### **1. Introduction**

The dynamic nature of cybersecurity necessitates ongoing vigilance and proactive strategies to protect digital systems from emerging threats and vulnerabilities. **Penetration testing**, also known as **ethical hacking**, plays a crucial role in this effort by providing organizations with valuable insights into the strength and defenses of their security measures. By simulating real-world cyber-attacks, penetration testing evaluates the resilience of a system’s defenses, identifying potential weaknesses before malicious actors can exploit them.

### **1.1 Objective and Scope**

There will be a network topology with different machines in it. This report details apenetrationtest conducted on the target system on the network **pWnOS**, with the primary goal of assessing its security posture. The test aims to:

* Identify security weaknesses through simulated attacks.
* Exploit vulnerabilities to demonstrate potential risks.
* Provide actionable remediation strategies.

The scenario assumes an **internal user compromise**, mimicking how attackers might infiltrate a system. The ultimate objective is to gain **root access** and locate a file named **proof.txt** (if present) in the root directory, serving as evidence of successful exploitation.

### **2. Summary and Recommendations**

#### **2.1 Summary**

The penetration testing process revealed several critical vulnerabilities within the **pWnOS** environment, exposing potential security risks. These issues spanned from weak authentication mechanisms to out-of-date software, both of which create loopholes for unauthorized privilege and system threat.

One predominant and concerning finding was the **lack of strong user access controls and weak authentication/password policies**, which posed a significant security risk. The **Webmin interface**, in particular, was not adequately protected by strong authentication methods or **Multi-Factor Authentication (MFA)**, making it vulnerable to unauthorized access. This highlights the need for **robust privilege/access control measures**, including MFA and strict password management practices, to enhance security.

Additionally, the presence of **exploitable vulnerabilities** in outdated system services was another major concern. These loopholes could be leveraged by attackers to gain access to sensitive data or compromise the system's integrity.

Another critical issue was the **absence of persistent security patches and updates**, which significantly weakened **pWnOS's** security posture. Failing to promptly deal with known vulnerabilities in a timely manner increases the risk of systems being successfully compromised by attackers, making the system more susceptible to threats.

#### **2.2 Recommendations**

To enhance the security of **pWnOS**, the following measures should be implemented:

* **Consistent Security and Vulnerability Assessment**: Conduct **frequent security audits** and vulnerability assessments to proactively identify and address weaknesses before they can be exploited.
* **Strengthen Access Controls & Implement MFA**: Enforce strict access control policies and require **Multi-Factor Authentication (MFA)** for sensitive services like **Webmin** to prevent unauthorized access.
* **Enhance Security Awareness Among Users**: Provide **ongoing cybersecurity training** to system users, emphasizing best practices such as locking screens, securing workstations, and turning off computers before leaving work to minimize security risks.
* **Enforce a Strong Password Policy**: Require passwords to be **10–12 characters long**, containing **both uppercase and lowercase letters**, at least **one special-case character, and a digit**. When combined with MFA, this will significantly reduce the likelihood of credential-based attacks.
* **Ensure Regular Updates & Patch Management**: Keep all system services and applications updated to protect against known security threats and vulnerabilities.

### **3. Methodology**

There are several recognized **penetration testing methodologies**, each providing structured guidelines for conducting security assessments. According to **Teaganne Finn of IBM (2024)**, the five most widely used methodologies include:

1. **Open-Source Security Testing Methodology Manual (OSSTMM)**
2. **Information System Security Assessment Framework (OISSG)**
3. **Open Web Application Security Project (OWASP)**
4. **National Institute of Standards and Technology (NIST)**
5. **Penetration Testing Execution Standard (PTES)**

For this penetration test, the **OSSTMM methodology** was selected due to its **scientific approach** and adaptability. This methodology provides well-structured, accessible guidelines for testers, ensuring a thorough and systematic assessment.

The specific approach followed in this assessment includes:

1. **Information Gathering** – Collecting details about the target system to identify potential entry points.
2. **Scanning and Mapping** – Analyzing the network structure and identifying active hosts.
3. **Enumeration** – Extracting information about services, users, and configurations.
4. **Gaining Access** – Exploiting vulnerabilities to penetrate the system.
5. **Privilege Escalation (or Denial of Service)** – Elevating privileges to gain deeper access or simulating a **Denial of Service (DoS) attack** to assess system resilience.

This structured methodology ensures a comprehensive evaluation of **pWnOS's** security weaknesses, helping to identify and mitigate risks effectively.

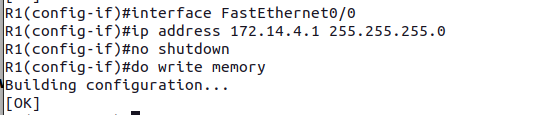
### **4. Network Design and Topology**

The Apporto environment was used to set up the network. Following the specification guide, a switch, a router (c7200), windows machine, kali machine and pWnOS were combined to form a network. The router was configured to act as the DHCP (Dynamic Host Configuration Protocol) server with the IP block 172.16.4.0/24. The following outlines how this was done:

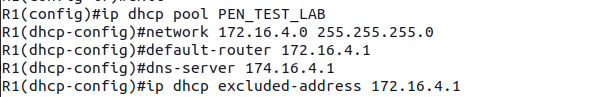
**Configuring the router terminal**



**Assigning the router interface**

ex

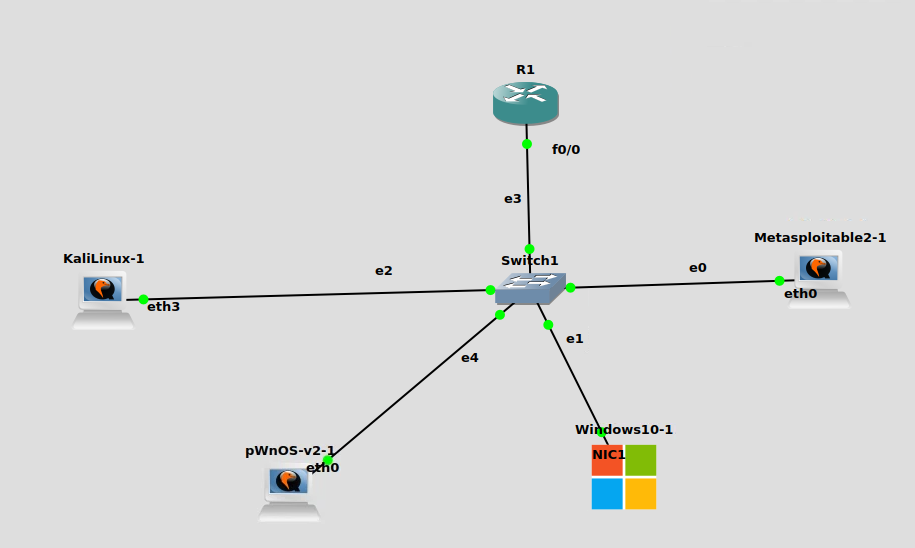
**Configuring the DCHP pool**



**Saving the configuration**



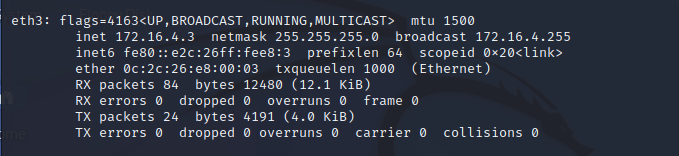
**Final network design and set up**



Switch information:  

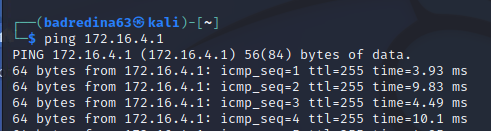

### **5. Information Gathering**

With the network topology done, the tester set up the kali machine to be used as the attaching VM, by creating and configuring the user as **badredina63**. The thing the tester did was verify if the router and network is set up properly and actively running. To do this, the IP of the target machine (kali) was checked with the command **ifconfig**, and it showed **172.16.4.3** as expected because from the network diagram, it was configured on **eth3** to the switch.

Verify on kali  
  


Now, to test if the DHCP server is up, a ping was used which successfully received messages from the router

Ping and test server:

  
  
**all devices and mac addresses**:  
0c:8e:c2:06:00:00 - windows

0c:1a:1a:87:00:00 - pwnos

0c:fc:57:b2:00:00 – metasploitable

With the preliminary verification above confirmed, it is time to discover the machines in the network to find the target. The command below was used to achieve this:

Table 1 (discovering IPs in the network)

|  |
| --- |
| sudo netdiscover -r 172.16.4.0/24 |

The **netdiscover** command requires superuser (**sudo**) permissions to grant privilege to network associated information that normal users are not permitted to. “**netdiscover**” is a CLI (Command Line Utility) that comes with kali to discover all devices on the network. The -r flag details the range of IP addresses configured on the network (172.16.4.0/24). The figure below shows the result after executing the command above:  
   
pic here:

After the tabular list from the result, the reporter identified the target is the machine with MAC address as 3423 and IP address as 234234. To verify if this is reachable, a **ping** was done which confirmed it is up as seen below:

With this discovery, the next step was to collect information about the target machine. The aim is to remain unknown and stealth in collection the information about the device on the network so as to reduce any traces and suspicions on the firewalls. To achieve this, nmap command was used with the flag “-sS”.

Table 2 (being stealth in gathering info)

|  |
| --- |
| sudo nmap -sS 192.168.1.78 |

The command shows opened ports running on the target VM.

Pic here:  
Nonetheless, collection information about opened and running ports is not enough, more information such as ssh-host key information, authentication level and operation system is needed about the target. To achieve this, the -A flag is used in combination with nmap. To additionally achieve non-suspicions and quietness in the network, random 100 IP addresses were spawned which will act as a decoy, making it much harder to reveal the actual IP address collecting the information. The command **-D RND:100** was used to achieve this.   
  
Table 3 (detailed information)

|  |
| --- |
| sudo nmap -A -D RND:50 192.168.1.78 |

The result of this is presented below:

|  |  |
| --- | --- |
| Target VM Details : pWnOS | |
| IP Address |  |
| Domain Name |  |
| Operating System |  |
| Host status |  |
| Authentication Level |  |
| Traceroute |  |
| Opened Ports |  |
| Ssh-host key info: |  |

### **5. Scanning and Mapping**

After information gathering, the target device had its port opened for the following numbers: