

**EASTERN INTERNATIONAL UNIVERSITY**

**SCHOOL OF COMPUTING AND INFORMATION TECHNOLOGY**

**DEPARTMENT OF COMPUTER NETWORKS AND DATA COMMUNICATIONS**



# PROJECT 2

PE

**Student**

Nguyen Thanh Dong – 2131220022

Ha Quang Minh - 1931220012

**Supervisor**

Ph.D. Huynh Tan Phuoc

# Binh Duong, 11, 2024

**EASTERN INTERNATIONAL UNIVERSITY**

**SCHOOL OF COMPUTING AND INFORMATION TECHNOLOGY**

**DEPARTMENT OF COMPUTER NETWORKS AND DATA COMMUNICATIONS**



# PROJECT 1

PENETRAT

**Student**

Nguyen Thanh Dong – 2131220022

Ha Quang Minh - 1931220012

**Supervisor**

Ph.D. Huynh Tan Phuoc

# Binh Duong, 11, 2024

### Abstract

Penetration testing network system is the process of identifying security vulnerabilities in an application by assessing the system or network for various malicious techniques. It can be simply understood as assessing security by attacking the system to find potential security problems or detecting traces when the system is compromised. The purpose of this testing is to secure important data from outsiders such as hackers who may have unauthorized access to the system. Once the vulnerability is identified it can be used to exploit the system to gain access to sensitive information.

We start by studying the documents about the basic theory of Pentest (Penetration Testing), then look up the software that supports Pentest.

First, deploy a virtual environment (VMware) to create a basic network system consisting of many computers with many different operating systems (Kali Linux, macOS). Next, install the Armitage software supported within the Kali Linux operating system, then we rely on Armitage to scan the machines on the network. Next, we will scan for vulnerabilities before entering and attacking a computer. Furthermore, by exploiting open ports, we have caused the attacked computer to shutdown. And the final goal we need to achieve is to provide solutions to overcome existing network vulnerabilities and enhance security performance in the future.

**KeyWord:** Penetration Testing, Network Security, Vulnerability Assessment, Kali Linux, Armitage, Cybersecurity, Virtual Environment, Open Ports, Exploitation, System Security.

### Acknowledgement

We would like to express our deepest gratitude to PhD. Huynh Tan Phuoc, whose expertise, guidance, and continuous support have been instrumental in the successful completion of this project. We are also grateful to the publishers of the Kali Linux, VMware and Armitage software for providing the resources and environment necessary for carrying out this research. Finally, we extend our appreciation to our peers and family for their encouragement throughout this journey.

## TABLE OF CONTENTS

**Abstract and Keywords** i

[**Acknowledgement** ii](#_heading=h.gjdgxs)

[**Table of Contents** iii](#_heading=h.30j0zll)

[**List of Figures** iv](#_heading=h.1fob9te)

**List of Tables** v

**List of Abbreviations** vi

[**Chapter 1: Introduction** 10](#_heading=h.3znysh7)

* 1. [Reason for Choosing the Topic 10](#_heading=h.2et92p0)
  2. [Research Content 10](#_heading=h.tyjcwt)
  3. [Scientific and Practical Significance 10](#_heading=h.3dy6vkm)

[**Chapter 2: Theoretical Basis and Tools Utilized** 12](#_heading=h.1t3h5sf)

* 1. [Overview of computer network information security theory 12](#_heading=h.4d34og8)
     1. [Key concepts 12](#_heading=h.2s8eyo1)
     2. [Threats in Network Security 12](#_heading=h.17dp8vu)
     3. [Security Measures and Techniques 12](#_heading=h.3rdcrjn)
     4. [Network Security Architectures 13](#_heading=h.26in1rg)
     5. [Standards and Frameworks 13](#_heading=h.lnxbz9)
     6. [Emerging Trends 13](#_heading=h.35nkun2)
  2. [Tools Utilized 13](#_heading=h.1ksv4uv)
     1. [Virtual environment design 13](#_heading=h.44sinio)
     2. [Scanning software 14](#_heading=h.2jxsxqh)

[**Chapter 3: The Architectural Design** 16](#_heading=h.z337ya)

* 1. [Survey of Current State and Requirements 16](#_heading=h.3j2qqm3)
     1. [Survey of Current State 16](#_heading=h.1y810tw)
     2. Requirements 16
  2. [Network System Analysis 17](#_heading=h.4i7ojhp)

[**Chapter 4: The Armitage Software** 18](#_heading=h.2xcytpi)

* 1. [Main Features of Armitage 18](#_heading=h.1ci93xb)
     1. [Graphical User Interface 18](#_heading=h.3whwml4)
     2. [Exploit Management 19](#_heading=h.2bn6wsx)
     3. [Target Scanning and Management 19](#_heading=h.qsh70q)
     4. Payload and Post-Exploitation 20
     5. [Visual Interaction and Control 21](#_heading=h.3as4poj)
     6. [Scripted attacks 21](#_heading=h.1pxezwc)
     7. [Real-Time Monitoring 22](#_heading=h.49x2ik5)
  2. [System Implementation 22](#_heading=h.2p2csry)

[**Chapter 5: Project Performance and Evaluation** 23](#_heading=h.147n2zr)

* 1. [Testing Programs and Results 23](#_heading=h.3o7alnk)
     1. [Testing Programs 23](#_heading=h.23ckvvd)
     2. [Results 26](#_heading=h.ihv636)
  2. [Solutions for Mitigation 26](#_heading=h.32hioqz)
  3. [Future Development Directions 29](#_heading=h.1hmsyys)

[**Chapter 6: Conclusion** 31](#_heading=h.41mghml)

* 1. [Summary 31](#_heading=h.2grqrue)
  2. [Conclusion 31](#_heading=h.vx1227)

[Reference 32](#_heading=h.3fwokq0)

## LIST OF FIGURES

Figure 1: The interface Virtual environment 14

Figure 2: The Armitage Tool 15

Figure 3: The TryHackMe Website 15

Figure 4: Graphical User Interface 18

Figure 5: Exploit Management 19

Figure 6: Network Scan Input Dialog 20

Figure 7: Exploit Configuration Window 20

Figure 8: Armitage Attack Visualization 21

Figure 9: Penetration Testing Stages 22

Figure 10 -14: Testing program process

Figure 10: Interface of Armitage 23

Figure 11: After the scanning 24

Figure 12: TCP Open 24

Figure 13: Access broken hole 25

Figure 14: Attack 25

Figure 15: Windows machine was reboosted 26

Figure 16: Check whether the firewall 27

Figure 17: The Window Firewall State is On 27

Figure 18: New Inbound Rule Wizard 28

Figure 19: Windows Firewall with Advanced Security 29

**LIST OF ABBREVIATION**

| Pentest RBAC  DoS DDoS SQL VPN IDPS PCI DSS GDPR HIPAA MITM  DoS DDoS AES RSA IDPS RBAC VPN  NIST Cybersecurity Framework  IoT CRM ERP SAST DAST  WAF | Penetration Testing  Role-Based Access Control Denial of Service  Distributed Denial of Service Structured Query Language Virtual Private Network  Intrusion Detection and Prevention Systems Payment Card Industry Data Security Standard General Data Protection Regulation  Health Insurance Portability and Accountability Act Man-in-the-Middle  Denial of Service  Distributed Denial of Service Advanced Encryption Standard Rivest–Shamir–Adleman  Intrusion Detection and Prevention Systems Role-Based Access Control  Virtual Private Network  National Institute of Standards and Technology Cybersecurity Framework  Internet of Things Security Customer Relationship Management Enterprise Resource Planning  Static Application Security Testing Dynamic Application Security Testing  Web Application Firewall |
| --- | --- |

| AI-driven monitoring  DevSecOps | Artificial Intelligence-driven Monitoring  Development, Security, and Operations |
| --- | --- |

# Chapter 1. Introduction

### Reason for Choosing the Topic

–Minh

Effective IT monitoring, visualization, and security assessment are essential for system stability and optimization. Zabbix, Grafana, OpenVAS, and Nessus are widely used tools that provide comprehensive solutions in these areas.

Zabbix is an open-source monitoring solution offering scalable and flexible monitoring for networks, servers, and applications. It provides proactive alerts, multiple data collection methods, and cost-effective enterprise-level features. Grafana enhances monitoring with customizable dashboards, real-time data analysis, and seamless integration with various databases, improving visualization and decision-making.

For security, OpenVAS and Nessus are powerful vulnerability scanners. OpenVAS, an open-source tool, detects weaknesses in servers and networks, offering regular database updates for enhanced security. Nessus, a commercial alternative, provides extensive vulnerability coverage, risk assessments, and compliance auditing, ensuring security policy adherence.

By integrating these tools, organizations achieve a complete monitoring and security solution. Zabbix collects data, Grafana visualizes it, and OpenVAS/Nessus identify vulnerabilities, helping administrators enhance performance, detect threats, and maintain cybersecurity. Their robust features, strong community support, and cost-effectiveness make them ideal choices for IT infrastructure management.

### Research Content

Analyze network monitoring and security assessment processes using Zabbix, Grafana, OpenVAS, and Nessus. Research methods for system monitoring, log analysis, and performance optimization. Study approaches for vulnerability scanning, risk assessment, and penetration testing. Explore the implementation of virtual environments (VMware) for testing network security through supporting tools such as OpenVAS, Nessus, Nmap, and other security frameworks.

### Scientific and Practical Significance

* **Proactive network and system monitoring:** Zabbix provides real-time network and system health tracking, helping administrators detect and resolve performance issues before they escalate.
* **Enhanced data visualization:** Grafana enables in-depth data analysis and dashboard customization, improving decision-making through clear and interactive visual representations.
* **Comprehensive vulnerability scanning:** OpenVAS and Nessus help identify security weaknesses in network infrastructures, ensuring that vulnerabilities are promptly detected and mitigated.
* **Compliance with security standards:** The use of OpenVAS and Nessus assists organizations in meeting security compliance requirements, such as PCI DSS, GDPR, HIPAA, and ISO 27001.
* **Efficient incident response and mitigation:** By integrating these tools, organizations can streamline their security operations, quickly respond to potential threats, and strengthen overall system resilience.
* **Optimization of IT infrastructure:** Continuous monitoring and vulnerability assessment improve system performance, reduce downtime, and enhance overall network security.

–Minh - Done

# Chapter 2: Theoretical Basis and Tools Utilized

**–Minh**

### Overview of computer network information security theory

Computer Network Information Security Theory encompasses the principles, models, and practices designed to protect data, resources, and communication over networks. Below is an overview:

#### Key Concepts

* **Confidentiality:** Ensuring that data is accessible only to those authorized to view it.
* **Integrity:** Ensuring that data remains unchanged and unaltered unless done by authorized processes.
* **Availability:** Ensuring that data and resources are available to users when needed.
* **Authentication:** Verifying the identity of a user or system before granting access.
* **Authorization:** Controlling access to resources based on user identity or role
* **Non-repudiation:** Guaranteeing that actions or transactions cannot be denied after the fact.

#### Threats in Network Security

* + - * **Eavesdropping**: Unauthorized interception of data during transmission.  
        Man-in-the-Middle (MITM) Attacks: Attacker intercepts and potentially alters communications between two parties.
      * **Phishing and Social Engineering**: Deceptive methods to gain confidential information.
      * **Malware**: Viruses, worms, ransomware, and spyware that harm or exploit systems.
      * **Denial of Service (DoS) and Distributed Denial of Service (DDoS)**: Overwhelming a system to make it unavailable to users.
      * **SQL Injection and Cross-Site Scripting (XSS):** Exploits targeting web applications.
      * **Zero-Day Exploits**: Exploiting vulnerabilities before they are known or patched.

#### Security Measures and Techniques

* + - * **Network and System Monitoring:** Zabbix provides real-time tracking of system performance, ensuring early detection of anomalies and potential security threats.
      * **Data Visualization for Security Insights:** Grafana helps analyze logs and monitoring data, offering clear dashboards that highlight security incidents and trends.
      * **Vulnerability Scanning and Risk Assessment:** OpenVAS and Nessus scan for security flaws in servers, networks, and applications, enabling proactive mitigation.
      * **Automated Security Alerts and Incident Response:** Zabbix and OpenVAS integrate to provide automated notifications, allowing security teams to respond quickly.
      * **Compliance and Security Auditing:** Nessus assists organizations in meeting compliance standards by identifying misconfigurations and weaknesses in security policies.
      * **Penetration Testing for System Hardening:** Using OpenVAS alongside other tools like Nmap allows security teams to simulate attacks, test defenses, and improve resilience

-Minh-

#### Network Security Architectures

* + - * Defense in Depth with Multi-Layer Monitoring**:** Zabbix continuously monitors system health, network traffic, and security events at multiple layers to provide comprehensive protection.
      * Zero Trust with Continuous Verification: Nessus and OpenVAS regularly scan for vulnerabilities, ensuring that no device or service is implicitly trusted.
      * Endpoint Security via Real-Time Alerts: Zabbix and Grafana integrate to track endpoint performance and detect suspicious activity, helping secure connected devices.
      * Secure Remote Access through VPN & Encryption: Implementing VPN solutions ensures safe remote access, while TLS/SSL encryption secures data transmission.
      * Automated Threat Detection and Response: Combining Zabbix, OpenVAS, and Nessus allows for automated security auditing, real-time alerts, and proactive incident response.

#### Standards and Frameworks

* + - * **ISO/IEC 27001 Compliance through Continuous Monitoring:** Zabbix helps organizations maintain ISO/IEC 27001 compliance by providing real-time monitoring, log analysis, and anomaly detection.
      * **NIST Cybersecurity Framework Implementation:** OpenVAS and Nessus support risk assessment, vulnerability scanning, and security control validation as recommended by NIST guidelines.
      * **Regulatory Compliance with GDPR & HIPAA:** Security auditing tools like Nessus ensure compliance with GDPR and HIPAA by identifying misconfigurations, enforcing data encryption, and monitoring access controls.
      * **Automated Security Reporting and Auditing:** Grafana and Zabbix generate detailed reports on security incidents, helping organizations track compliance with industry standards.

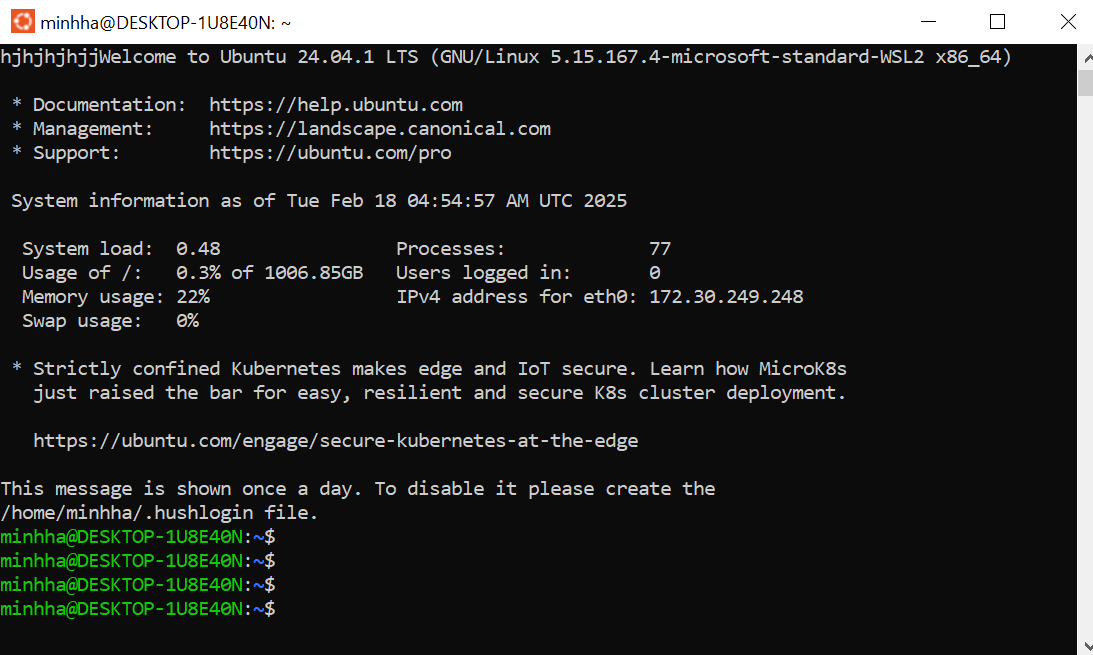
#### Emerging Trends

* + - * **AI-Driven Anomaly Detection:** Zabbix and OpenVAS are increasingly integrating AI and machine learning to detect unusual network behavior and predict potential security threats.
      * **Quantum-Safe Cryptography:** Future-proofing security measures by researching post-quantum cryptographic algorithms to secure sensitive network communications.
      * **IoT Security Challenges:** Monitoring IoT devices with Zabbix to detect vulnerabilities, ensuring real-time visibility and proactive threat mitigation.
      * **Cloud-Native Security Monitoring:** Using Grafana to analyze security logs from cloud environments, ensuring compliance with security policies and protecting against data breaches.
      * **Automated Threat Intelligence:** Enhancing Nessus and OpenVAS with automated security updates and real-time threat intelligence to identify zero-day vulnerabilities.

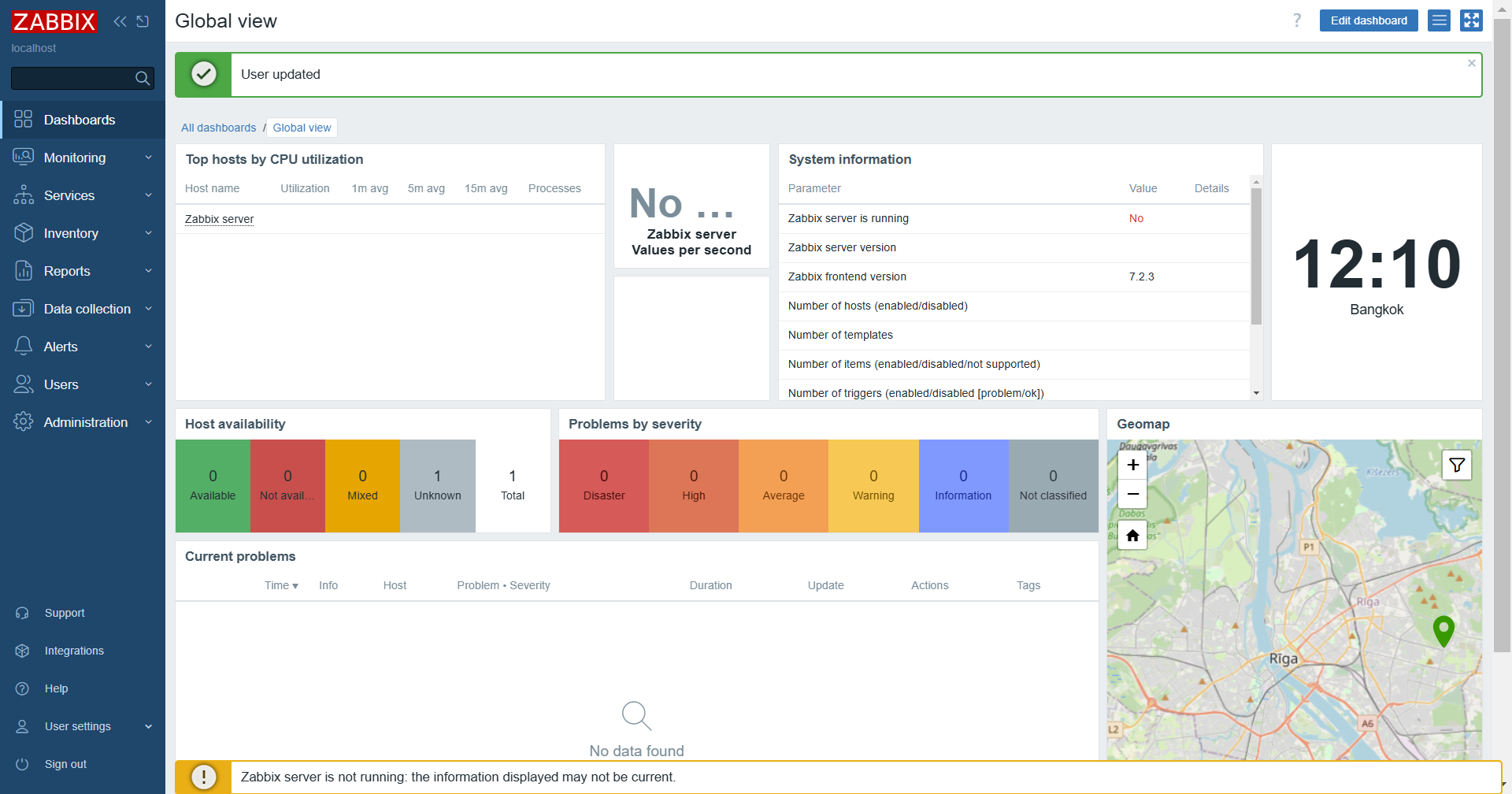
### Tools Utilized

#### Virtual environment design of Zabbix and Grafana.

In this project, **Ubuntu** is used as the primary operating system for deploying and testing network monitoring and security tools. By utilizing virtualization tools, we can establish a simulated network system, allowing us to assess monitoring performance and security vulnerabilities safely without impacting real systems.



*Figure 1: The interface Ubuntu*



*Figure 2: Zabbix interface*

2.2.2. Introduction to Grafana.

Grafana is an open-source platform for monitoring and observability that allows users to visualize, analyze, and understand their data through interactive dashboards. It supports various data sources, including Prometheus, InfluxDB, MySQL, PostgreSQL, and Zabbix. Grafana provides a user-friendly interface, powerful query editors, and customizable visualizations, making it an essential tool for infrastructure monitoring, application analytics, and operational intelligence.

Installing Grafana on Ubuntu

This section outlines the step-by-step process to install Grafana on an Ubuntu system.

Prerequisites

* Ubuntu 20.04 or later
* Root or sudo user privileges
* Internet connection
* **Zabbix Installed and Operational:** Grafana will retrieve data from Zabbix for visualization.

Step-by-Step Installation

1. **Install Grafana**

| sudo apt update  sudo apt install grafana -y |
| --- |

1. **Enable and Start Grafana Service**

| sudo systemctl daemon-reload  sudo systemctl enable grafana-server  sudo systemctl start grafana-server |
| --- |

1. **Verify Grafana Status**

| sudo systemctl status grafana-server |
| --- |

### Basic Configuration and Initial Setup

1. **Access Grafana Web Interface:**

Open a web browser and go to ***http://localhost:3000***. The default login credentials are:

**Username:** admin

**Password:** admin1

1. **Change Default Password:**

After the initial login, Grafana will prompt you to change the default password for security purposes.

1. **Add Data Source:**

Navigate to **Configuration** > **Data Sources** and click **Add data source**. Choose the desired data source and provide the required details for connection.

**Grafana**

Khởi Động grafana cùng với Ubuntu

Chạy trong ubuntu

Các câu lệnh :

sudo systemctl daemon-reload

sudo systemctl enable grafana-server

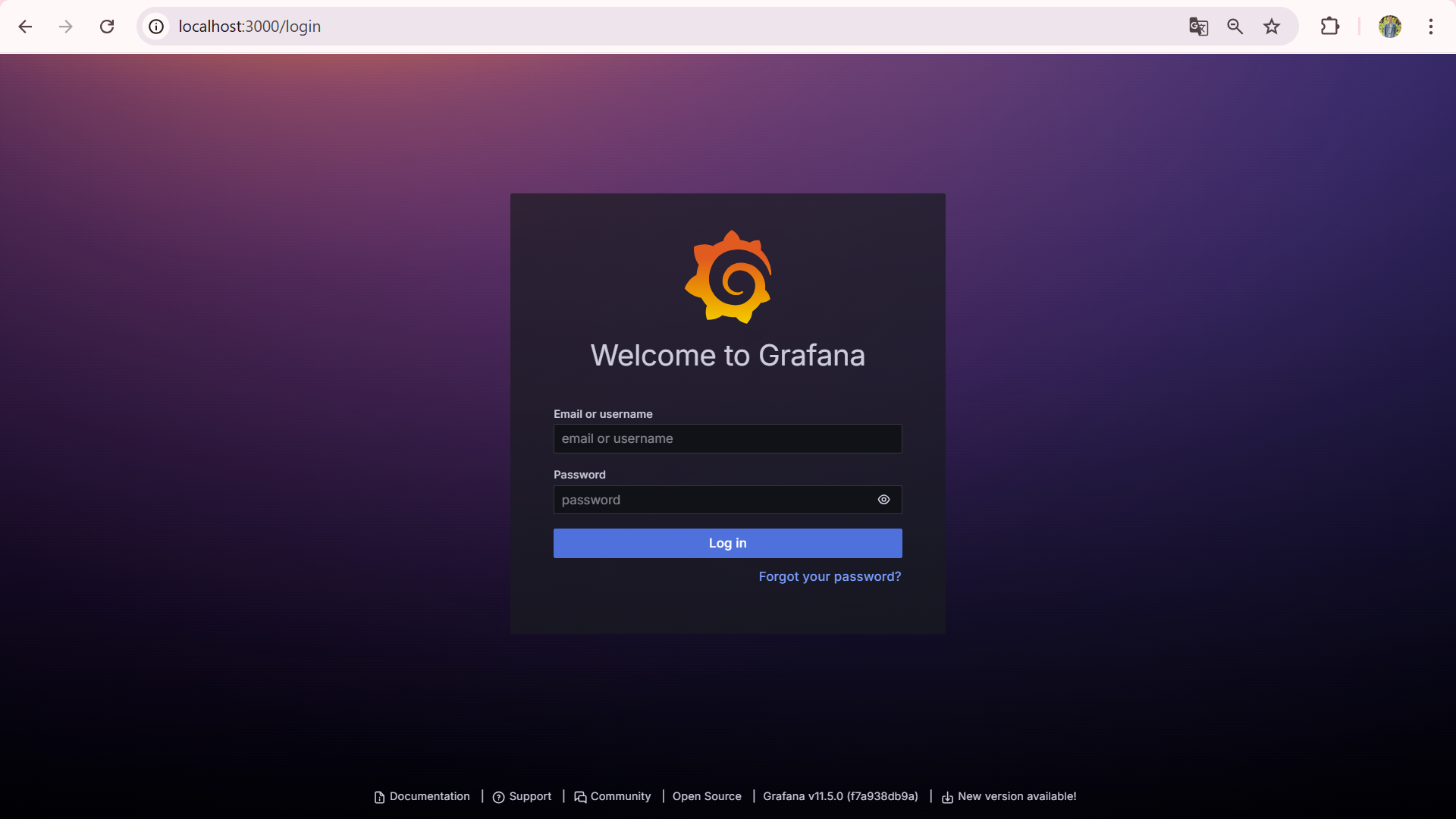
sudo systemctl start grafana-server

Kiểm tra trạng thái của Grafana : phải là active (enabled ) là ok

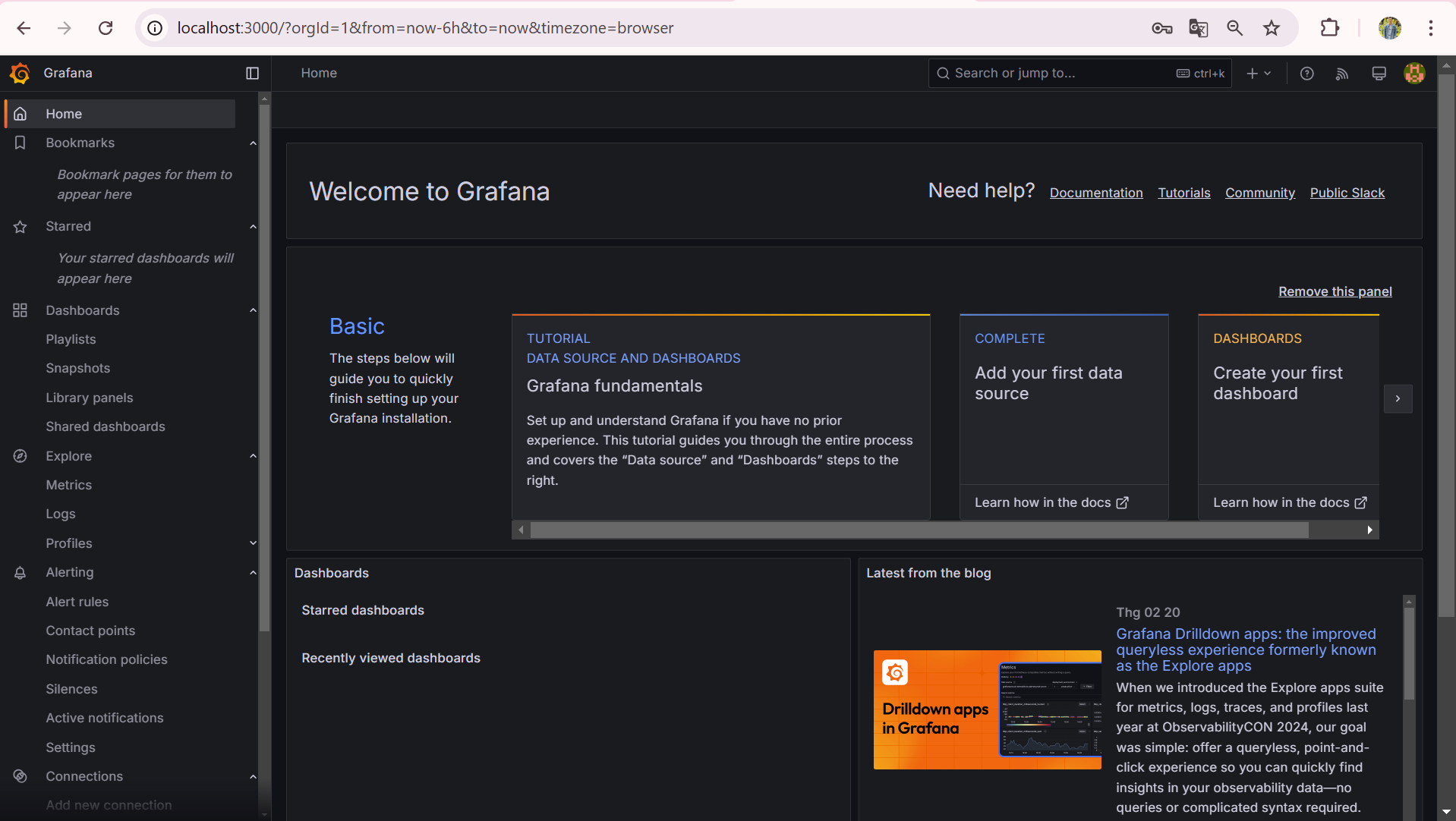
sudo systemctl status grafana-server

Truy cập vào trình duyệt và bấm :

<http://localhost:3000/login>



Admin 123456



#### Scanning software



*Figure 3: The TryHackMe Website*

# Chapter 3: The Architectural Design

### Survey of Current State and Requirements

#### Survey of Current State

* + - * Environmental system:
      * Network: LAN type, architectural network (intranet), system hierarchy network.
      * Operating systems and software: Operating systems being used (Windows, Linux, macOS), software and applications.
      * Web and mobile applications: Web and mobile applications connected to the system, especially online transaction applications, CRM, ERP.
      * Database: System administration database being used (SQL Server, MySQL, Oracle), system storage data, data protection method (backup, encryption).
      * Current security knowledge:
      * Security policies: Password management policy, data protection policy, access management and control, intrusion prevention.
      * Security tools: Firewall, IPS/IDS, encryption solutions, intrusion detection and prevention systems (IDS/IPS), anti-virus and malware protection tools.
      * Current threats:
      * Common vulnerabilities: Identify common vulnerabilities in the system (e.g., vulnerabilities in web applications, weak wireless networks, lack of software updates).
      * Past attacks: Review previous cyber attacks (if any) and their consequences.
    1. **Requirements**

Testing objective:

* + - * Identify vulnerabilities: Detect exploitable vulnerabilities and assess their severity.
      * System security assessment: Review the system's protective capabilities against threats, test the system's resilience against cyber attacks.
      * Regulatory compliance: Ensure that the organization complies with regulatory requirements and security standards such as PCI DSS, GDPR, HIPAA.

Scope and subject of testing:

* + - * Test subjects: Systems, networks, applications, mobile devices, or any resources that the organization wants to test.
      * Testing scope: Clearly define which system elements will be tested, avoiding unrelated resources.

Testing approach:

* + - * Black Box Testing: Testing without prior information about the system.
      * White Box Testing: Testing with complete detailed information of the system, including source code, documentation, and processes.
      * Grey Box Testing: Testing with certain access to information. Testing methods:
      * Testing techniques: Using various testing tools and methods such as network scanning, vulnerability exploitation, web application testing.
      * Simulated attacks: Conduct simulated attacks such as social engineering and denial of service (DoS) to test the system's protective capabilities.
      * Reports and recommendations:
      * Reporting requirements: Specific requirements for the test result report, including findings, severity levels, and remediation recommendations.
      * Time and format: Specify the time required for testing and the format of the documentation to be used.

Ethics and legality [3]:

* + - * Ensure there is legal permission from the organization before conducting the test.
      * Comply with ethical and legal principles related to penetration testing.
      * By surveying the current situation and clearly defining the requirements, organizations can conduct penetration testing effectively, better protecting their systems and assets against cyber threats.

### Network System Analysis

**Network System Analysis for Pentest** involves assessing the network structure, identifying vulnerabilities, and evaluating security measures to safeguard against potential attacks. Key steps include:

* **Information Gathering**: Scanning networks to identify devices, IP addresses, services, and network topology.
* **Network Structure Analysis**: Examining firewall configurations, NAT policies, and VLAN setups to prevent unauthorized access.
* **Service Security Assessment**: Checking configurations of network services (e.g., DNS, DHCP, web servers, mail servers) for vulnerabilities.
* **Vulnerability Scanning**: Using tools like Nessus and Nexpose to detect weaknesses in network configurations.
* **Review of Security Policies**: Assessing access management policies and device configurations to ensure compliance with security standards.

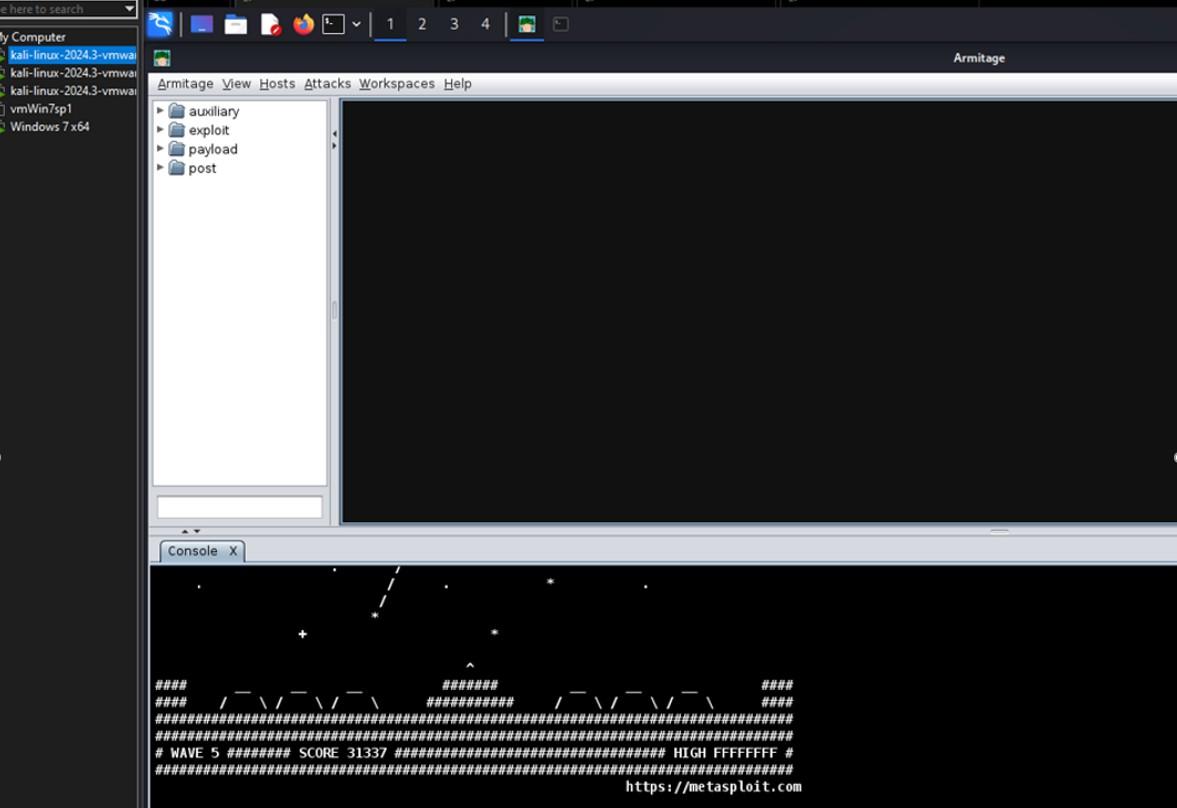
# Chapter 4: The Armitage Software

### Main Features of Armitage

#### Graphical User Interface:

Figure 4 shows us the highlights of the main screen of the Armitage application:

* + - * **User-Friendly Interface**: Armitage simplifies the complex commands and configuration of Metasploit, making it accessible to both beginners and experienced users.
      * **Drag-and-Drop Functionality**: Allows users to easily select exploits, payloads, and targets using drag-and-drop.
      * **Integrated with Metasploit**: Provides access to all the capabilities of Metasploit, including vulnerability scanning, exploitation, and post-exploitation tasks.

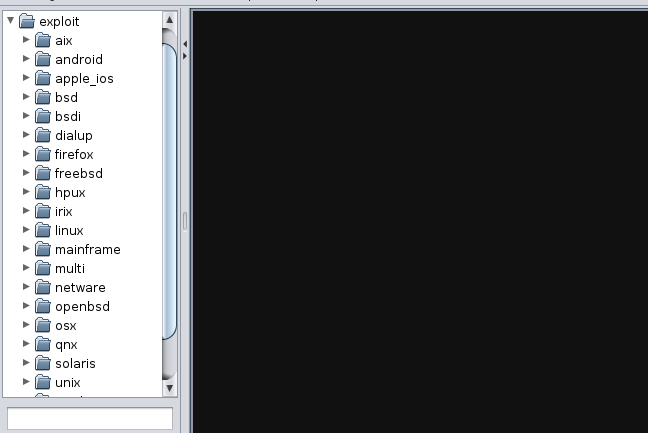


*Figure 4: Graphical User Interface*

#### Exploit Management:

Armitage also has an exploitation management panel on the left corner of the software as shown in Figure 5:

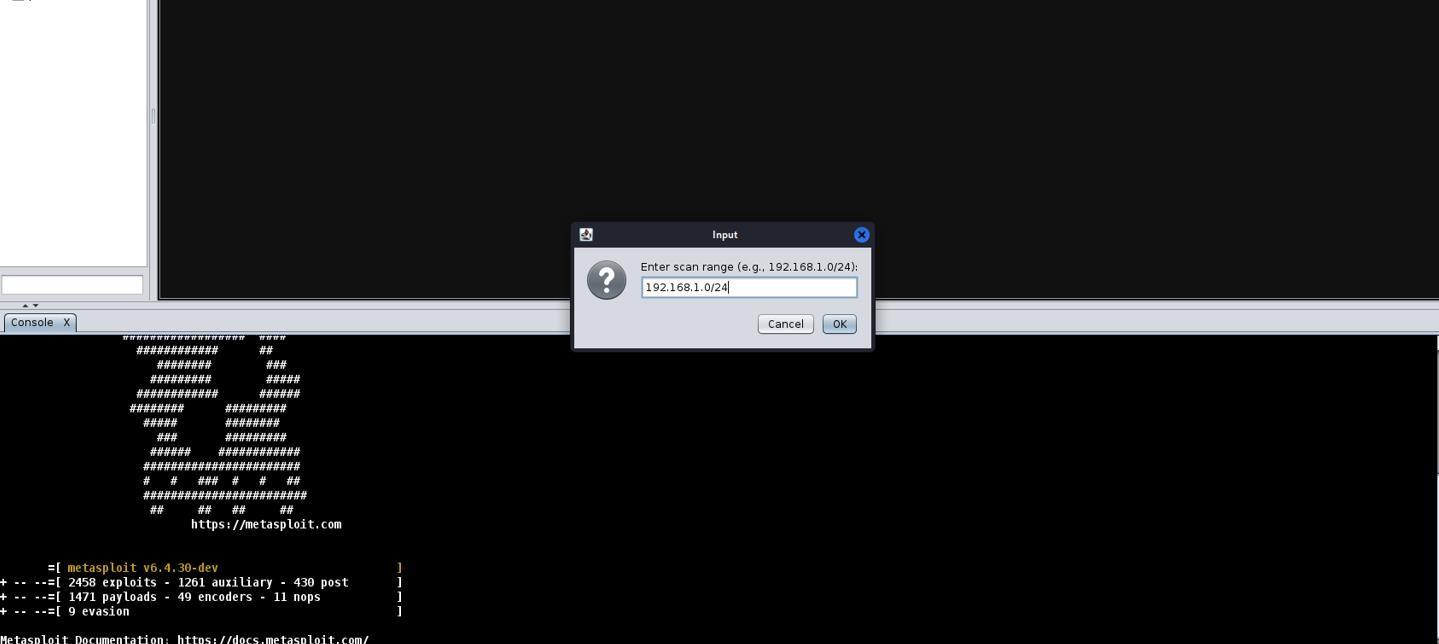
* + - * **Easy Selection of Exploits**: Lists available exploits in a simple, searchable interface, making it easy to find and use specific tools.
      * **Visualization of Vulnerabilities**: Visual representation of vulnerabilities in a network, making it easier to understand the attack surface.



*Figure 5: Exploit Management*

#### Target Scanning and Management:

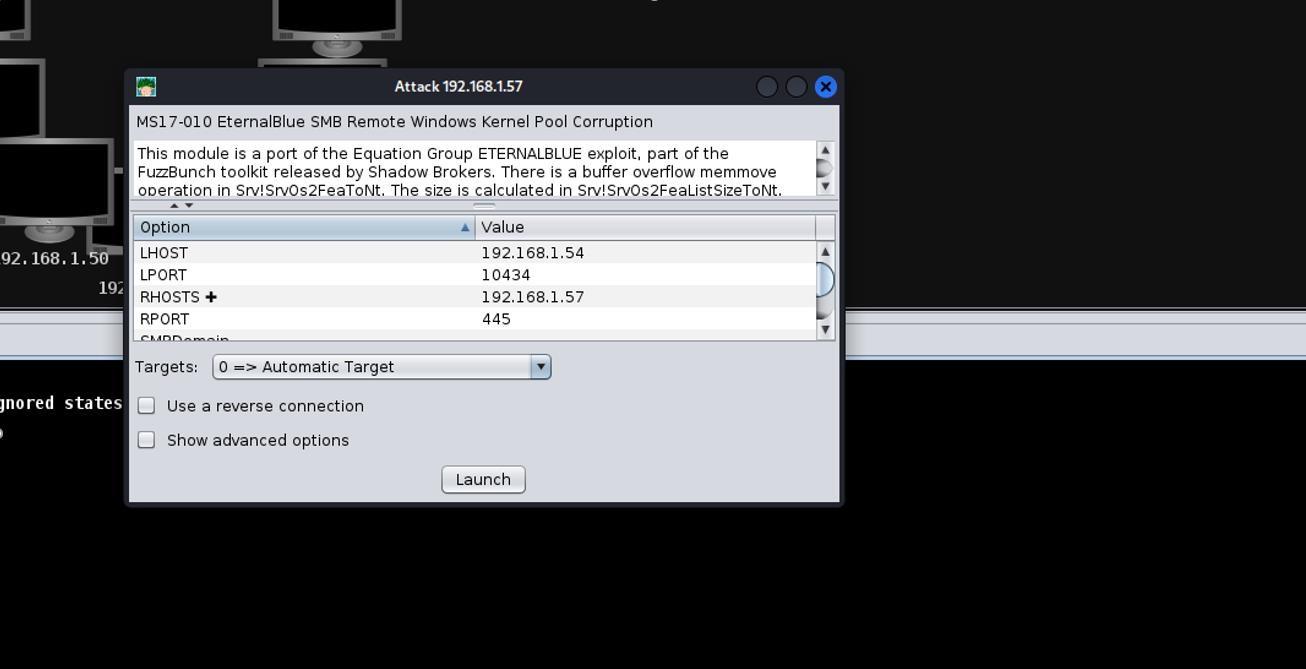
* + - * **Network Scanning**: Can scan local networks or specific IP ranges to identify potential targets like the Figure 6.
      * **Session Management**: Provides a session management dashboard to monitor active sessions and interact with compromised systems.



*Figure 6: Network Scan Input Dialog*

#### Payloads and Post-Exploitation:

* + - * **Payload Creation**: Users can select and create payloads for specific exploits using a visual interface as shown in Figure 7.
      * **Post-Exploitation Tools**: Access to tools that help in maintaining access to compromised systems, gather information, and execute commands remotely.



*Figure 7: Exploit Configuration Window*

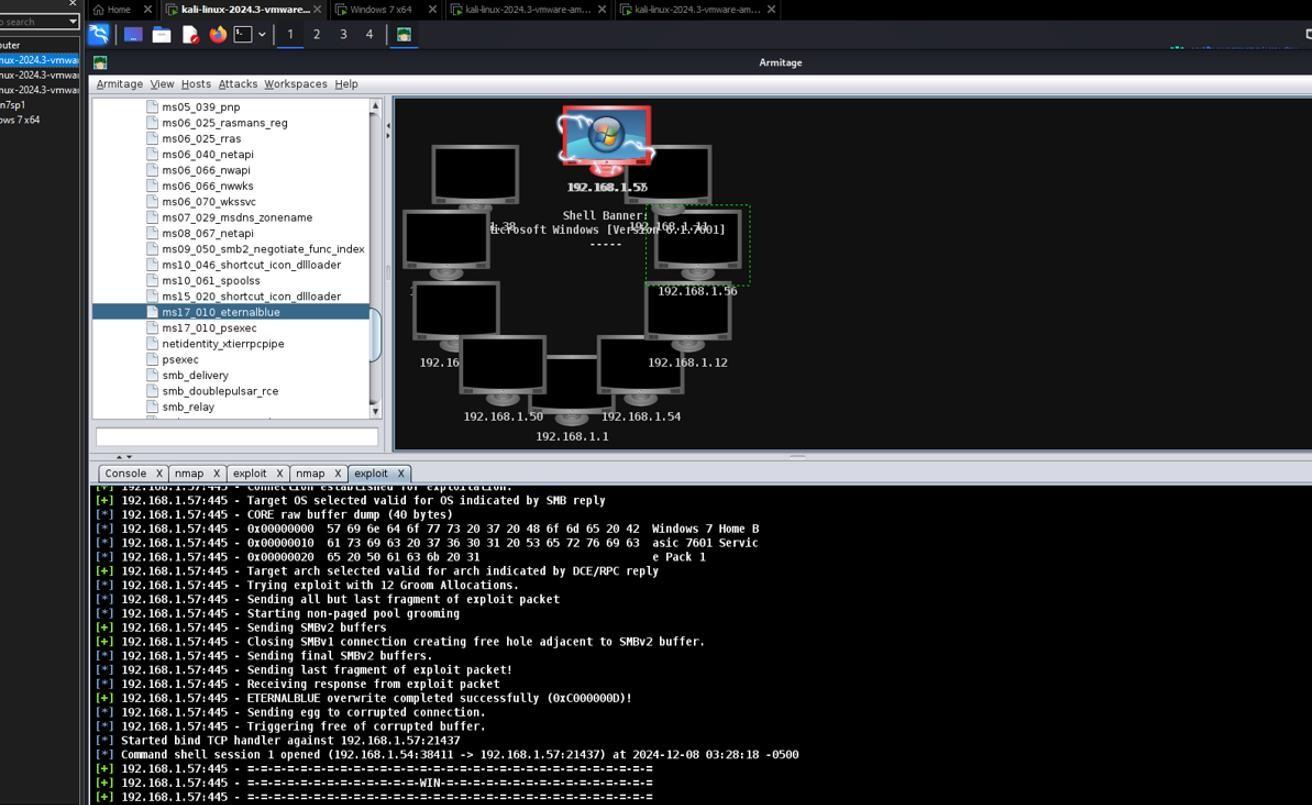
#### Visual Interaction and Control:

* + - * **Attack Simulation**: Simulates the attack process visually, allowing users to see the steps of an attack in real-time.
      * **Graphical Interaction with Targets**: Provides the ability to interact with and control compromised systems through a visual interface.
      * Session Tracking and Control:
      * **Session Dashboard**: Tracks the status and details of active sessions, allowing users to maintain and manage multiple sessions simultaneously.
      * **Session Control**: Control sessions, interact with system shells, and execute commands remotely from the GUI.

#### Scripted Attacks:

Armitage also supports users to attack by exploiting vulnerabilities as shown in Figure 8:

* + - * **Automated Attack Sequences**: Ability to create and run custom attack scripts using a built-in scripting language.
      * **Automation of Repetitive Tasks**: Simplifies the execution of common tasks in penetration testing by automating steps within the attack chain.



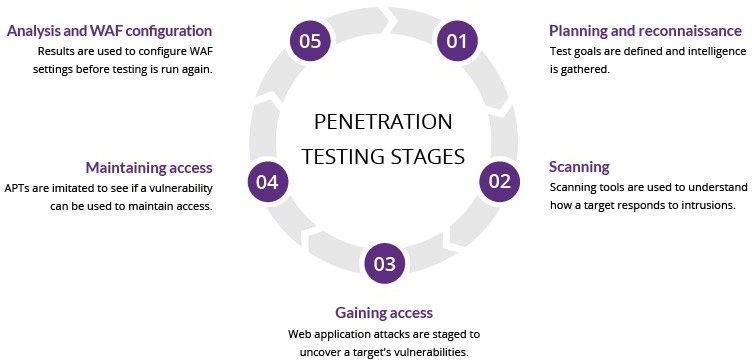
*Figure 8: Armitage Attack Visualization*

#### Real-Time Monitoring:

* + - * **Live View of Attacks**: Monitors the progress of attacks in real-time, providing feedback on the status of each exploit or payload.
      * **Logging and Reporting**: Logs activity and can generate reports of the penetration test findings.

### System Implementation

The Figure 9 shows Pentest system deployment process: [4]



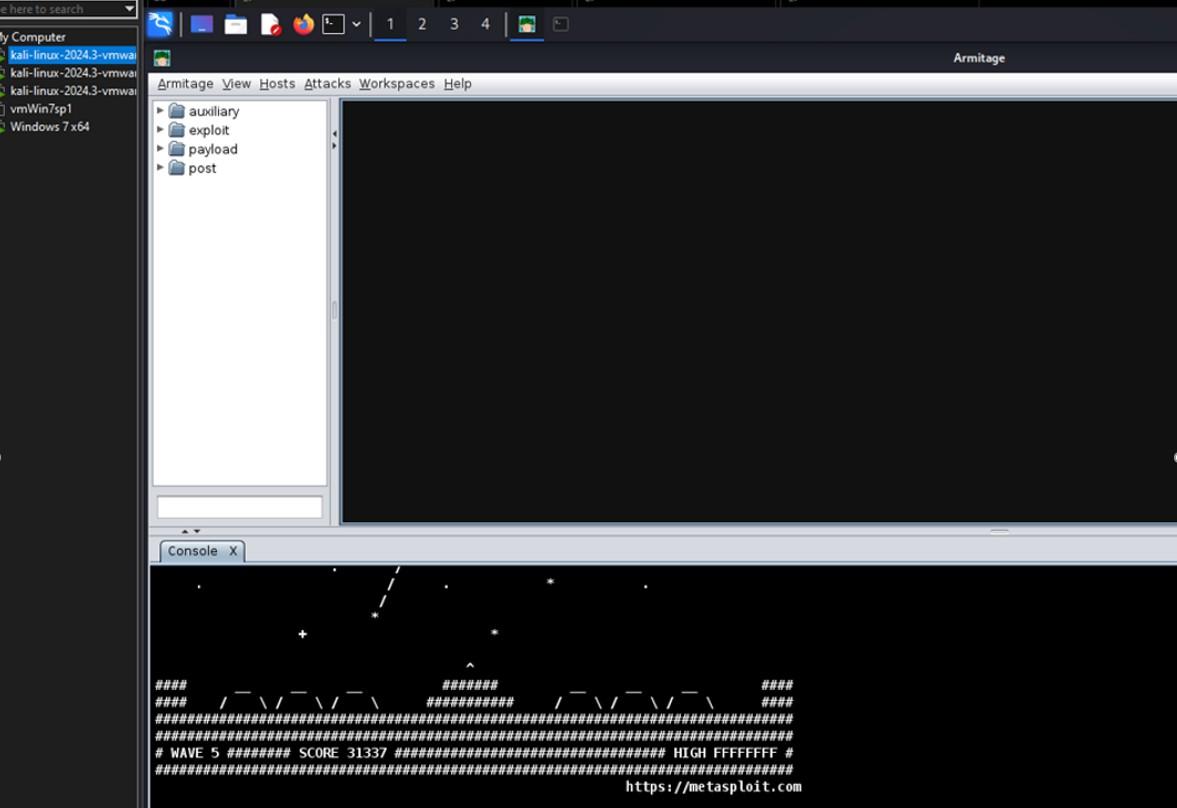
*Figure 9: Penetration Testing Stages*

# Chapter 5: Project Performance and Evaluation

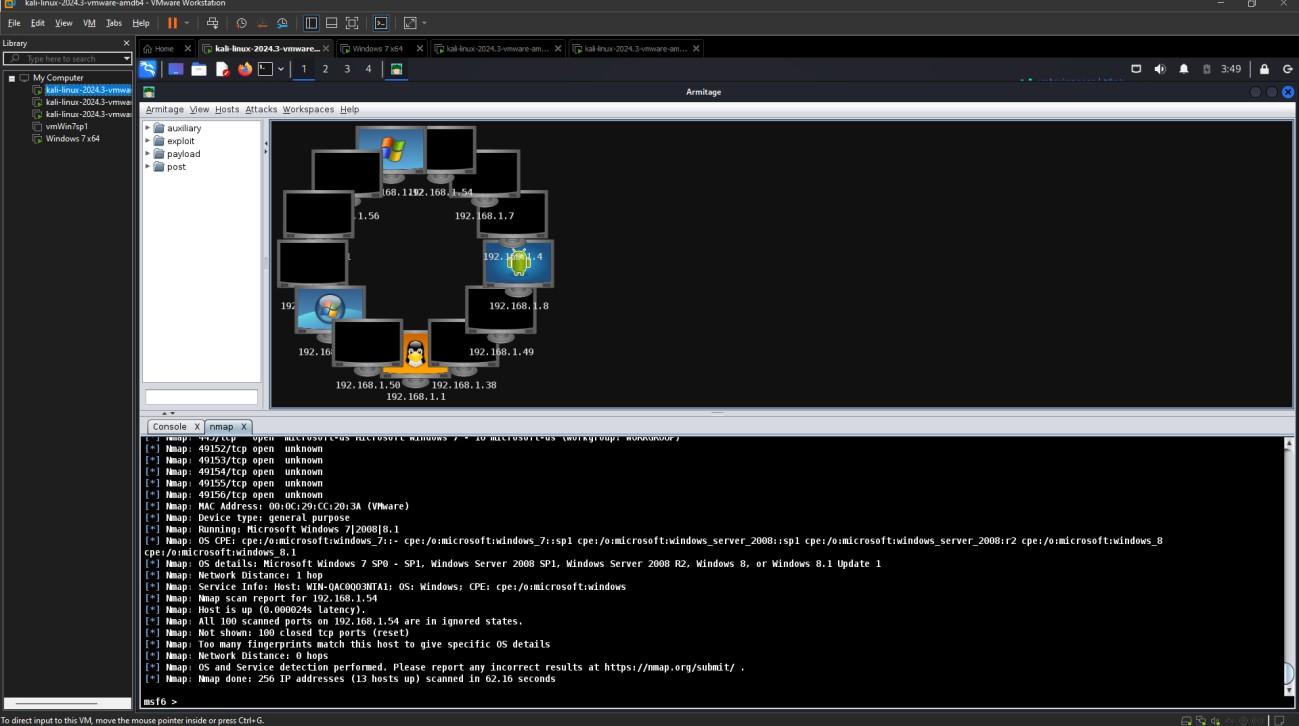
### Testing Programs and Results

#### Testing programs

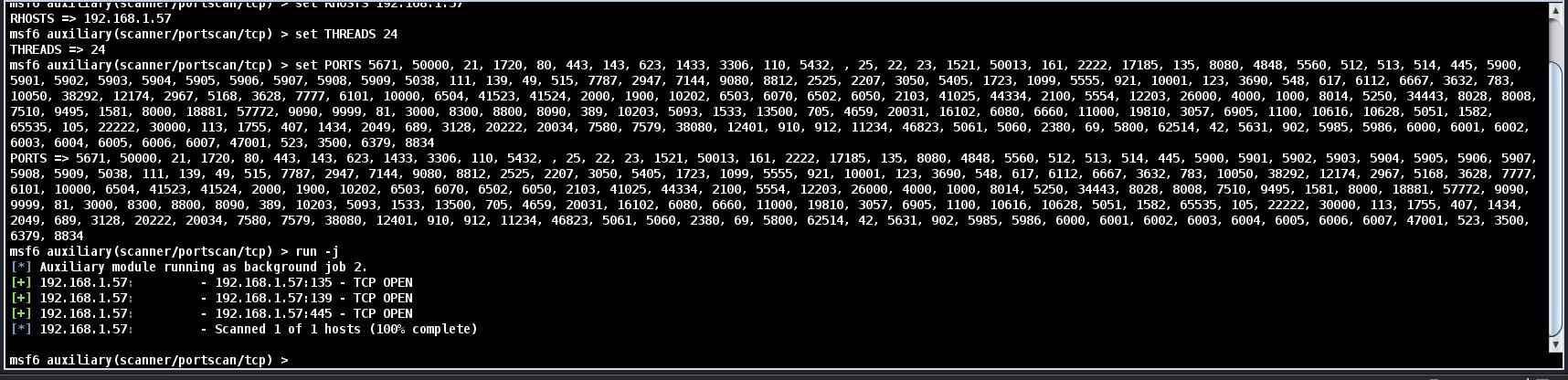
First, when opening Armitage, it initially does not show us the devices on our network as shown in Figure 10. After taking a few steps to scan to my network specifically 192.168.1.0/24, it returned me a multitude of devices connected to the network as shown in Figure 11. From there, I can start scanning for existing vulnerabilities on the computer specifically here I choose a computer with a windows operating system. So the results like Figure 12 are a lot of TCP which is open. And then I can able to connect as shown in Figure 13. Furthermore, based on the opened ports above, I decided to attack the Windows machine through ms17\_010\_enternalblue. Finally as shown in Figure 14, I received that the computer was attacked.



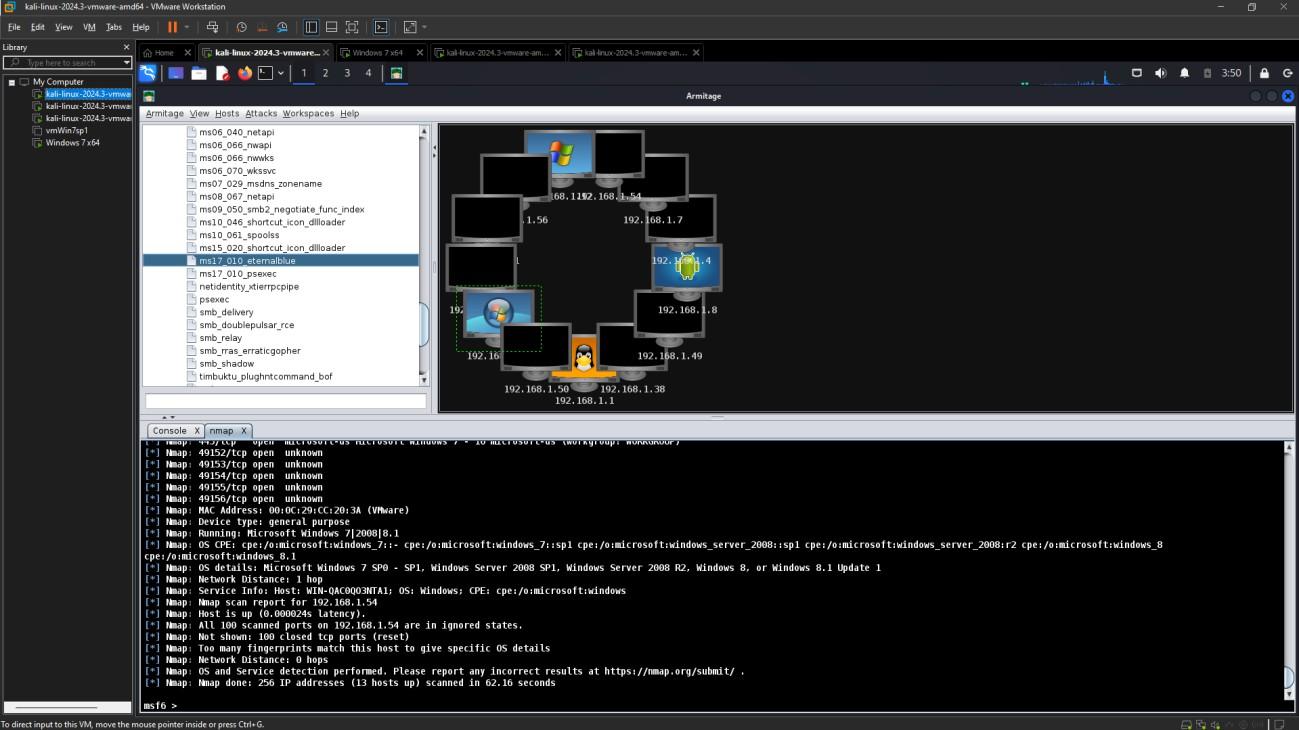
*Figure 10: Interface of Armitage*



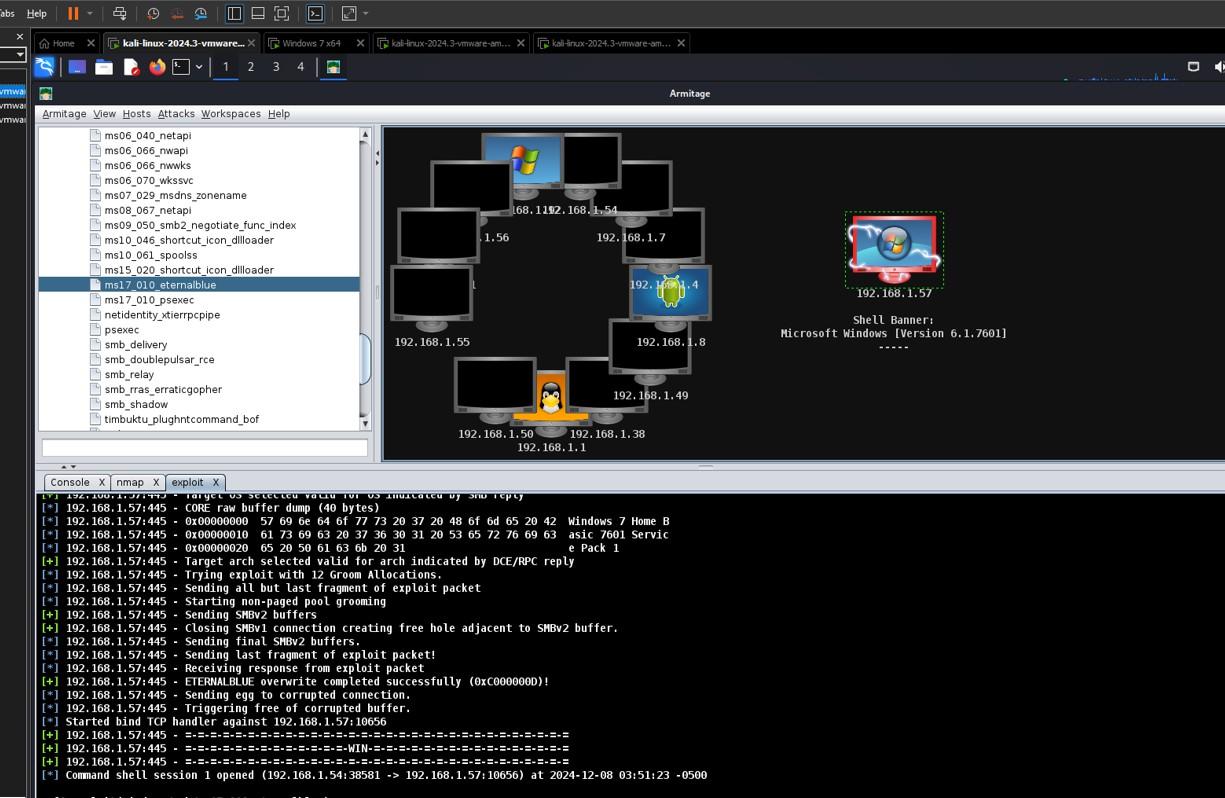
*Figure 11: After the scanning*



*Figure 12: TCP Open*



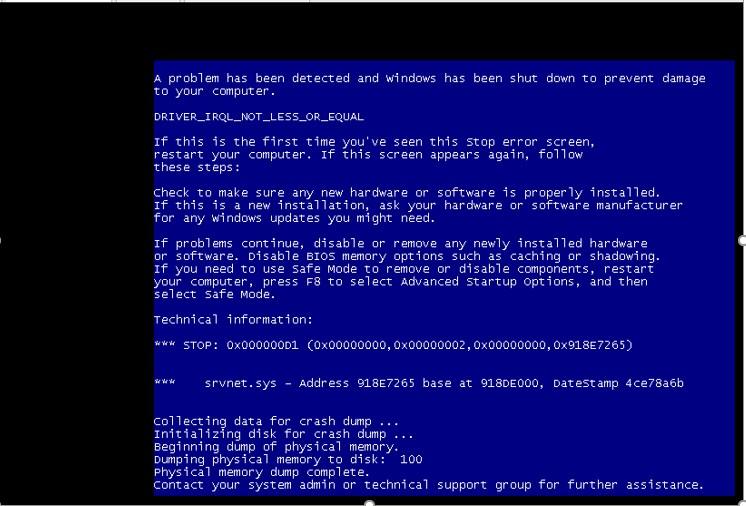
*Figure 13: Access broken hole*



*Figure 14: Attack*

#### Results

From Figure 15, I realized that the hacked computer had shut down and showed a blue screen and I couldn't do anything on it. So, Windows machine was reboosted:

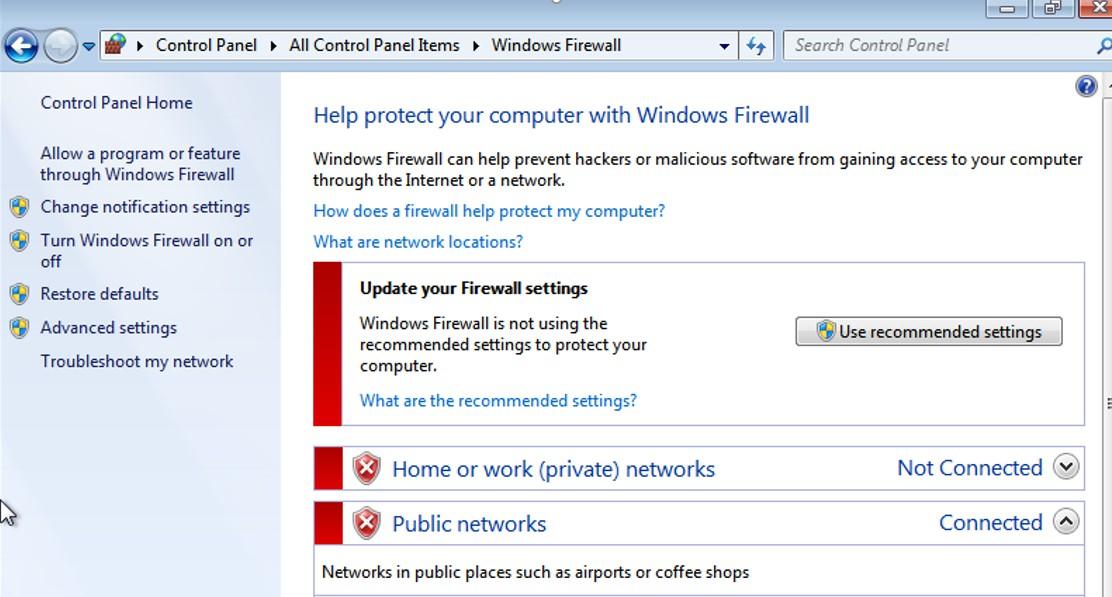


*Figure 15: Windows machine was reboosted*

### Solutions for Mitigation

I also have some measures to prevent being attacked as follows:

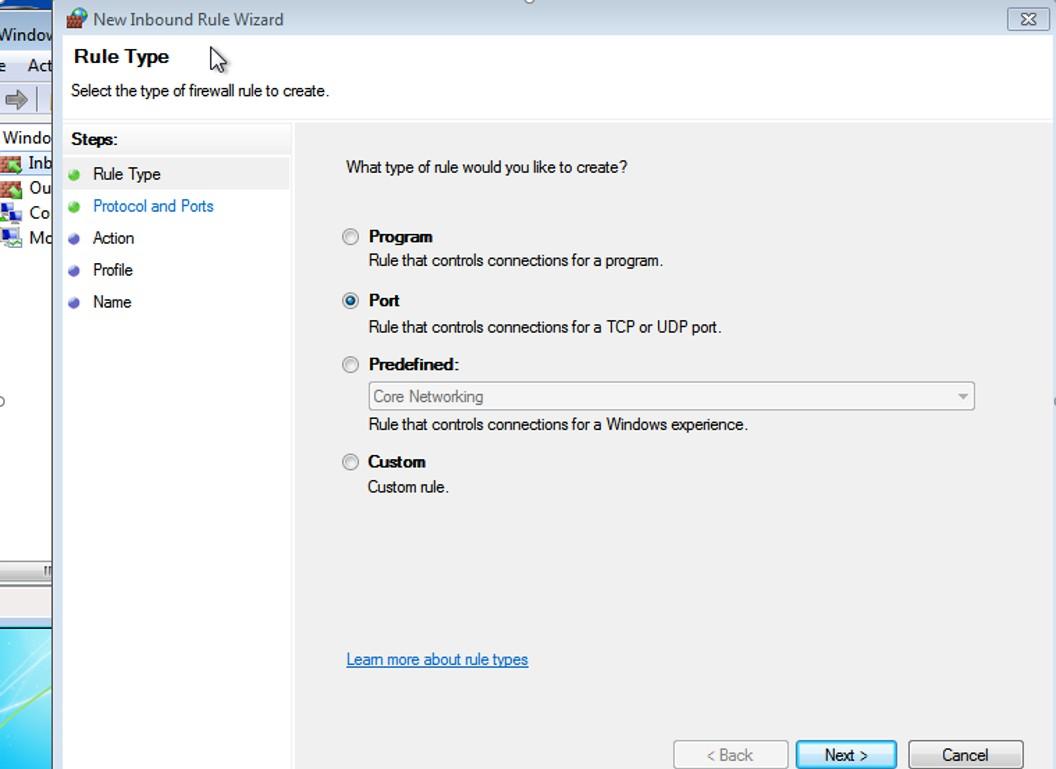
* First, check to see if the firewall is turned on or not. As shown in Figure 16, the firewall is not yet open. It was then activated as shown in Figure 17:
* Secondly, I have a few steps to access the operating system to turn off open ports on the machine as shown in Figure 19 and 20.



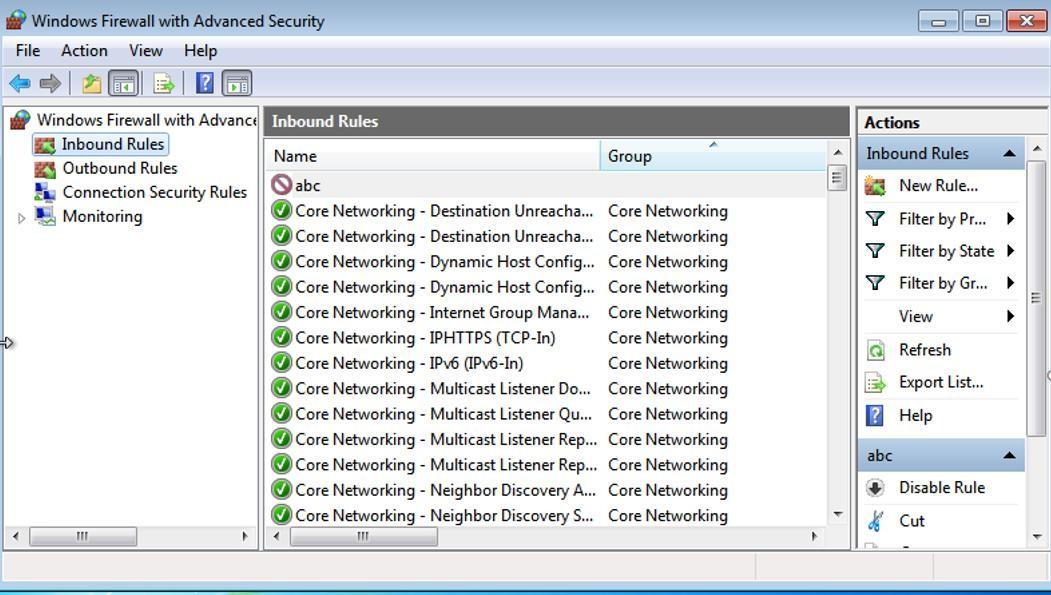
*Figure 16: Check whether the firewall*



*Figure 17: The Windows Firewall state is On*



*Figure 18: New Inbound Rule Wizard*



*Figure 19: Windows Firewall with Advanced Security*

#### Future Development Directions

* Based on the above, we can improve the security of a computer system in the future. Avoid cases where data is stolen or the computer system is paralyzed by viruses/malicious code. A good security solution will minimize management costs and avoid financial loss if a leak occurs.
* Fix vulnerabilities and adopt secure coding practices.
* Integrate security tools (e.g., SAST, DAST) into the development lifecycle.
* Establish patch management and monitoring systems.
* Automate regular security testing and refine incident response plans.
* Train employees on secure practices and run attack simulations.
* Promote knowledge sharing and collaboration across teams.
* Launch bug bounty programs and foster a security-first culture.
* Regularly schedule pentests and expand their scope over time.
* Deploy advanced tools (e.g., WAF, AI-driven monitoring).
* Align with compliance standards and update security policies.
* Integrate DevSecOps and conduct threat modeling workshops.
* Encourage cross-functional teams to address security holistically.

# Chapter 6: Conclusion

### Summary

We start by studying the documents about the basic theory of Pentest (Penetration Testing), then look up the software that supports Pentest.

First, deploy a virtual environment (VMware) to create a basic network system consisting of many computers with many different operating systems (Kali Linux, MacOs). Next, install the Armitage software supported within the Kali Linux operating system, then we rely on Armitage to scan the machines on the network. Furthermore, we will scan for vulnerabilities before entering and attacking a computer. So, by exploiting open ports, we have caused the attacked computer to shutdown. And the final goal we need to achieve is to provide solutions to overcome existing network vulnerabilities and enhance security performance in the future.

### Conclusion

Through this project (CSE 320), I realized that information security on the network is absolutely necessary for any information system in any field. Being attacked leads to loss or disclosure of information, leading to unpredictable losses for both an individual and a company.

During the research process, I learned how to deploy a network system and found holes, based on which I came up with solutions to fix them for the present and the future.

# Reference

[1]. Cybersecurity threats on the rise in Việt Nam's SMB sector: reports, from [https://vietnamnews.vn/economy/1594537/cybersecurity-threats-on-the-rise-in-viet-nam-s-smb-](https://vietnamnews.vn/economy/1594537/cybersecurity-threats-on-the-rise-in-viet-nam-s-smb-sector-reports.html) [sector-reports.html](https://vietnamnews.vn/economy/1594537/cybersecurity-threats-on-the-rise-in-viet-nam-s-smb-sector-reports.html)

[2]."Cryptography and Network Security: Principles and Practice", from [https://dl.hiva-](https://dl.hiva-network.com/Library/security/Cryptography-and-network-security-principles-and-practice.pdf) [network.com/Library/security/Cryptography-and-network-security-principles-and-](https://dl.hiva-network.com/Library/security/Cryptography-and-network-security-principles-and-practice.pdf) [practice.pdf](https://dl.hiva-network.com/Library/security/Cryptography-and-network-security-principles-and-practice.pdf)

[3]. "Computer Networking: A Top-Down Approach" - James F. Kurose, Keith W. Ross, from <https://qige.io/network/Kurose-7.pdf>

[4]. Penetration testing, from [https://www.imperva.com/learn/application-security/penetration-](https://www.imperva.com/learn/application-security/penetration-testing/) [testing/](https://www.imperva.com/learn/application-security/penetration-testing/)

[5]. Stallings, W. (2000). \*Network Security Essentials: Applications and Standards\*. Pearson Education.

[6]. Stuttard, D., & Pinto, M. (2011). \*The Web Application Hacker's Handbook: Discovering and Exploiting Security Flaws\* (2nd ed.). Wiley Publishing, Inc.

[7]. Faircloth, J. (2014). \*Practical Network Security\*. Syngress.