**PROJECT REPORT  
ON**

**DIGITAL FORTRESS: CRAFTING AN INVINCIBLE HONEYPOT NETWORK WITH DISCORD ALERTS FOR CYBER SECURITY VIGILANCE**

**BY**

**Ali Rehman Class No. 12**

**Wajahat Hussain Class No. 24**

**Atif Zeb Class No. 139**

**SESSION: 2020-2024**

*A project report submitted to The University of Agriculture Peshawar. In The partial*

*Fulfillment of the Requirement for the degree of*

**BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY**



**SUPERVISED BY**

**MR. KASHIF ALI**

**INSTITUTE OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY**

**FACULTY OF MANAGEMENT AND COMPUTER SCIENCES**

**THE UNIVERSITY OF AGRICULTURE, PESHAWAR**

**KHYBER PAKHTUNKHWA-PAKISTAN**

**AUGUST, 2024**

**PROJECT REPORT  
ON**

**DIGITAL FORTRESS: CRAFTING AN INVINCIBLE HONEYPOT NETWORK WITH DISCORD ALERTS FOR CYBER SECURITY VIGILANCE**

**BY**

**Ali Rehman Class No. 12**

**Wajahat Hussain Class No. 24**

**Atif Zeb Class No. 139**

**SESSION: 2020-2024**

*A project report submitted to The University of Agriculture Peshawar. In The partial*

*Fulfillment of the Requirement for the degree of*

**BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY**

**Approved by**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Supervisor

Mr. Kashif Ali

Lecturer, ICS/IT

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Member

Dr. Muhammad Shahid

Assistant Professor, ICS/IT

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Member

Ms. Maliha

Lecturer, ICS/IT

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** HOD CS/IT (FMCS)

Dr. Asfandyar Khan

Assistant Professor

**INSTITUTE OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY**

**FACULITY OF MANAGEMENT AND COMPUTER SCIENCES**

**THE UNIVERSITY OF AGRICULTURE, PESHAWAR**

**KHYBER PAKHTUNKHWA-PAKISTAN**

**AUGUST, 2024**

**TABLE OF CONTENTS**

**CHAPTER NO. TITLE PAGE NO.**

ACKNOWLEDGEMENT i

ABSTRACT ii

**Chapter1:Introduction 1**

1.1 Background 1

1.2 Objectives 1

1.3 Scope 2

1.4 Importance of Cyber Security 2

**Chapter 2: Literature Review 3**

2.1 Honeypots and Their Role in Cyber Security 3

2.2 Network Monitoring Tools 3

2.3 Automation in Network Security 5

2.4 Previous Research 6

**Chapter 3: Methodology 7**

3.1 Network Setup 7

3.2 Vulnerability Simulation 7

3.3 Monitoring and Data Collection 7

3.4 Automation and Reporting 7

3.5 Data Analysis 8

**Chapter 4: Implementation 9**

4.1 Setting up the Honeypot 9

4.2 Simulating an Attack 9

4.3 Monitoring with Wire shark 11

4.4 Automation with Python 13

4.5 Enhancing Security 14

**Chapter 5: Firewall Configuration 15**

5.1 Importance of Firewalls 15

5.2 Key roles of Firewalls 16

5.3 Firewalls relevance in this project 16

5.4 Firewall 1: Configuration and Role 18

5.5 Firewall 2: Configuration and Role 23

5.6 Traffic Management and Security 28

**Chapter 6: Results and Analysis 29**

6.1 Attack Data 29

6.2 Network Security Enhancements 29

6.3 Insights and Observations 29

6.4 Challenges and Solutions 29

6.5 Case Studies 30

**Chapter 7: Conclusion 31**

7.1 Summary 31

7.2 Future Work 31

7.3 Recommendations 31

**Chapter 8: Appendices 33**

8.1 Python Script for Automation 33

8.2 Wire shark Captures 36

8.3 Discord Web-hook Integration 36

**References 38**

**ACKNOWLEDGEMENT**

All praises to ALLAH ALMIGHTY, the most merciful, kind and beneficent, and source of all knowledge, wisdom within and beyond our knowledge. All respected and possible tributes go to our holy Prophet Muhammad (S.A.W) who is forever guidance and knowledge for all human beings on this earth.

We are very grateful and thankful to our supervisor **Mr. Kashif Ali** for his great support and guidance during our study at the department and especially throughout the planning and duration of the whole research work. We pray for his long life and courage to serve the institute and deliver his expertise to students forever. He has supervised us in a good manner and also gives us a lot of coverage and information about this topic. It could not be possible without the help and support of our supervisor for giving me such attention and time to complete our thesis. Indeed, He is very intelligent and honest.

We would like to express our deepest gratitude to our parents for our parents .Their unwavering support, encouragement, and understanding throughout the course of this research project. Their constant belief in our abilities and their sacrifices have been instrumental in our academic success. Their moral and emotional support has been a source of strength and motivation, and we are truly grateful for their love and guidance.

**Ali Rehman**

**Wajahat Hussain**

**Atif Zeb**

**ABSTRACT**

In today’s world, cyber-attacks and threats are a significant issue. This thesis, titled “Digital Fortress: Crafting an Invincible Honeypot Network with Discord Alerts for Cyber Security Vigilance,” its show how to improve network security using a honeypot system. A honeypot system is a decoy system that attracts attackers to trap them. We use this honeypot to monitor and study the attacker’s activities and actions what they performed. In there we will use multiples operating systems. Honeypot will deploy on Ubuntu operating system and attacker will used the Kali Linux operating system. Windows operating system will use for real production environment. In there we also used firewalls and define the rules in firewalls, if attacker want to enter in our network firewall will redirect the attacker on honeypot and the real user will allow to real systems. The next step is to automate monitoring so by using Python scripts we automate the wire-shark which help to can watch the attackers and send live alerts to Discord server. These alerts include details like the attackers IP and Mac address. On the basis of this information we can find the hacker details using multiples methods. And also this help us to secure our network better and we fix that vulnerabilities. Our results show that this system is effective in finding and stopping cyber threats.

**Chapter 1:**

**INTRODUCTION**

* 1. **Background**

In today’s era, almost everything connected the internet and computer networks. From shopping online to managing our bank accounts, we depend on digital systems. However, this reliance also comes with risks. Cyber criminals, often called hacker, are constantly looking for vulnerabilities to exploit that and steal confidential information. Therefore, protecting our digital information and networks is more important than ever. One best way to enhance cyber security is by using a honeypot. A honeypot is a special computer system look like a real machine for attackers. When attacker try to attack to this decoy machine, we can monitor what they done in our system. And learn about their methods, and use this is information to make our real environment systems more secure.

* 1. **Objectives**

**The main goals of this project are:**

* To create a honeypot network that can attract the hackers.
* To Monitor and log activities of these attackers using wires shark it’s a free tool for network monitoring.
* To use Python Scripts to automate the process of sending alerts to Discord server. This way, we get real-time updates on what the attackers doing in our network.
* Implement firewalls that allow real user to real systems, and intruders allow to honeypot
* To analyze the collected data and use it to enhance the security of our network and protect our valuable data.
  1. **Scope**

**This project focuses on several key tasks:**

* Setting up a honeypot within a network to attract hackers.
* Simulating vulnerabilities, like a weak SSH service, to make the honeypot more appealing to attackers.
* Using wire shark to monitor the network traffic and collect data on attacker actions.
* Writing Python scripts to automate the alerting process and send notifications to a Discord server.
* Analyzing the data collected from these activities and action which performed by attacker use to enhance the overall security of the network and data.
  1. **Importance of Cyber Security**

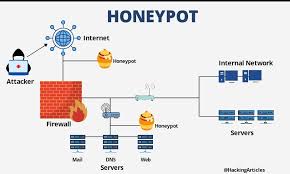
Cyber ​​security is critical to protecting sensitive information from attackers, theft and loss. In this digital era, the increasing number of cyber-attacks requires the best measures to protect networks and data. Effective cyber security ensures the integrity, confidentiality and availability of information, which is important to the functioning of modern society.

**Chapter 2:**

**LITERATURE REVIEW**

**2.1 Honeypots and Their Role in Cyber Security**

A honeypots is a fake system or decoy system designed to attract the attackers. It helps security teams understand attack methods and strategies to enhance defenses. Honeypots provide valuable insights into malicious activities and serve as a warning system for potential threats.



**2.2 Network Monitoring Tools**

Tools like wire shark are important for monitoring network traffic. Wire shark captures and analyzes data packets, providing detailed information network activity. This tool is GUI based. User can monitor all activity easily. This helps identify and respond to security incidents effectively. There are multiples tools for network monitoring but in this project we will use the Wire Shark.



**2.3 Automation in Network Security**

Automation of network tools enhances the efficiency of network security operations. By using python scripts we can automate wire shark such as monitoring, alerting, and reporting. This reduces the workload on security teams and ensures timely responses to threats. Automating tools play big role in this project.



**2.4 Previous Research**

Previous studies have shown the benefits of honeypots and analyzing cyber threats and attacks. Research suggests that automated network tools like monitoring and alert systems to improve the speed and accuracy of threat detection and response. Collect this all information about hacker’s methods and action what they done with our system, so we can used to enhance overall network security and defenses.

**Chapter 3:**

**METHODOLOGY**

**3.1 Network Setup**

We set up a network with multiple systems, including a honeypot. The network is configured to simulate the real environment, hackers easily attract to this decoy machine. It involves configuring various services and devices that mimic the infrastructure of a typical organization. The honeypot is strategically placed within the network to ensure it appears as a legitimate target.

**3.2 Vulnerability Simulation**

To attract attackers, we introduce a known vulnerability, such as an SSH vulnerability, in this honeypot system. This makes the honeypots an appealing target for hackers scanning the network. By simulating these vulnerabilities, we can observe how attackers exploit them and gather valuable data on their techniques and methods. This step is critical to making the honeypot effective at attracting real-world attackers.

**3.3 Monitoring and Data Collection**

We use Wire shark to monitor network traffic and collect data on attacker activities. Wire shark captures data packets, provide a detailed information about network interactions. User can find any type data what that required. But in there just include the source IP address and MAC address and also destination IP addresses, MAC addresses, and the type of traffic. The captured data is stored for further analysis, allowing us to study about the attackers’ behavior in detail.

* 1. **Automation and Reporting**

Python scripts are used to automate the monitoring and reporting process. In this scripts we automate the SSH traffic capture from wire shark and send on real time to discord server. The scripts send real-time alerts and updates if any issue or event occur to a Discord server, providing continuous updates on attacker activities. This automation ensures that we promptly notified of any suspicious activities, allowing for quick response. The alerts include important information such as the attacker’s IP address, MAC address, and the nature of the attack.

* 1. **Data Analysis**

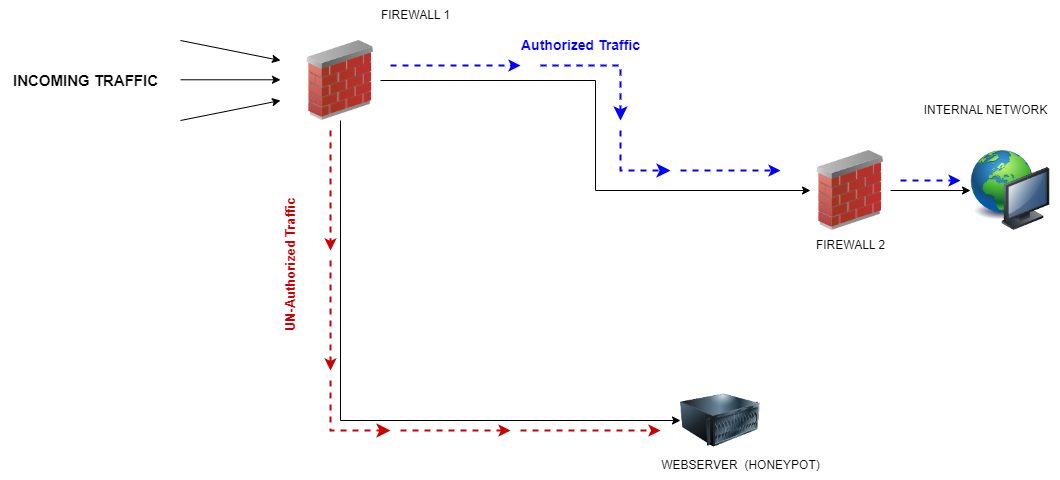
The collected data is analyzed to understand the attack patterns and methods. This analysis help in enhance network security and preventing future attacks. By studying the behavior of attackers, we can identify vulnerabilities in our defenses and take positive action. The data analysis process includes examining captured packets, identifying trends, and generating reports that provide insight into network security.

**Chapter 4:**

**IMPLEMENTATION**

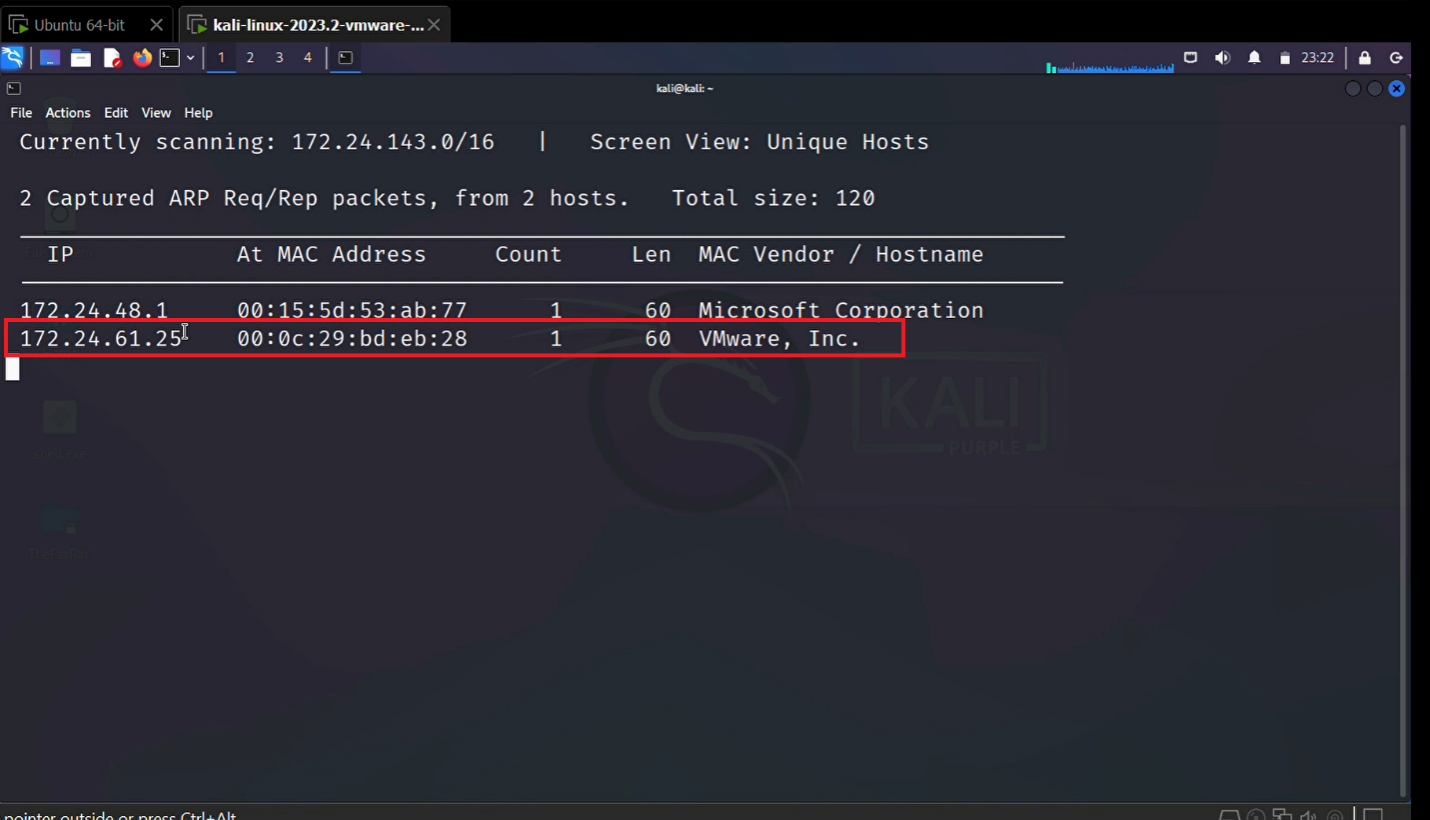
**4.1 Setting up the Honeypot**

We configure the honeypot within the network ensuring it appears as a vulnerable system. The honeypot is designed on Ubuntu machine, which will be vulnerable machine. We designed honeypot in such that way to attract attackers without compromising the rest of the network. This involves installing and configuring various services and applications that mimic a real target. The honeypot is carefully monitored to ensure it remains isolated from the production network.



**4.2 Simulating an Attack**

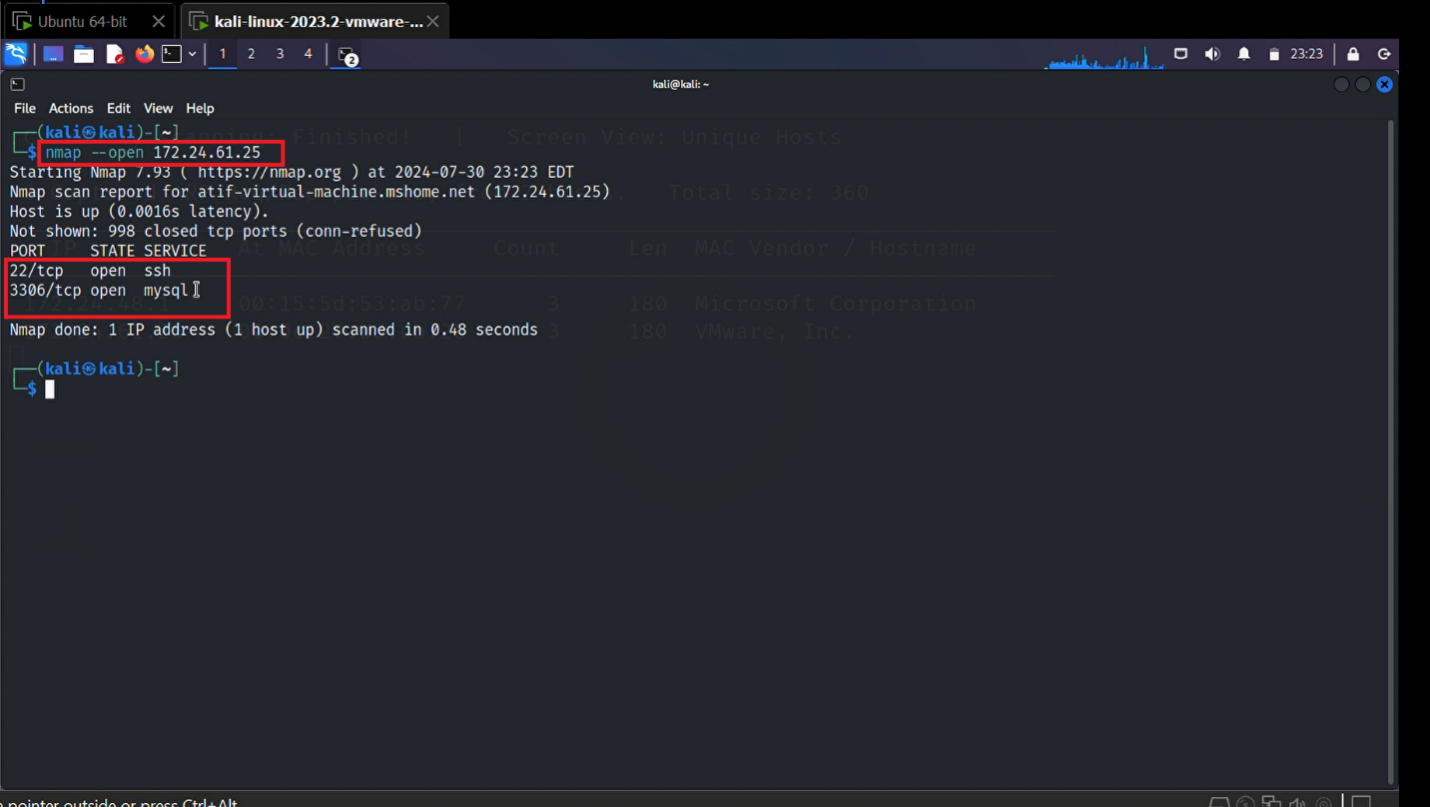
We simulate an attack by exploiting the SSH vulnerability in the honeypot. This helps test the effectiveness of the monitoring and alert system. The simulated attack provides a controlled environment to observe how the honeypot responds and how well our monitoring tools capture the attack details. This step is essential for validating the setup and ensuring that the honeypot can effectively lure real attackers.



* You can see attacker on the same network can see which machines available on the network, so they can easily scan a machine and exploit vulnerability. In the red circle this is our honeypot machine
* This can done by net-discover command and target IP range.

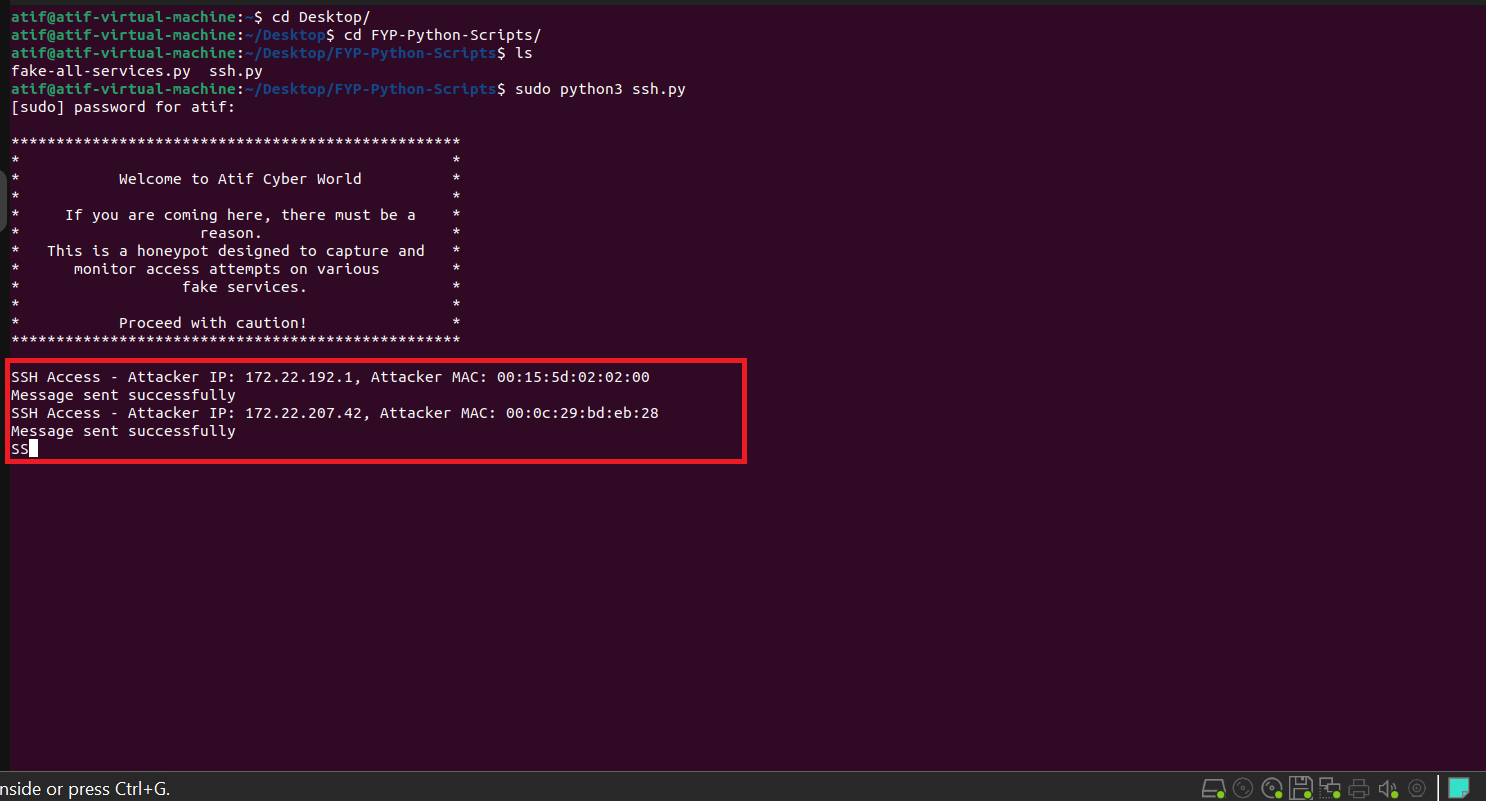
**Vulnerability Simulation**

* When attacker scan our network with a simple command of they can show our machine vulnerabilities which is TCP/22 SSH and SQL, so with mind the attacker they will exploit that vulnerability on the base what they need.
* In this phase attacker will exploit the SSH vulnerability.

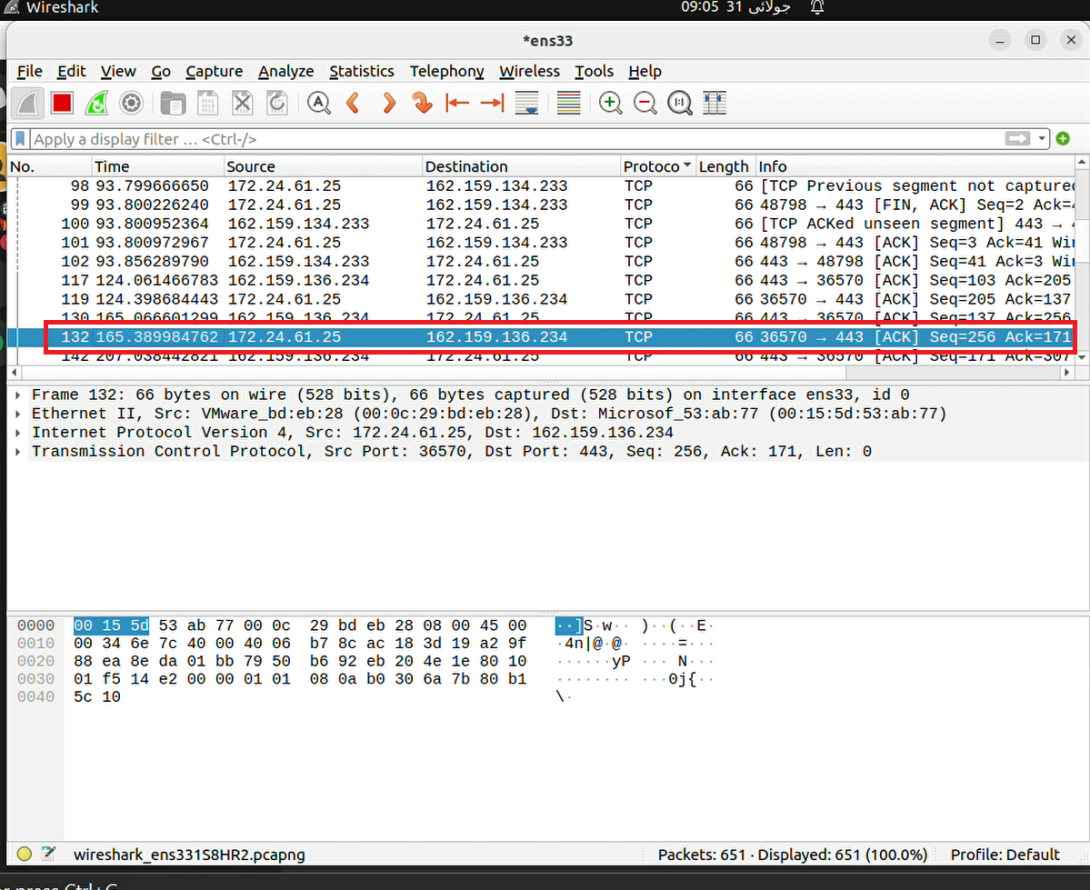


**4.3 Monitoring with Wire-shark**

Wire-shark is used to capture and analyze the network traffic. The tool provides detailed insights into the attack if the attacker enter in our network we can monitor with wire-shark. It will provide us the source IP and destination IP and MAC address of both. Wire-shark's ability to dissect and analyze packets helps to understand the nature of the attack and the methods used by the attacker. Captured data is stored for further analysis and reporting.



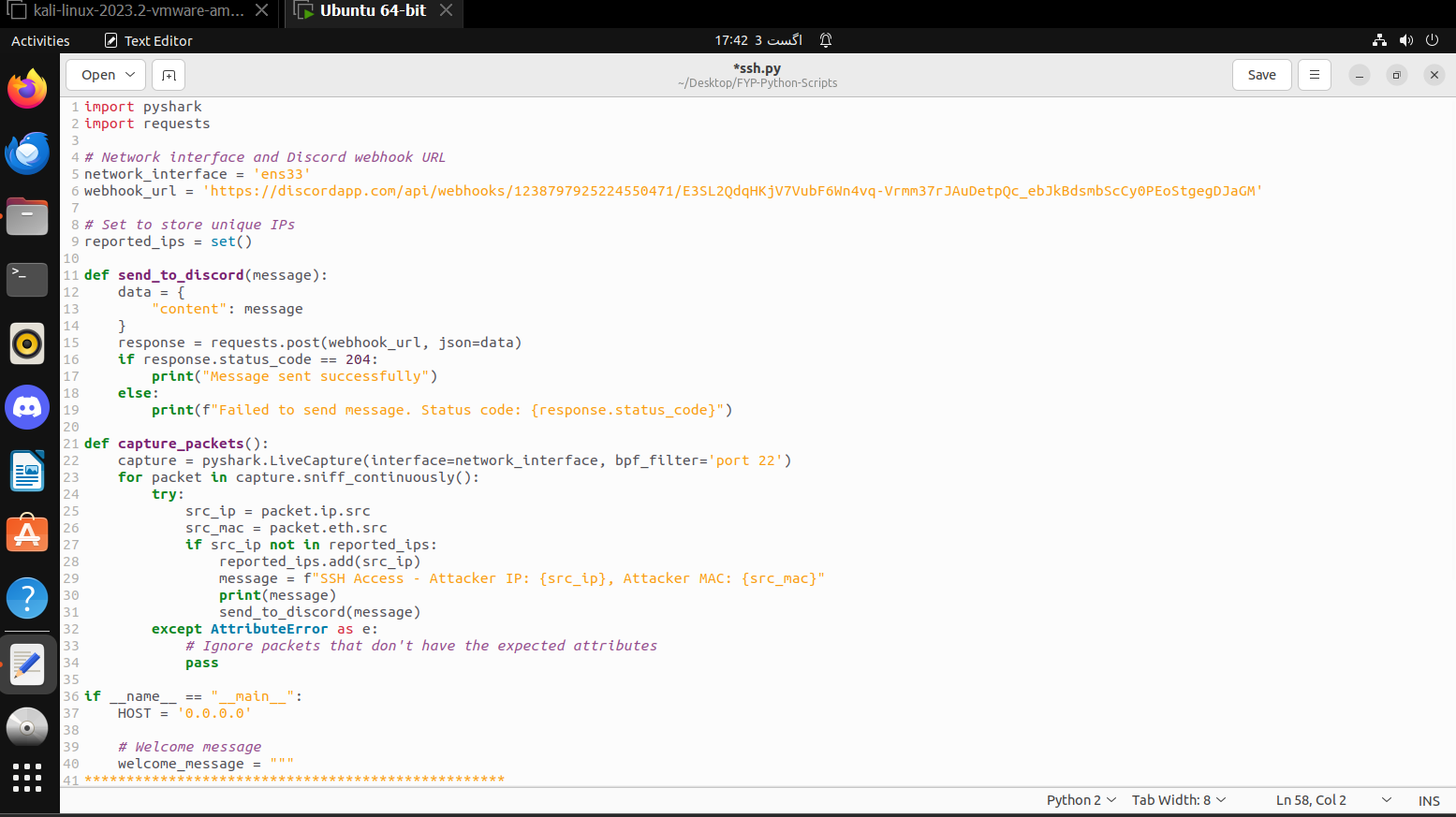
* There we run multiples scripts it’s also include fake services script if the attacker trying the exploit another port like my SQL and http port 80, the python script will give to us the attacker IP and MAC.



* We can monitor network by manually with wire-shark
* Wire-shark interface look like this
* We can use filters for finding about some specific IP details or port we can use multiples filter to filter the traffic

**4.4 Automation with Python**

Python scripts are developed to automate the process of monitoring and sending alerts and updates to a Discord server. These python scripts continuously monitor the network traffic and send real-time update to a Discord server when a suspicious activity is detected it will generate or send notification on Discord server. The alerts include detailed information about the attack, such as the IP and MAC of the attacker and the target. Automation ensures that we are always informed of any suspicious activities, allowing for quick and effective response. Monitoring of a network manual is too much hard so the automation play a big role in this project.



* This script will automate the wire shark traffic to send on discord channel server.
* In there we set the web hook to send the data on the server.

**4.5 Enhancing Security**

Based on the data collected and analyzed, we implement additional security measures and methods to protect the network. This includes updating firewall rules, patching vulnerabilities, and implementing best practice for network security. The Insights from honeypot monitoring help strengthen a network's overall security posture. By continually improving our defenses, we can better protect against future attacks.



* After attack we know the attacker behavior on that bases we can enhance our security polices and implement more securities layers.

**Chapter 5:**

**FIREWALL CONFIGURATION**

**5.1 Importance of Firewalls**

Firewalls are the cornerstone of network security. They act as a crucial barrier between your internal network and the external world, safeguarding your valuable data and systems from potential threats.

Firewalls are essential components of network security, serving as the first line of defense against cyber threats. They monitor and control incoming and outgoing network traffic based on predetermined security rules, effectively creating a barrier that prevents unauthorized access to or from private networks

**Types of firewalls**

1. **Packet-Filtering Firewall**

These examine packets and either allow or deny them based on source and destination IP addresses, ports, or protocols.

1. **Stateful Inspection Firewalls**

These track the state of active connections and make decisions based on the context of the traffic (i.e., the state of the connection).

1. **Proxy Firewalls**

These act as intermediaries between end-users and the services they are accessing. They can provide additional security by inspecting the content of the traffic.

1. **Next-Generation Firewalls**

These provide more advanced features, such as deep packet inspection, intrusion prevention systems (IPS), and application awareness.

1. **Cloud Firewalls**

These are hosted in the cloud and provide firewall functionality as a service, ideal for cloud-based infrastructure.

**5.2 Key Roles of Firewalls**

**1. Access Control**

Firewalls enforce security policies by filtering incoming and outgoing network traffic based on predefined rules. This prevents unauthorized access to your network resources.

1. **Threat Prevention**

Firewalls can protect against a wide range of threats, including viruses, malware, and hacking attempts. They can block malicious traffic and prevent it from spreading within your network.

1. **Data Protection**

By controlling network access, firewalls help protect sensitive data from unauthorized disclosure, modification, or destruction.

1. **Performance Optimization**

Firewalls can improve network performance by blocking unnecessary traffic and optimizing traffic flow.

**5.3 Relevance of Firewalls to Our Project**

Firewalls are indispensable for the security and integrity of any network, including the one we're designing for our project. They serve as the first line of defense against a multitude of threats, safeguarding our system and data from unauthorized access, malicious attacks, and data breaches.

**Protection of Sensitive Data**

Our project involves handling sensitive data, likely within the organization's network. Firewalls act as a crucial barrier, preventing unauthorized access to this data. Directing un authorize trafic to vulnerable web server acts as honeypot

1. **Distraction :-**

A honeypot can divert attackers away from critical systems, reducing the risk of successful attacks on your main network.

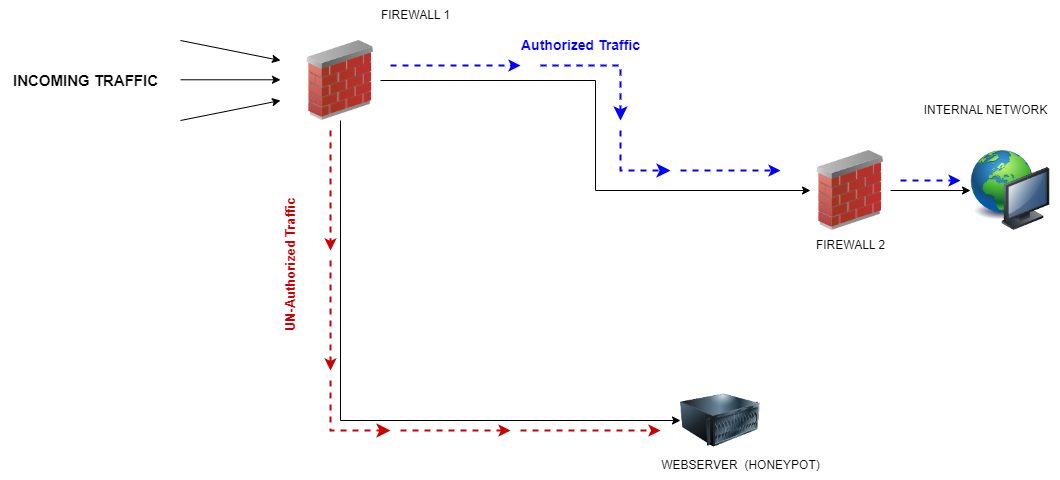
1. **Early Warning System :-**

By detecting and analyzing attacks on the honeypot, you can identify potential threats before they impact your production systems.

1. **Dedicated Network Segment**

Isolate the honeypot on a separate network segment to prevent it from affecting your main network.

**Firewall placement and network architecture**



**Firewalls Configurations**

**5.4 Firewall 1 : Configuration and role**

**Objective**

To set up Firewall 1 to allow access to everyone but direct authorized traffic to the second firewall, while directing unauthorized access to a vulnerable web server.

**Configuration Steps**

1. **Initial Setup and Installation**

* Install the Fortinet VM firewall on the designated VM.
* Access the firewall's web-based interface or CLI for configuration.

1. **Create Interfaces**

-->we will create three interfaces

* **External Interface**

The external interface connects Firewall 1 to the internet. This interface will be configured with a public IP address.

**CLI Configurations**

* + config system interface
  + edit "port1"
  + set vdom "root"
  + set ip <public-ip-address> 255.255.255.0
  + set allowaccess ping https ssh
  + set type physical
  + set role wan
  + next
  + End
* **Internal Interface : Connected to firewall 2**

The Internal Interface Connects Firewall 1 to Firewall 2

**CLI Configurations**

* + config system interface
  + edit "port2"
  + set vdom "root"
  + set ip 192.168.10.1 255.255.255.0
  + set allowaccess ping
  + set type physical
  + set role lan
  + next
  + End
* **Honeypot Interface**

The honeypot interface connects Firewall 1 to a separate network segment for the honeypot server.

**CLI Configurations**

* + config system interface
  + edit "port3"
  + set vdom "root"
  + set ip 192.168.20.1 255.255.255.0
  + set allowaccess ping
  + set type physical
  + set role dmz
  + next
  + End

1. **Create Address Objects for Policies**

Before creating the policies, define address objects for Authorized\_Users, Unauthorized\_Users, Firewall2\_IP, and Vulnerable\_Server\_IP

**CLI Configurations**

* + config firewall address
  + Edit ‘Authorized\_Users’
  + set subnet <authorized-user-subnet>
  + next
  + edit ‘Unauthorized\_Users’
  + set subnet<Unauthorized-users-subet>
  + next
  + edit ‘firewall2\_IP’
  + set subnet 192.168.10.2/32
  + next
  + edit ‘Vulnerable\_server\_ip’
  + set subnet 192.168.20.2/32
  + next
  + end

1. **Firewall Policies**

* **Allow Access to everyone**

Create a policy to allow all incoming traffic from the external network to the internal network.

**CLI Configurations**

* + config firewall Policy
  + edit 1
  + set name "Allow\_All"
  + set srcintf "port1"
  + set dstintf "port2"
  + set srcaddr “all”
  + set dstaddr “all”
  + set action accept
  + set schedule “always”
  + set service “ALL”
  + next
  + end
* **Direct Authorized Traffic to Second Firewall**

Create a policy to route traffic with authorized credentials (e.g., valid user credentials) to the second firewall.

**CLI Configurations**

* + config firewall Policy
  + edit 2
  + set name "Authorized\_to\_firewall2"
  + set srcintf "port1"
  + set dstintf "port2"
  + set srcaddr “Authorized\_users”
  + set dstaddr “firewall2\_IP”
  + set action accept
  + set schedule “always”
  + set service “ALL”
  + next
  + end
* **Redirect Unauthorized Traffic**

Create a policy to route traffic without proper authorization to the vulnerable web server. Ensure logging is enabled for unauthorized access attempts for monitoring purposes.

**CLI Configurations**

* + config firewall Policy
  + edit 3
  + set name "Unauthorized\_to\_Vulnerableserver"
  + set srcintf "port1"
  + set dstintf "port3"
  + set srcaddr “Unauthorized\_users”
  + set dstaddr “Vulnerable\_Server\_IP”
  + set action accept
  + set schedule “always”
  + set service “ALL”
  + next
  + end
* Routing Rules
* Configure static routes to ensure traffic is correctly directed:
  + Authorized traffic to the IP address of the second firewall.
  + Unauthorized traffic to the IP address of the vulnerable web server.

**CLI Configurations**

* + config router static
  + edit 1
  + set dst “192.168.10.2/32”
  + set gateway “192.168.10.1”
  + set device “port2””
  + next
  + edit 2
  + set dst “192.168.20.2/32”
  + set gateway “192.168.20.1”
  + set device “port3”
  + next
  + End

**Role of Firewall 1**

**Access Control and Monitoring:**

* Enable logging for all firewall policies to monitor traffic.
* Set up alerts for unauthorized access attempts.

**First Line of Defense:**

* Acts as the initial point of contact for incoming traffic from the external network.
* Filters and directs traffic based on authorization status.

**Traffic Management:**

* Ensures that authorized traffic is securely routed to the second firewall for further validation.
* Redirects unauthorized traffic to a controlled environment (vulnerable web server) for isolation and analysis.

**Enhanced Security :**

* By placing the vulnerable web server in the DMZ, it is isolated from the internal network, reducing the risk of unauthorized access to critical resources.
* Provides a buffer zone where potential threats can be contained and monitored without compromising the internal network.
  1. **Firewall 2 : Configuration and role**

**Purpose**

Protects the internal network, allowing access only to authorized devices based on device certificates.

**Configuration Steps**

1. Configure strict inbound and outbound firewall rules to allow only authorized traffic.
2. Implement certificate-based authentication for incoming connections.
3. Deny all other traffic.

* **Create Interfaces**

-->we will create two interfaces

* **Internal Interface : Connected to Internal network**

The Internal Interface Connects Firewall 2 with internal network

**CLI Configurations**

* + config system interface
  + edit "port1"
  + set vdom "root"
  + set ip 192.168.1.1 255.255.255.0
  + set allowaccess ping https ssh
  + set type physical
  + set role lan
  + next
  + end
* **External Interface : Connected to Firewall 1.**

The Internal Interface Connects Firewall 2 with internal network

**CLI Configurations**

* + config system interface
  + edit "port2"
  + set vdom "root"
  + set ip 192.168.1.2 255.255.255.0
  + set allowaccess ping
  + set type physical
  + set role wan
  + next
  + End
* **Configure Certificate Authority (CA)**
* Set up a certificate authority (CA) to issue device certificates.
* Configure Firewall 2 to trust the CA and verify device certificates for incoming connections.

**CLI Configurations to Import CA Certificates**

* + Config vpn certificate ca
  + edit ‘CA\_Certificate’
  + set ca ‘<base64\_encoded\_CA\_Certs>’
  + next
  + end
* **Configure Device Certificate Authentication**

Create a peer user group for device certificate verification.

**CLI Configurations**

* + config user peer
  + edit ‘Authorized\_Devices’
  + set ca ‘CA\_Certificate’
  + set subject ‘CN=AuthorizedDevice’
  + next
  + end

**CLI Configurations for an SSL/SSH inspection profile to use the certificate.**

* + config firewall ssl-ssh-profile
  + edit "Device\_Certificate\_Profile”
  + config ssl
  + set client-cert-request enable
  + set client-cert-check enable
  + set client-cert-ca "CA\_Certificate”
  + End
  + next
  + end
* **Create Firewall Policies**
* Allow inbound traffic only from devices with valid certificates on the internal interface.
* config firewall policy
* edit 1
* set name "Allow\_Authorized\_Devices"
* set srcintf "port2"
* set dstintf "port1"
* set srcaddr "all"
* set dstaddr "Internal\_Network"
* set action accept
* set schedule "always"
* set service "ALL"
* set ssl-ssh-profile "Device\_Certificate\_Profile"
* next
* end
* **Deny all other inbound traffic.**
* config firewall policy
* edit 2
* set name "Deny\_All\_Other\_Traffic"
* set srcintf "port2"
* set dstintf "port1"
* set srcaddr "all"
* set dstaddr "all"
* set action deny
* set schedule "always"
* set service "ALL"
* next
* End
* **outbound traffic policies based on our projet requirements.**
* config firewall policy
* edit 3
* set name "Allow\_All\_Outbound"
* set srcintf "port1"
* set dstintf "port2"
* set srcaddr "Internal\_Network"
* set dstaddr "all"
* set action accept
* set schedule "always"
* set service "ALL"
* next
* End

**5.6 Traffic Management and Security**

* **Load Balancing (Optional):** If you have multiple servers in your internal network, you might consider load balancing to distribute traffic across them.
* **Intrusion Prevention System (IPS):** Enable IPS features on both firewalls to detect and block malicious attacks.
* **VPN (Optional):** If remote access is required, configure VPN tunnels on Firewall 1 for secure connections.
* **Monitoring and Logging:** Implement logging and monitoring to track network activity and identify potential security threats.

**CLI COMMANDS OVERVIEW**

* **edit**: Starts editing a specific policy or route.
* **set name**: Sets a name for the policy for easy identification.
* **set srcintf / set dstintf**: Specifies the source and destination interfaces.
* **set srcaddr / set dstaddr**: Specifies the source and destination addresses.
* **set action accept**: Sets the action to accept the traffic.
* **set schedule "always"**: Ensures the policy is always active.
* **set service "ALL"**: Applies to all types of services (protocols and ports).
* **set gateway**: Specifies the next-hop gateway IP address.
* **set device**: Specifies the interface through which the traffic should be routed

**Chapter 6:**

**RESULTS AND ANALYSIS**

**6.1 Attack Data**

The data collected from honeypot show detailed information about the attacks. This include the IP and MAC addresses of the attackers, and the method they used to exploit vulnerabilities, and the nature of the attacks. On the bases of this data helps in understanding the behavior of attackers and identifying weaknesses in the network. This data help us to remove that vulnerabilities which can easily exploit by hackers and secure the network.

**6.2 Alert Effectiveness**

The automated alerts sent to the Discord server prove to be effective in providing real-time updates on attacker activities. The alerts include crucial information that helps in quickly identifying and responding to threats quickly. The automation ensures that we are promptly notified of any suspicious activities, allowing for quick and effective response.

**6.3 Security Improvements**

Based on the data analysis, multiples security improvements are implemented. This includes firewall rules, SSH allowing scripts, patching vulnerabilities, and implementing best practices for the network security. The insights gained from monitoring the honeypot help in strengthening the overall security posture of the network. By continuously improving defenses, we can better protect against future attacks.

**6.4 Challenges and Solutions**

**1. Challenge: Honeypot Detection and Evasion**

* **Description:** Many sophisticated attackers can identify honeypots and avoid them, making the data collection less effective.
* **Solution:** Implementing advanced decoy techniques and using realistic services to mimic genuine vulnerabilities. Regularly updating the honeypots configurations to reflect current security threats.

**2. Challenge: Data Overload and Management**

* **Description:** A honeypot can generate a large volume of data, which can be overwhelming to analyze and difficult to manage that data manually.
* **Solution:** Employing automated data filtering and analysis tools like Security information and event management (SIEM) systems. Utilizing machine learning models to categorize and prioritize the data meaningful insights.

**3. Challenge: Security Risks and Isolation**

* **Description:** Honeypots, by their nature, are vulnerable and can potentially be exploited to lunch attacks on other systems.
* **Solution:** Ensuring proper network isolation and segmentations. Implementing strict firewall rules and monitoring to prevent any unintended spillover effects.

**4. Challenge: Legal and Ethical Considerations**

* **Description:** Capturing and logging potentially sensitive data might raise legal and ethical issues.
* **Solution:** in this project we store the dump data, look like its real data of organization. Because it’s huge risk on real data.

**6.5 Case Studies**

1. **Educational Institution:** We can set up a honeypot in a university network to study attacks. This revealed common attack patterns and helped improve the institutions security measures.
2. **Corporate Environment:** in a corporate setting, our honeypot can integrated with existing security systems, providing insights into attack methods and improving overall defense strategies.
3. **Government Network:** We can deployed a honeypot in a government network to enhance threat intelligence. This case study demonstrated effective monitoring and response to sophisticated threats.

**Chapter 7:**

**CONCLUSION**

**7.1 Summary**

In this chapter, we recap the main points of our honeypot project. Our goal was to deploy honeypot on Ubuntu operating system, and add SSH vulnerability to attract attacker easily. In the honeypot store dummy data of the organization useless data. In second part we set the monitoring tools to monitor the attacker and automate the whole process of monitoring with python scripts and Discord server. This scripts automate the process of monitoring to send live data from wire shark to Discord channel dashboard which include attacker IP and MAC. In the other part deploy the firewall from outside the network and inside the network. Outside firewall allow authorized user to real systems, and unauthorized traffic will redirect on the honeypot system. Further of collecting all the data mean behavior of the attackers and method we enhance the security, and update the firewalls rules, and patching vulnerabilities.

**7.2 Future Work**

Looking ahead, there are several areas of potential improvement and further research. A key area is expanding Honeypot's capabilities using advanced machine learning algorithms to better detect and classify threats in real time. Additionally, our honeypot scope can provide a broader understanding of potential threats by simulating more diverse environments and attack vectors. Future work could explore honeypots with other security systems such as intrusion detection systems to create a more comprehensive defense strategy. Researching ways to make honeypots less detectable and more resilient to evasion techniques is another important avenue for future investigation.

**7.3 Recommendations**

Based on the findings of this project, several recommendations can be made for effectively using honeypots in cyber security. First it’s critical to deploy honeypots with realistic configuration to ensure they attract genuine attacks. Regularly updating these configurations to reflect current vulnerabilities and threats is also important. We recommended integrating honeypots with broader security measures, such as network monitoring and incident response systems, to maximize their effectiveness. Additionally, careful consideration should also be given to the legal and ethical implications of honeypot data collection, while ensuring compliance with privacy laws and best practices. Finally, organizations should invest in training and awareness to ensure that honeypot results are properly interpreted and used to enhance the overall security posture.

**Chapter 8:**

**APPENDICES**

**8.1 Python Script for Automation**

* **Details:** Include the full python script used for automating honeypot data collection and analysis.
* **Here is the Python script**

import pyshark # Import the pyshark library for packet capturing

import requests # Import the requests library to send HTTP requests

# Network interface and Discord webhook URL

network\_interface = 'ens33' # Specify the network interface to capture packets from

webhook\_url = 'https://discordapp.com/api/webhooks/1238797925224550471/E3SL2QdqHKjV7VubF6Wn4vq-Vrmm37rJAuDetpQc\_ebJkBdsmbScCy0PEoStgegDJaGM' # URL for the Discord webhook

# Set to store unique IPs

reported\_ips = set() # Initialize a set to keep track of IP addresses that have already been reported

def send\_to\_discord(message):

"""

Sends a message to the specified Discord webhook URL.

Args:

message (str): The message to send to Discord.

"""

data = {

"content": message # Create a dictionary with the message content

}

response = requests.post(webhook\_url, json=data) # Send the POST request with the message data

if response.status\_code == 204: # Check if the request was successful

print("Message sent successfully") # Print a success message

else:

print(f"Failed to send message. Status code: {response.status\_code}") # Print an error message with the status code

def capture\_packets():

"""

Captures packets from the specified network interface and sends alerts to Discord if SSH traffic is detected.

"""

capture = pyshark.LiveCapture(interface=network\_interface, bpf\_filter='port 22') # Create a live capture object with a BPF filter for SSH (port 22)

for packet in capture.sniff\_continuously(): # Continuously capture packets

try:

src\_ip = packet.ip.src # Extract the source IP address from the packet

src\_mac = packet.eth.src # Extract the source MAC address from the packet

if src\_ip not in reported\_ips: # Check if the source IP is not already reported

reported\_ips.add(src\_ip) # Add the source IP to the set of reported IPs

message = f"SSH Access - Attacker IP: {src\_ip}, Attacker MAC: {src\_mac}" # Create a message with the attacker's IP and MAC addresses

print(message) # Print the message to the console

send\_to\_discord(message) # Send the message to Discord

except AttributeError as e:

# Ignore packets that don't have the expected attributes (e.g., non-IP packets)

pass

if \_\_name\_\_ == "\_\_main\_\_":

HOST = '0.0.0.0' # Define the host address (not used in this script)

# Welcome message

welcome\_message = """

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* Welcome to Atif Cyber World \*

\* \*

\* If you are coming here, there must be a \*

\* reason. \*

\* This is a honeypot designed to capture and \*

\* monitor access attempts on various \*

\* fake services. \*

\* \*

\* Proceed with caution! \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

"""

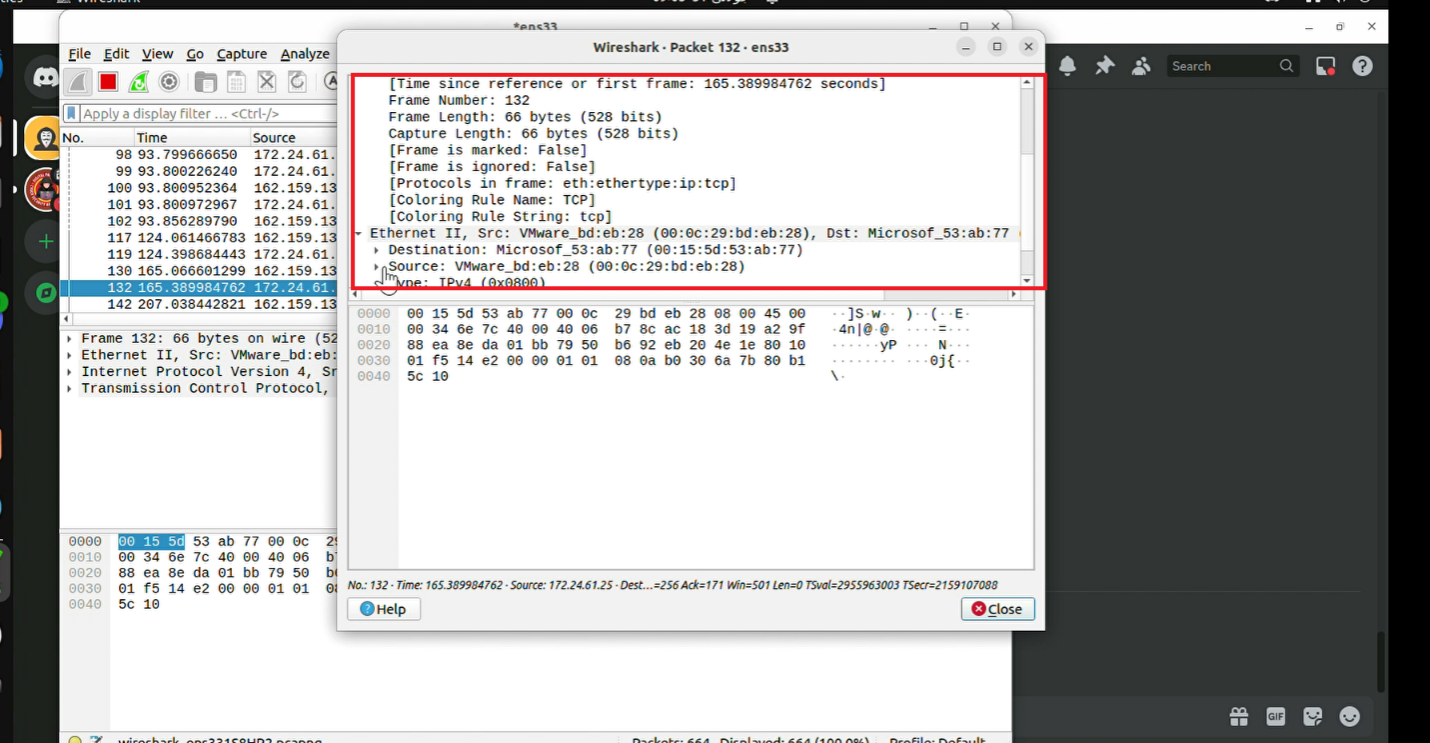
print(welcome\_message) # Print the welcome message to the console

# Start capturing packets

capture\_packets() # Call the function to start packet capturing

**8.2 Wire shark Captures**

We’ve included sample Wire shark captures to show the type of traffic our honeypot collected.

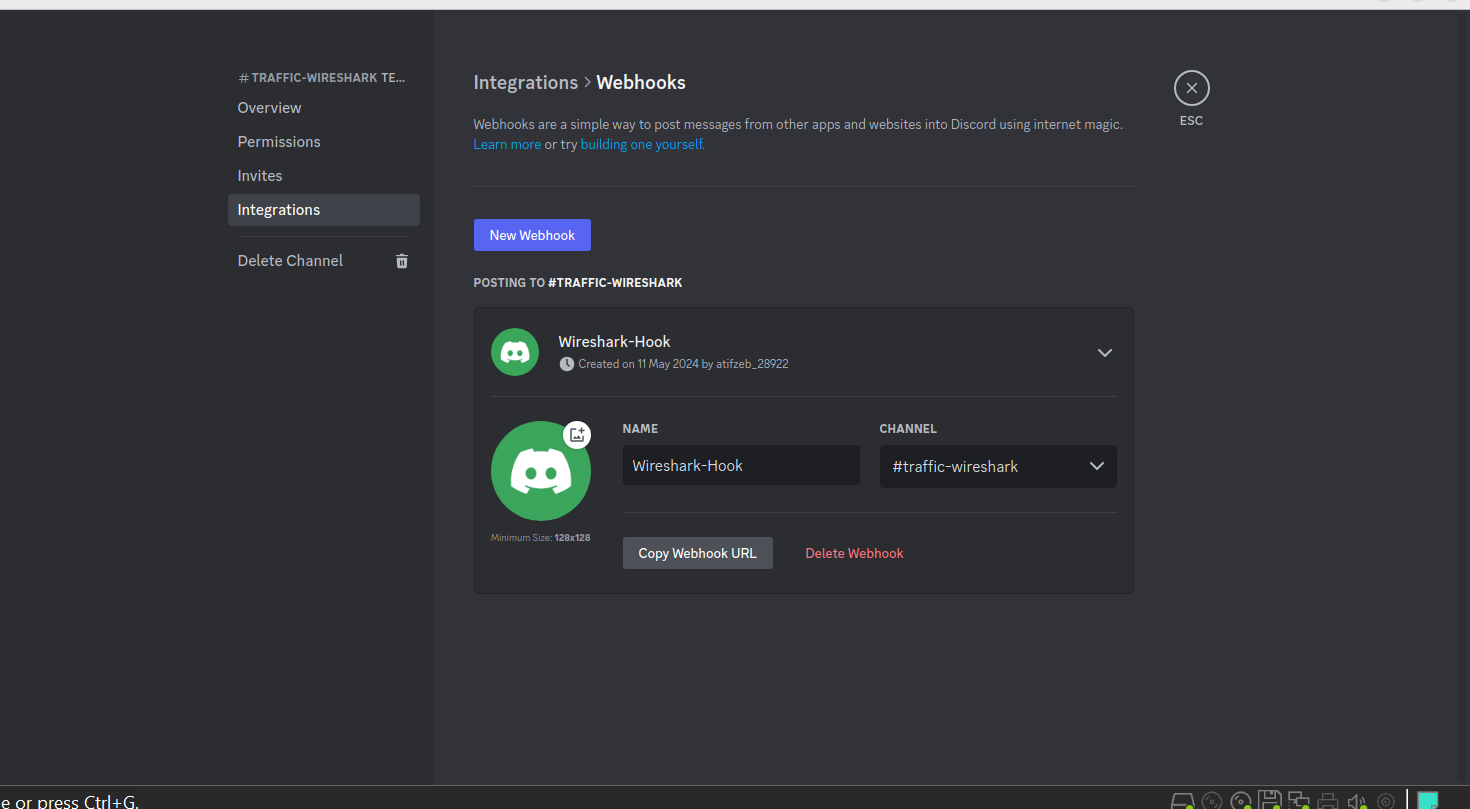


* In there we collect multiple type information of a specific IP from a packet.
* We can use for this multiple filter to filter the traffic and show that traffic what we want

**8.3 Discord Web-hook Integration**

* **Setup Instruction:**

1. In the very first step we download Discord app from Google and install on Ubuntu.
2. In the second step we create a server on Discord name is “Wireshark-Traffiic-Server”.
3. The third and last step is to copy the web-hook link from integration and paste in code.

****

**REFERENCES**

**Wires-hark Foundation. (2024).** Wire-shark User’s Guide.

**Chappell, L. (2021).** Wire-shark for Beginners. O’Reilly Media.

**Discord Inc. (2024).** Discord API Documentation.

**Herman, T. (2020).** Python for Cyber security: Using Python for Cyber security and Penetration Testing. Packt Publishing.

**O’Reilly, T. (2021).** Python Network Programming Cookbook: A Practical Guide to Building Network Applications Using Python. Packt Publishing.

**Wright, J. (2019).** Practical Packet Analysis: Using Wire-shark to Solve Real-World Network Problems. No Starch Press.

**Sharma, R. (2022).** Automating Security Operations with Python. Cyber security Books.

**O’Neill, M. (2023).** Real-Time Network Monitoring with Wire-shark. McGraw-Hill Education.

**Bland, S. (2021).** *Creating Custom Discord Bots with Python*. *Packt Publishing*.

1. <https://www.wireshark.org/docs/wsug_html_chunked>

2. <https://www.oreilly.com/library/view/wireshark-for/9781801074058>

3. <https://discord.com/developers/docs/intro>

4. <https://www.packtpub.com/product/python-for-cybersecurity/9781801074607>

5. <https://www.packtpub.com/product/python-network-programming-cookbook/9781801070708>

6. <https://www.springer.com/gp/book/9783642049464>

7. <https://nostarch.com/packetanalysis2>

8. <https://www.amazon.com/Automating-Security-Operations-Python-Sharma/dp/1947487442>

9.<https://www.mhprofessional.com/9781264264480-usa-real-time-network-monitoring-with-wireshark>