018, pp. 79–82.

[28] P. Wang and H. D'Cruze, "Honeypots and knowledge for network defense," *Issues in Information Systems*, vol. 22, no. 3, pp. 241–254, 2021.

[29] S. Ravji and M. Ali, "Integrated intrusion detection and prevention system with honeypot in cloud computing," in *2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE)*. IEEE, 2018, pp. 95–100.

[30] P. Szewczyk and R. Macdonald, "Broadband router security: History, challenges and future implications," 2017.

[31] F. Khattak, P. Ginzboorg, V. Niemi, and J.-E. Ekberg, "Role of border router in 6lowpan security," in *Workshop on Smart Object Security*.

[32] A. Cui, J. Kataria, and S. J. Stolfo, "Killing the myth of cisco {IOS} diversity: Recent advances in reliable shellcode design," in *5th USENIX Workshop on Offensive Technologies (WOOT 11)*, 2011.

[33] N. Naik, P. Jenkins, R. Cooke, and L. Yang, "Honeypots that bite back: A fuzzy technique for identifying and inhibiting fingerprinting attacks on low interaction honeypots," in *2018 IEEE International Conference on fuzzy systems (FUZZ-IEEE)*. IEEE, 2018, pp. 1–8.

[34] T. Shorey, D. Subbaiah, A. Goyal, A. Sakxena, and A. K. Mishra, "Performance comparison and analysis of slowloris, goldeneye and xerxes ddos attack tools," in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2018, pp. 318–322.