****

|  |
| --- |
| *BS Cyber Security Department AU* |

|  |  |
| --- | --- |
| **Registration ID** | **233026** |
| **Submitted By** | **Muhammad Sohaib Rafiq** |
| **Submitted To** | **Iram Fatima Hashmi** |
| **Date of Submission** | **12/02/2024** |
| **Assign NO.** | **2 & 3** |
| **Subject** | **NS LAB** |

## **Assignment Report Structure**

### 1. VM Name and Source

**Target VM**: Metasploitable2  
**Source**: https://sourceforge.net/projects/metasploitable/

**1. Set Up the Network:**

### 🔧 **1. Set Up the Network**

You need **Kali (host)** and **Metasploitable (VM)** on the **same network**:

* Open **VirtualBox / VMware > Metasploitable VM > Settings > Network**
* Set **Adapter 1 = Bridged Adapter** or **Host-Only Adapter**
* Do the same for Kali if it's a VM. If Kali is native (host OS), it’s already ready.

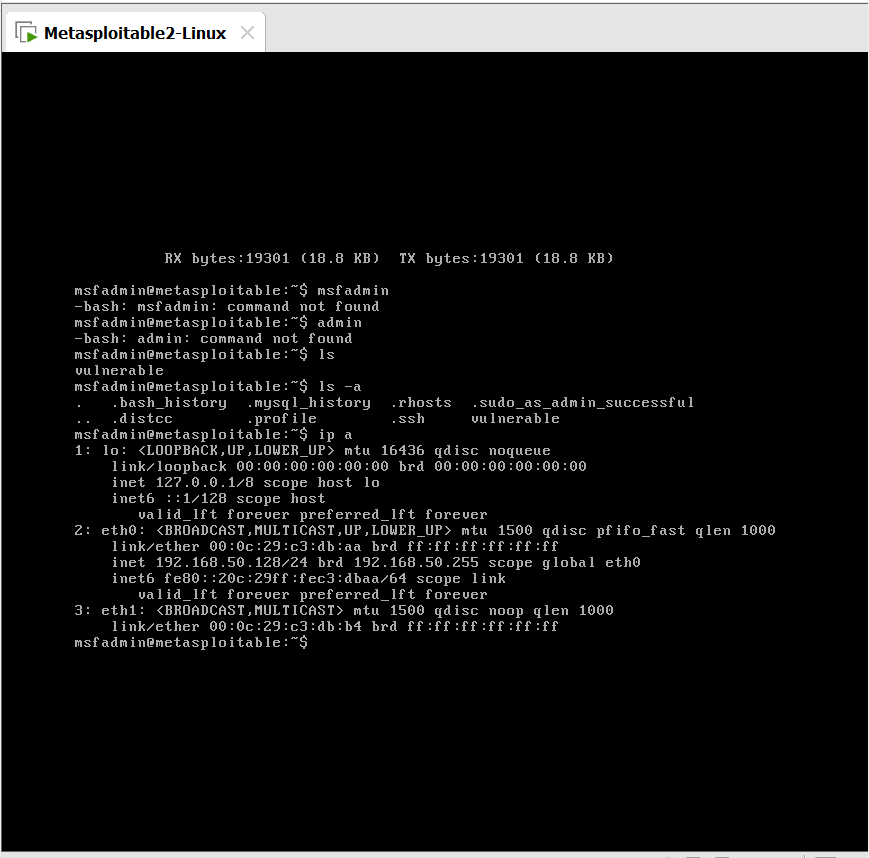
#### ✅ Verify:

On **Windows 10**, run:

cmd

Copy code

Ipconfig

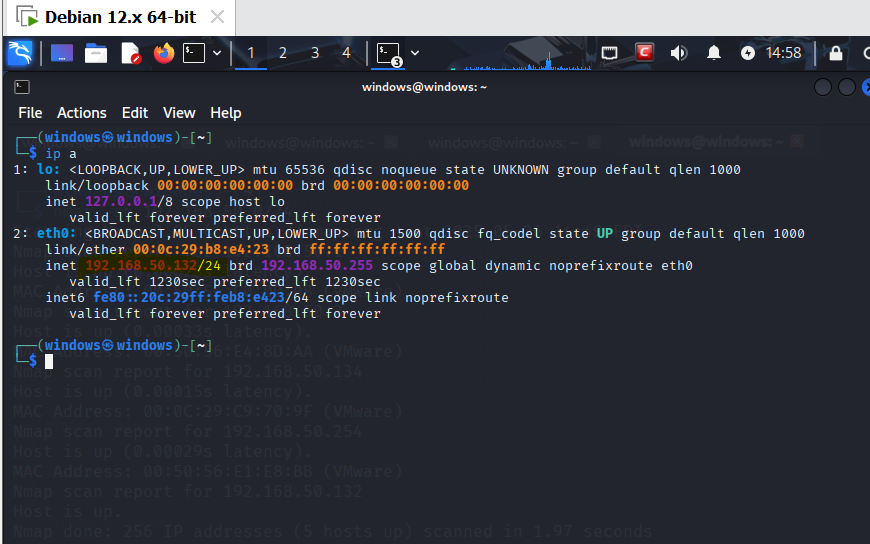


On **Kali**, run:

bash

Copy code

ip a



Make sure both are on the same subnet (e.g., 192.168.50.0/24)

So the network is correctly established

### 🔍 **2. Reconnaissance & Vulnerability Scanning**

#### 🔎 2.1 Discover Target IP:

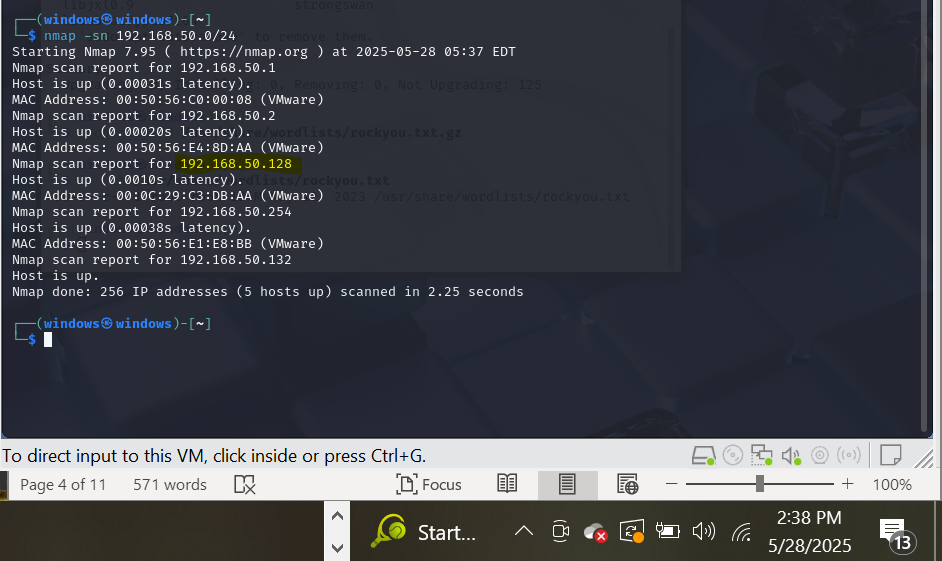
From Kali:

bash

Copy code

nmap -sn 192.168.50.0/24

Find the IP of your Windows 10 target 192.168.50.142

****

.

### ✅ Step: Scanning the Target System

To begin identifying vulnerabilities on the Windows 10 machine, I used **Nmap**, a powerful network scanning tool. My goal was to discover which services were running on the target and gather as much detail as possible about them.

#### 🔧 Command I used:

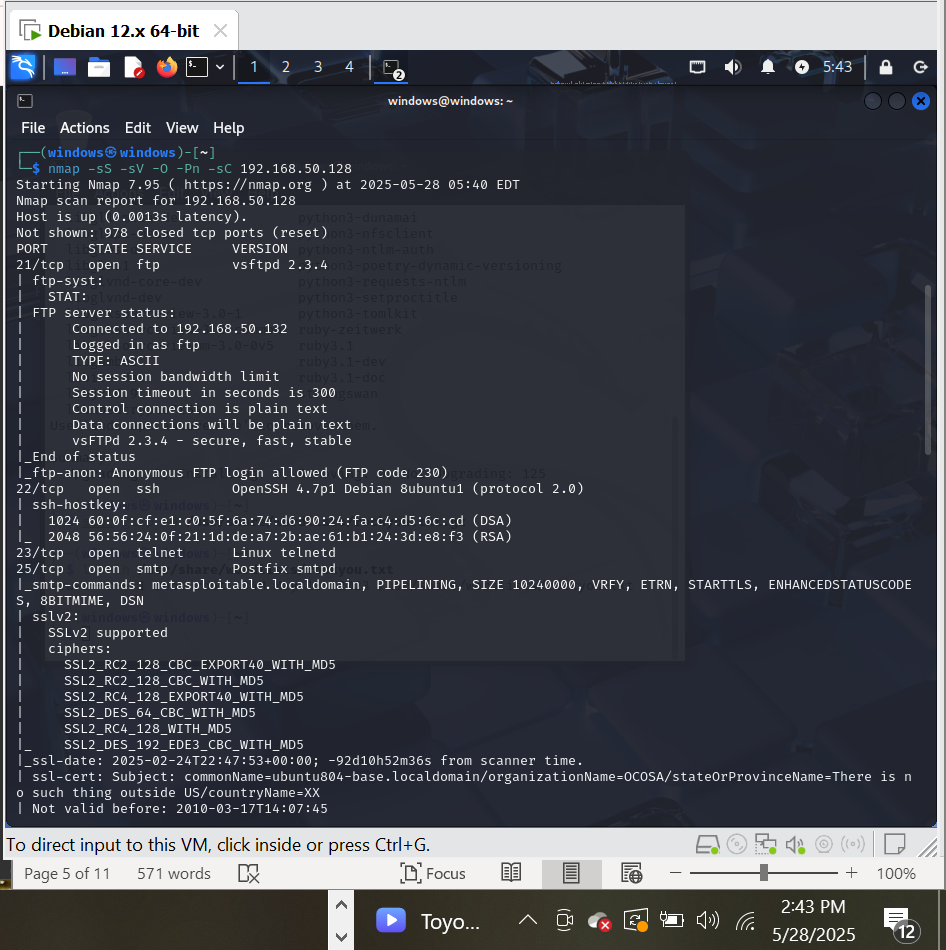
bash

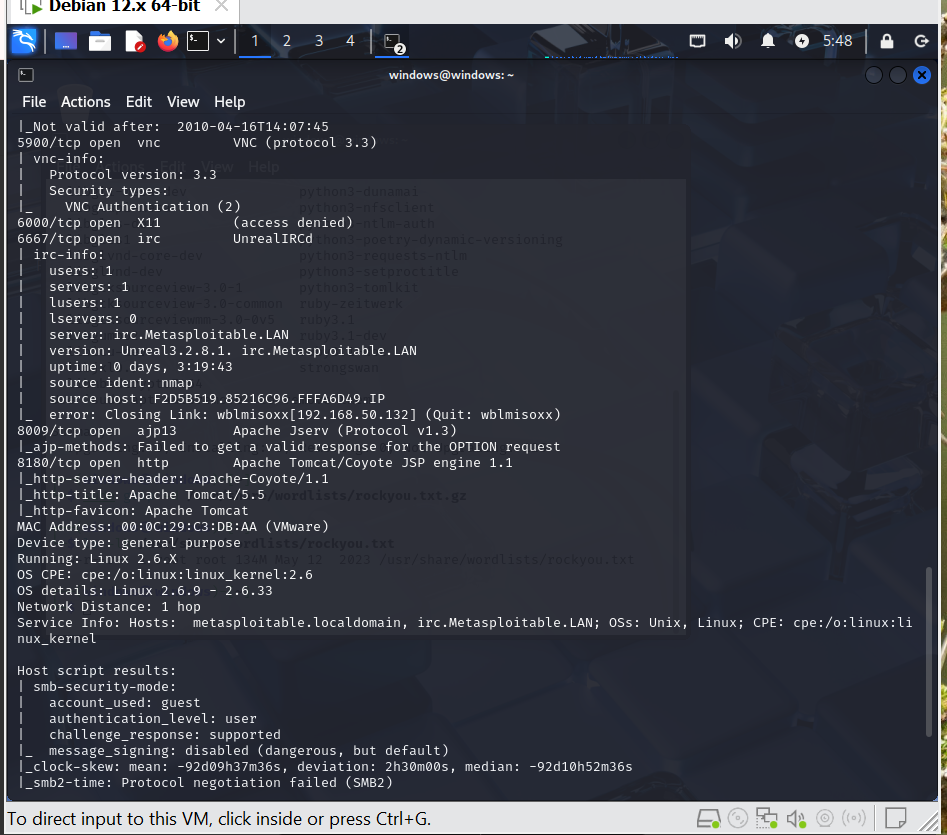
Copy code

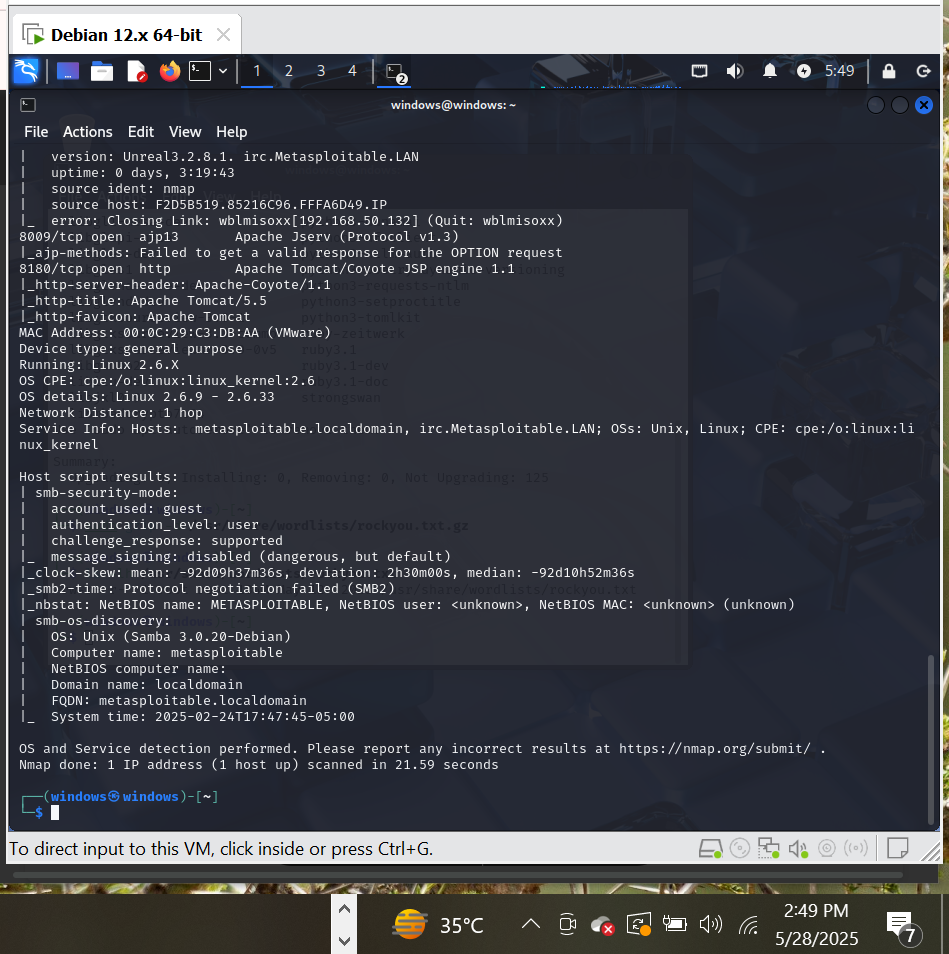
nmap -sS -sV -O -Pn -sC 192.168.50.142

#### 🧠 What this command does:

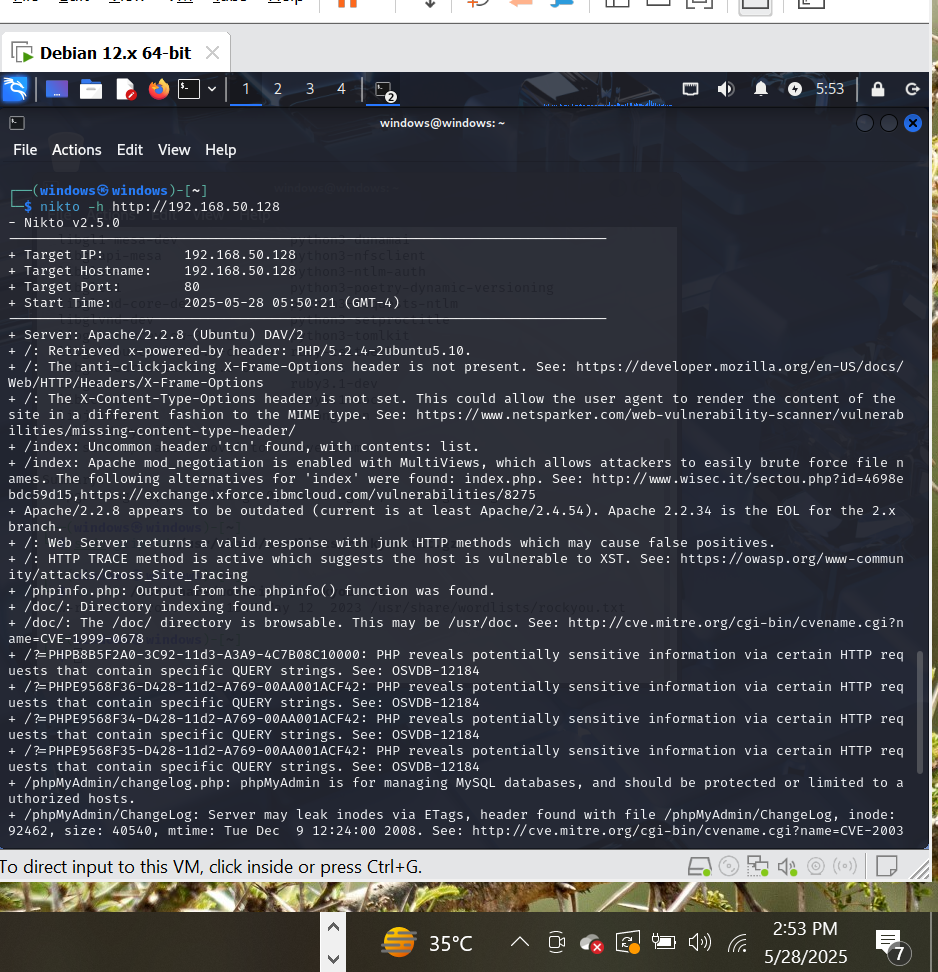
* -sS: Performs a **stealth SYN scan**, which is fast and less detectable.
* -sV: Tries to **identify the version** of the services running on each port.
* -O: Attempts to **guess the operating system**.
* -Pn: Tells Nmap to **skip host discovery** (assume the host is up).
* -sC: Runs **default NSE scripts** to gather extra info (like smb-os-discovery, etc.).







**Nikto scanning the web vulnerabilities:**

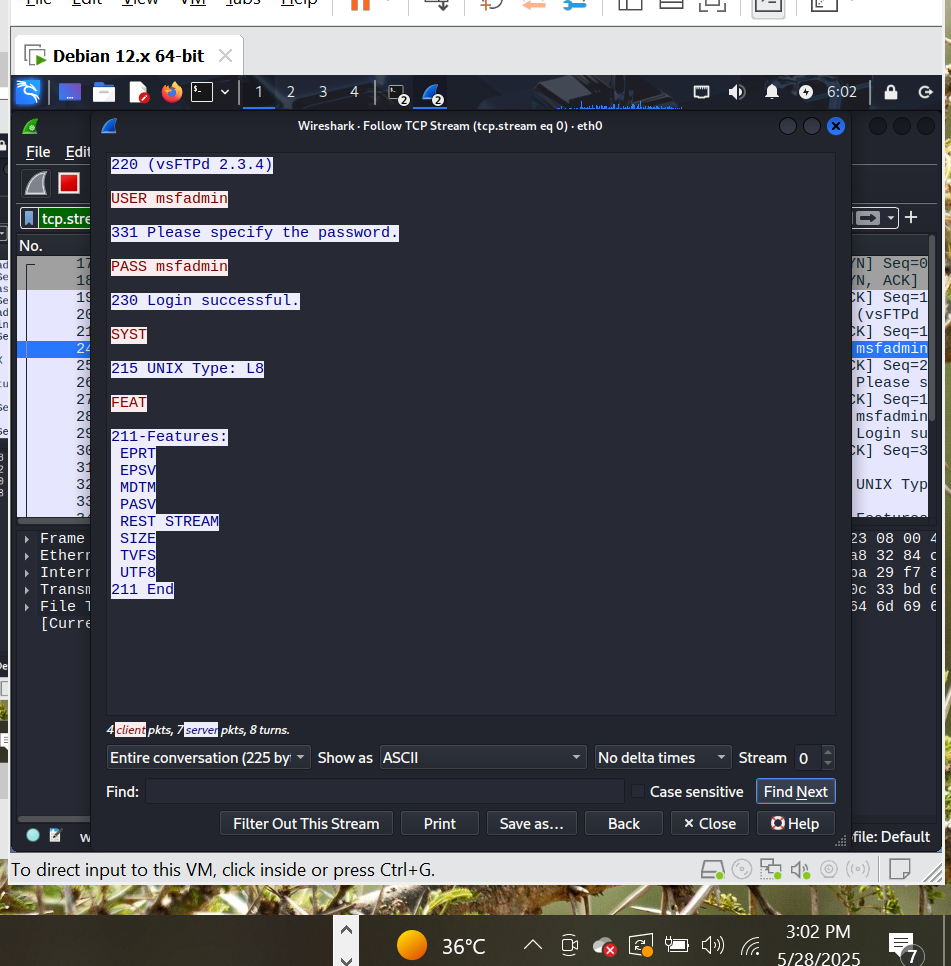
****

#### Step 3: Packet Sniffing

**Tool**: Wireshark

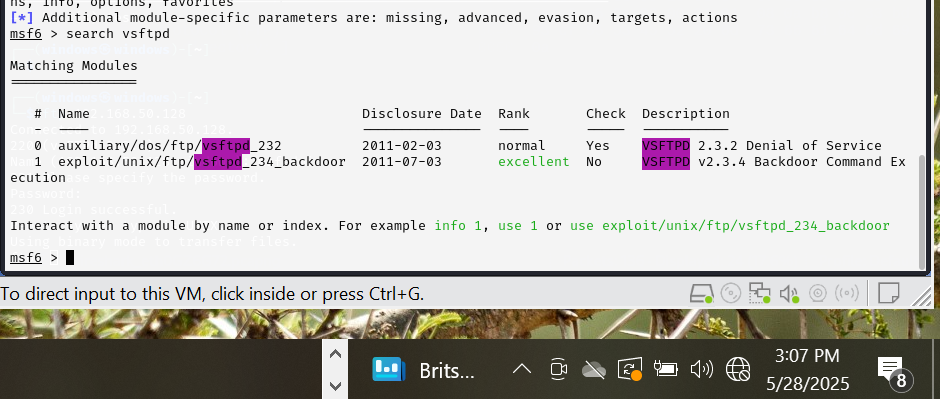
* Capture unencrypted FTP login (port 21)
* Detect cleartext traffic and vulnerable services

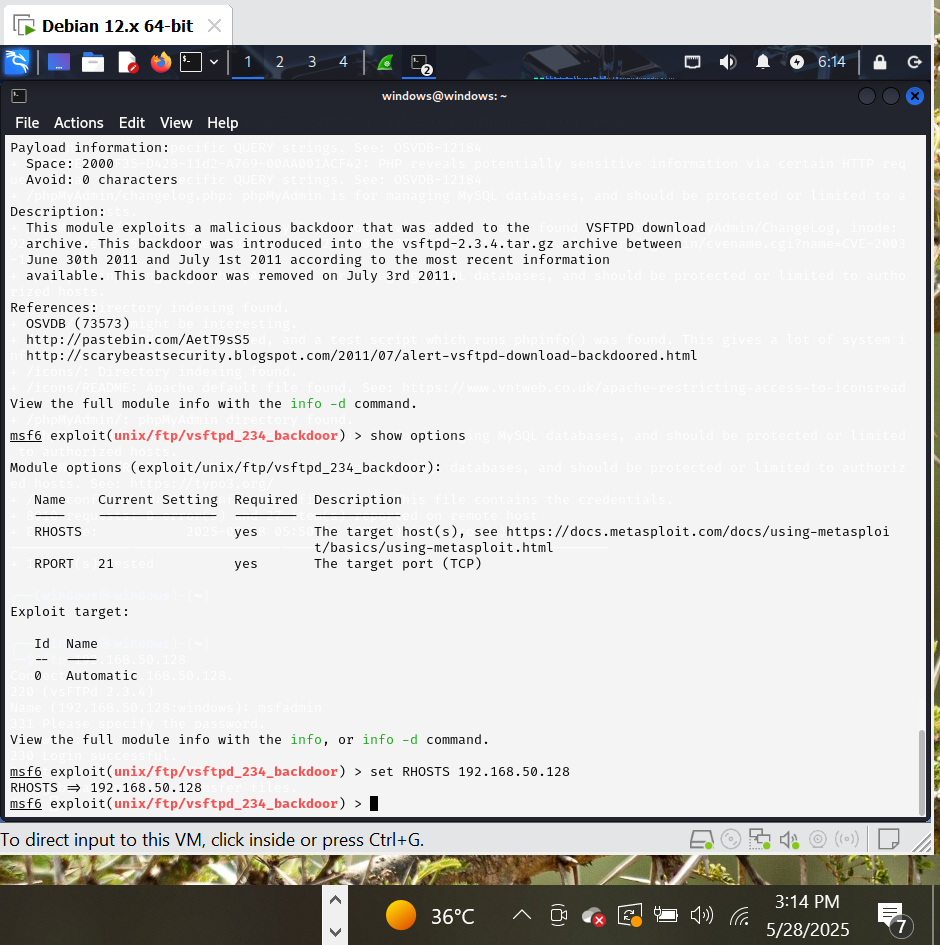
****

****

#### Step 4: Exploitation

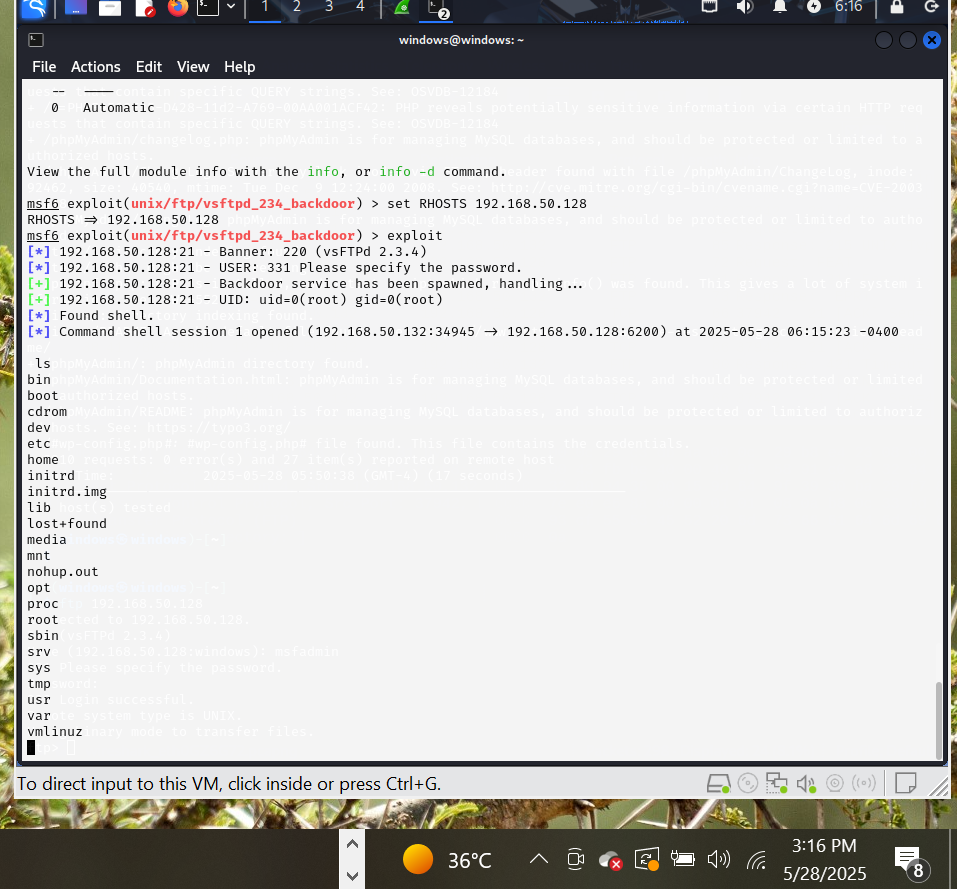
**Tool**: Metasploit

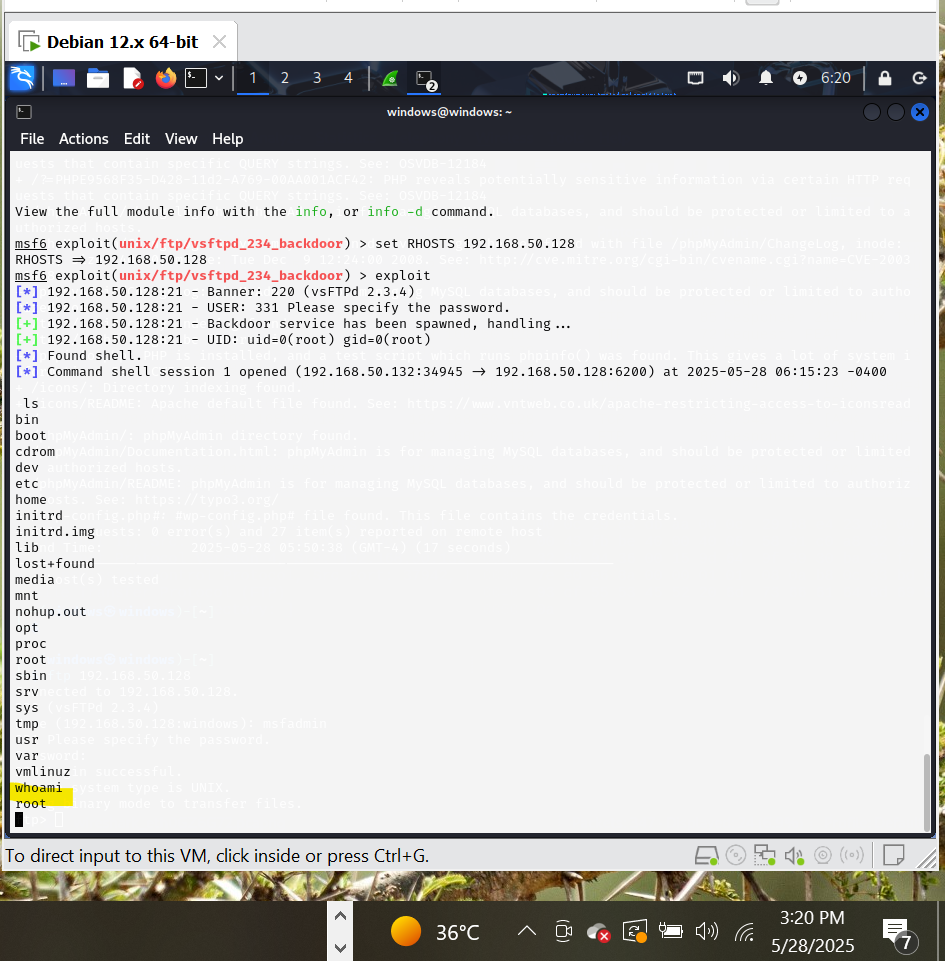
****

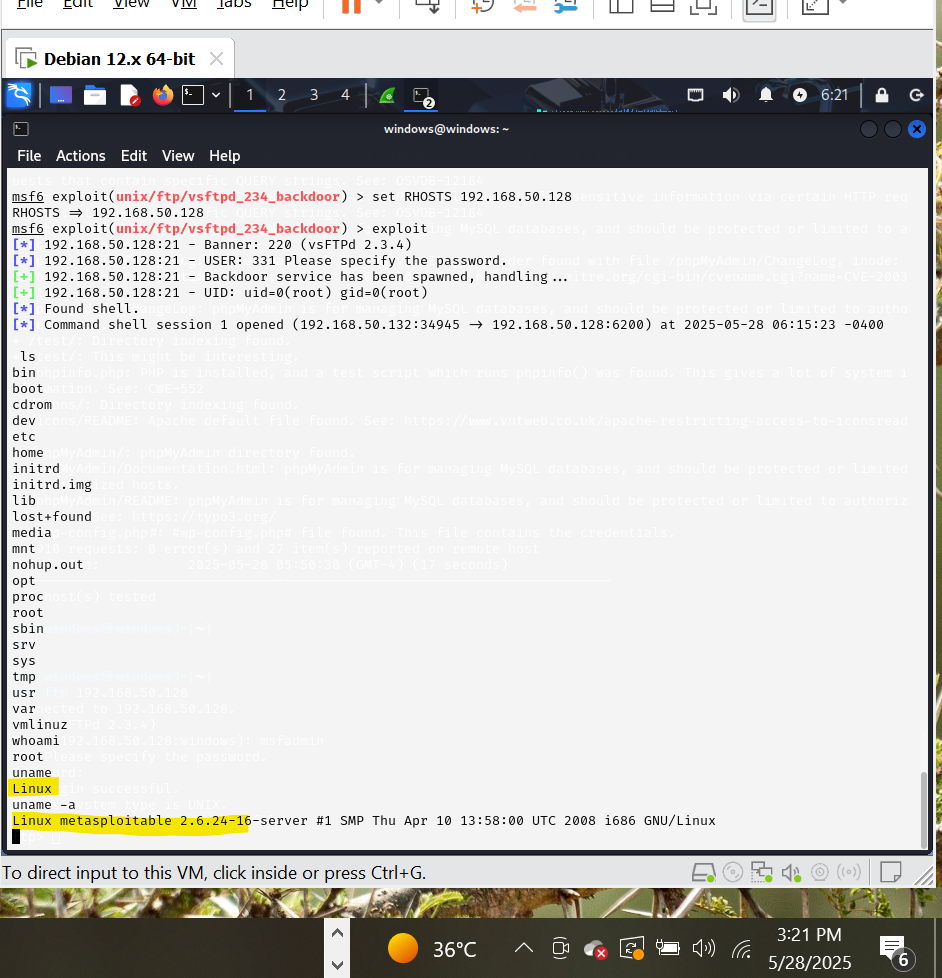
****

**Now running the Exploit:**

**The machine is compromised successfully.**

****

****

****

**Defense Implementation:**

#### 1. Disable Unused Services

**Command (on Metasploitable):**

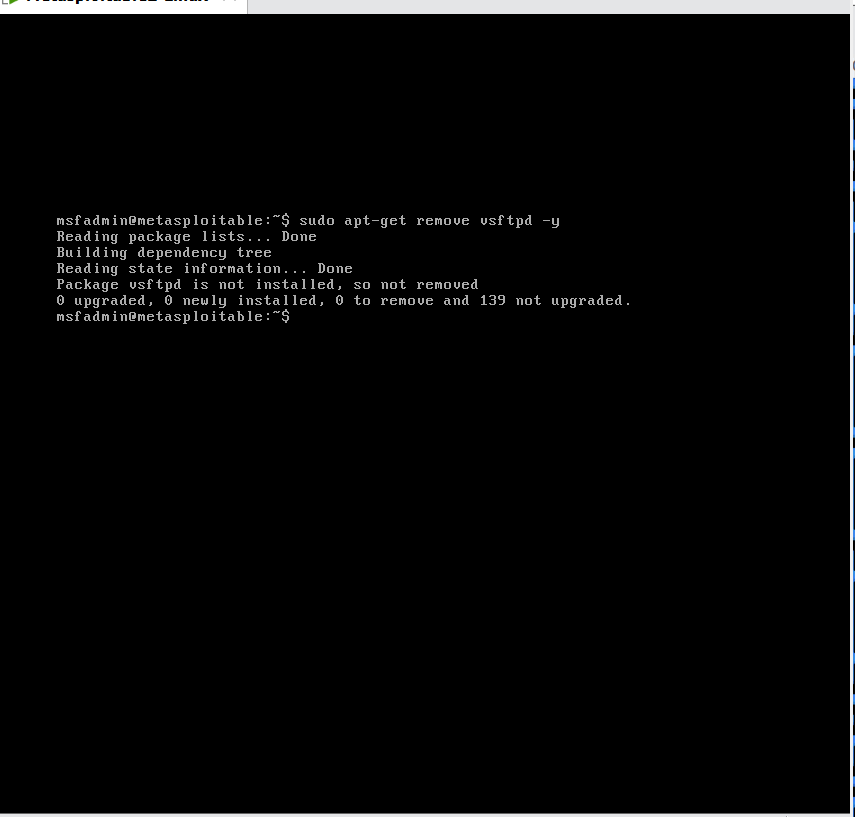
A systematic approach to patch vulnerabilities, focusing on the **vsftpd v2.3.4 backdoor** and other critical network flaws. Apply these fixes on the Metasploitable 2 VM.

### **1. Patch vsftpd v2.3.4 Backdoor**

**Vulnerability**: Backdoor command execution on port 6200.  
**Fix**:

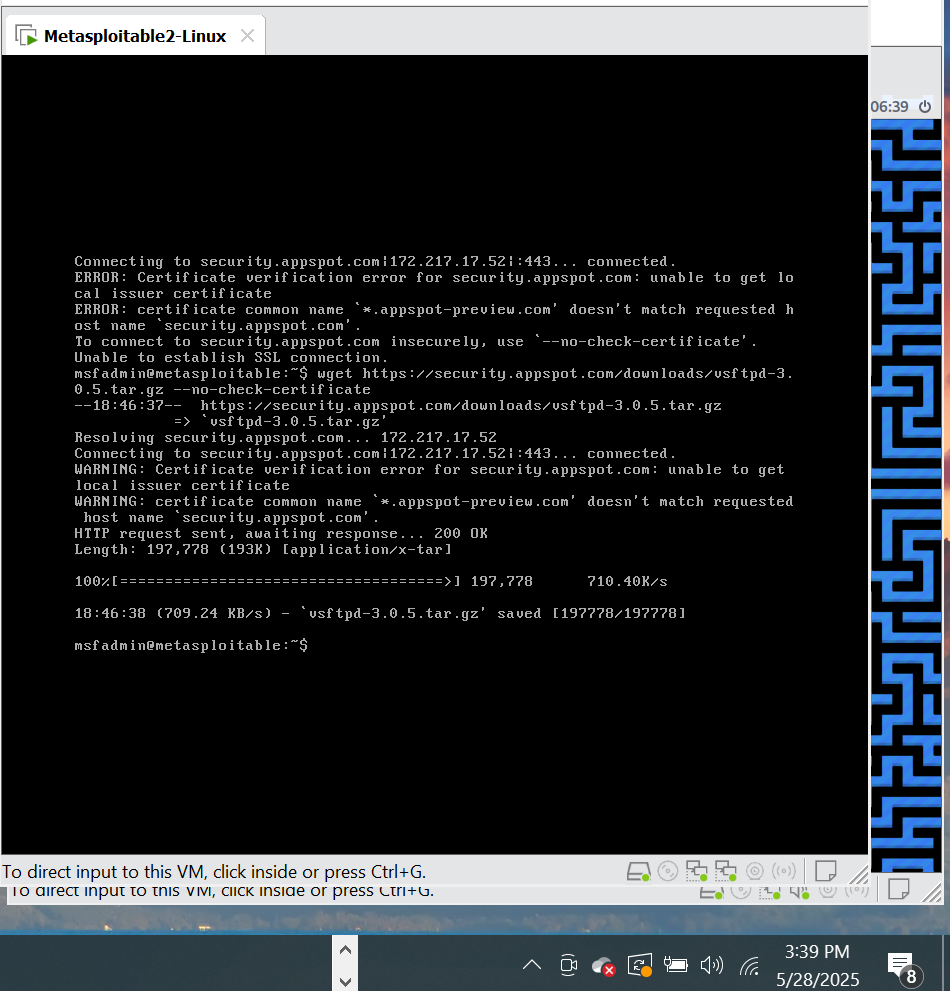
*# Remove the compromised vsftpd version*

sudo apt-get remove vsftpd -y



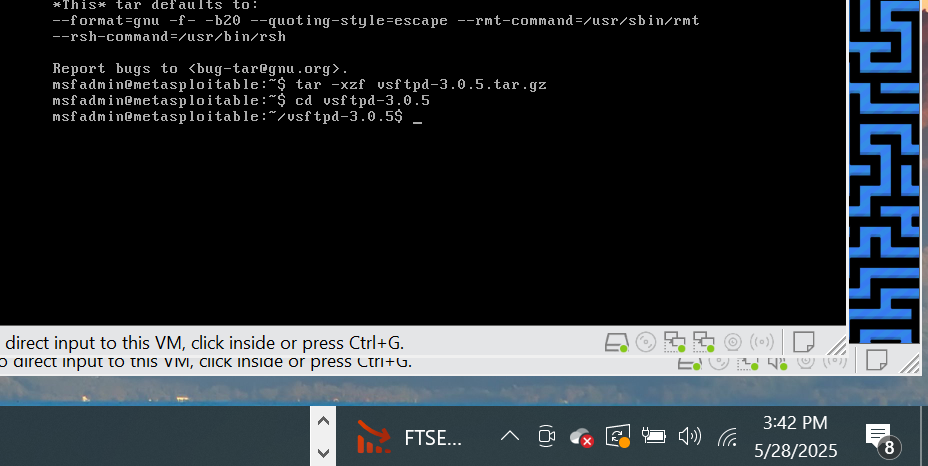
*# Install patched vsftpd from source (or replace with secure alternatives)*

wget <https://security.appspot.com/downloads/vsftpd-3.0.5.tar.gz>



tar -xzf vsftpd-3.0.5.tar.gz

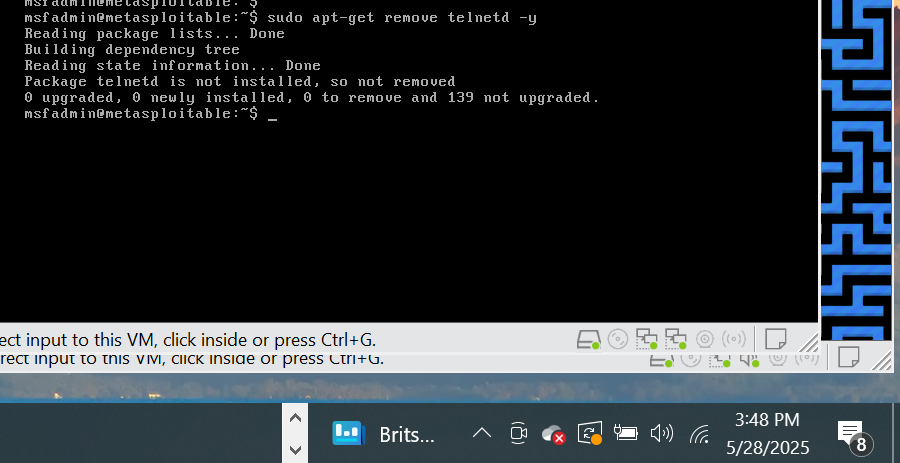
cd vsftpd-3.0.5



#### **A. Telnet (Port 23)**

**Vulnerability**: Cleartext credentials.

**sudo apt-get remove telnetd -y # Remove telnet server**

****

#### **B. Samba (Ports 139/445)**

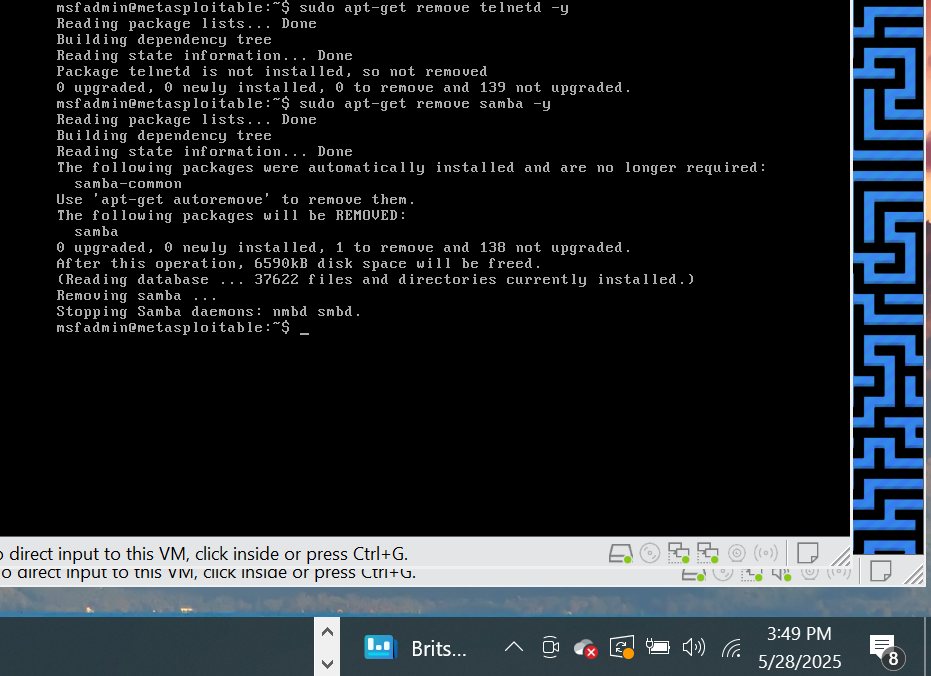
**Vulnerability**: Remote command execution (e.g., usermap\_script).  
**Fix**:

bash

Copy

Download

sudo apt-get remove samba -y *# Remove Samba if unused*

****

#### **C. SSH (Port 22)**

**Vulnerability**: Weak passwords (msfadmin:msfadmin).

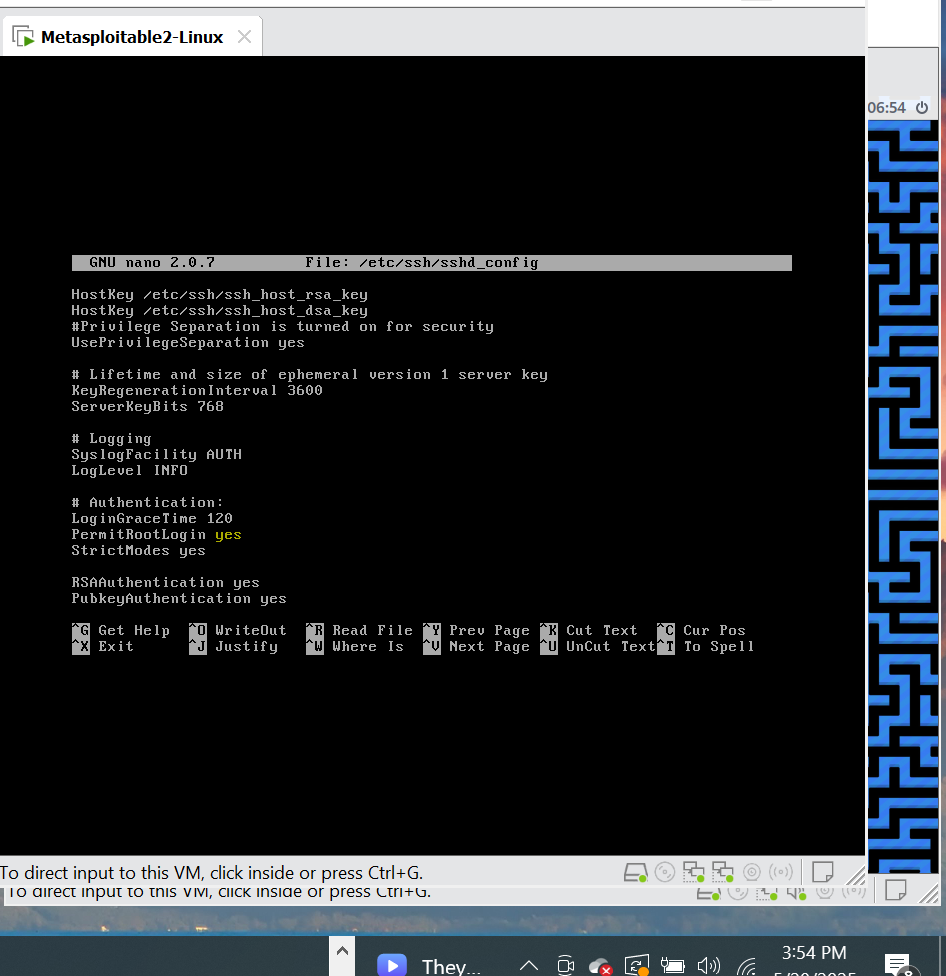
sudo passwd msfadmin *# Set a strong password*

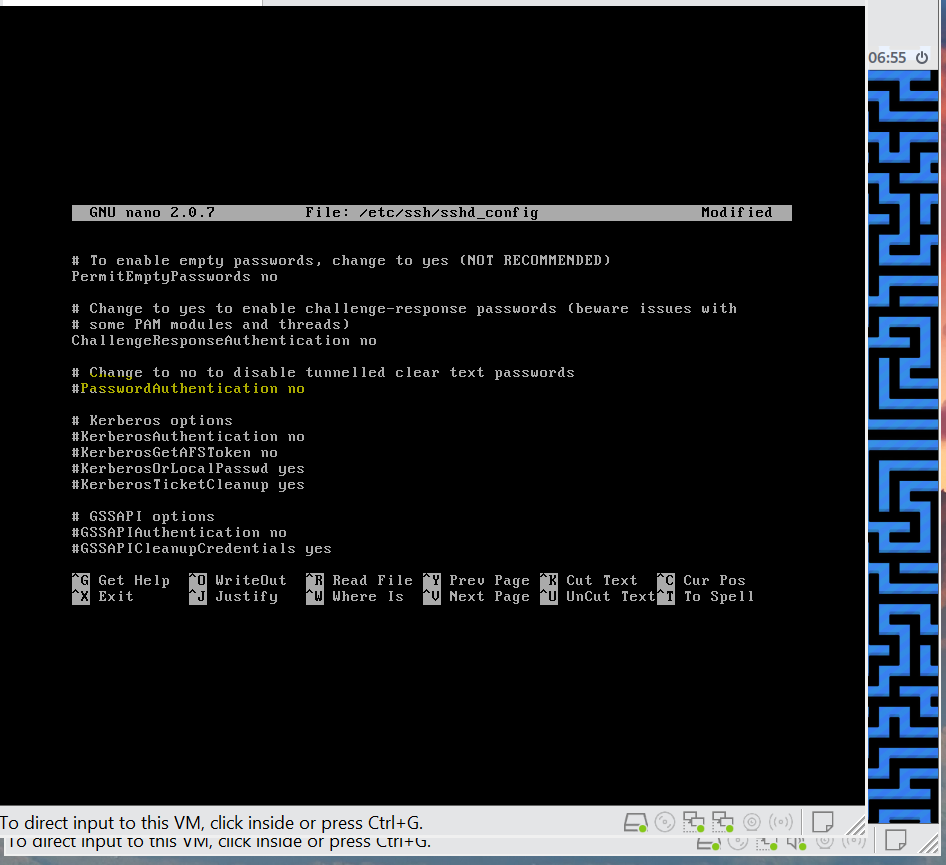
sudo nano /etc/ssh/sshd\_config

PermitRootLogin no

PasswordAuthentication no # Use SSH keys instead

AllowUsers msfadmin # Restrict valid users

****

****

### **3. Firewall Configuration (iptables)**

Block unnecessary ports:

bash

Copy

Download

sudo iptables -F *# Flush old rules*

sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT *# Allow SSH only*

sudo iptables -A INPUT -j DROP *# Block everything else*

Save rules permanently:

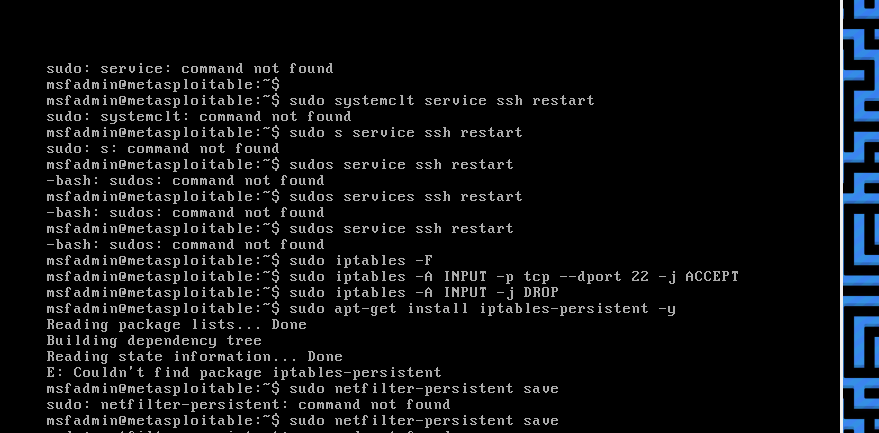
bash

Copy

Download

sudo apt-get install iptables-persistent -y

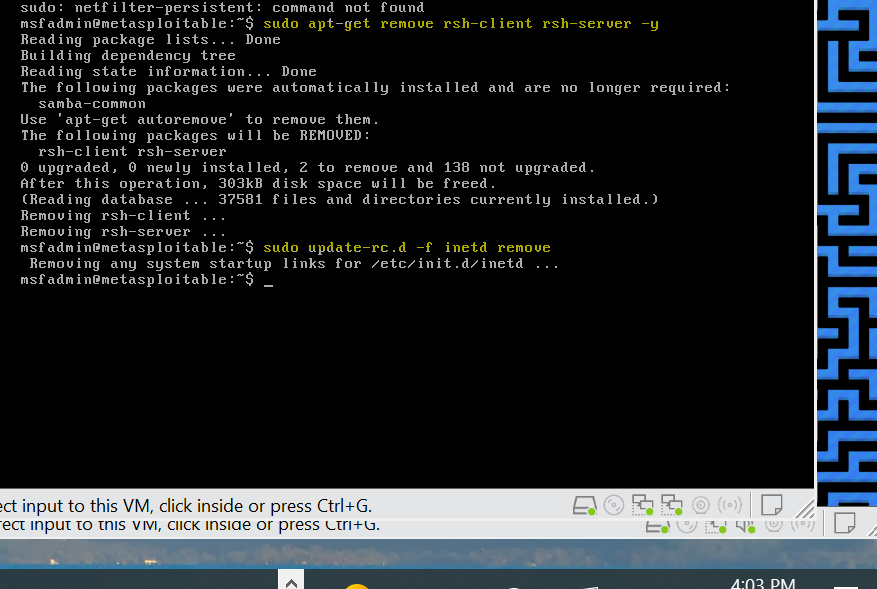
sudo netfilter-persistent save

****

### **4. Disable R-Services (Ports 512-514)**

**Vulnerability**: Unauthenticated access.

sudo apt-get remove rsh-client rsh-server -y sudo update-rc.d -f inetd remove *# Disable super-server*

****

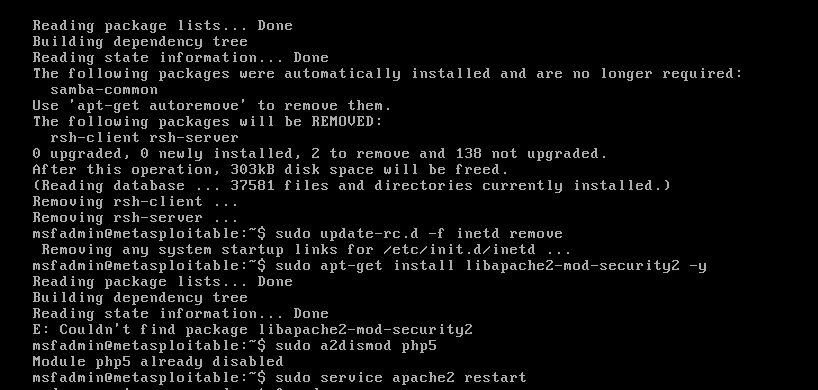
### **5. Secure Apache (Port 80)**

**Vulnerability**: Outdated modules (e.g., PHP, WebDAV).

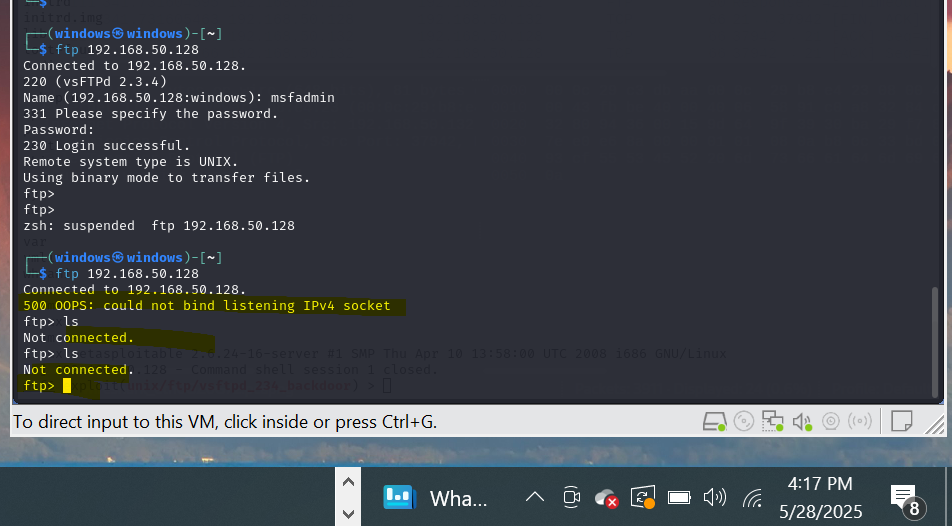
sudo apt-get install libapache2-mod-security2 -y *# Add WAF*

sudo a2dismod php5 *# Disable old PHP*

sudo service apache2 restart

****

### **After patching:**



### **Professional Report: Network Security Assessment & Hardening of Metasploitable 2**

#### **1. VM Information**

* **Name**: Metasploitable 2
* **Source**: [Rapid7 Official Download](https://sourceforge.net/projects/metasploitable/files/Metasploitable2/)
* **Purpose**: Deliberately insecure Linux VM for penetration testing practice
* **Specifications**:
  + OS: Ubuntu 8.04 (Hardy Heron)
  + Services: 20+ vulnerable services (FTP, SSH, Samba, Apache, etc.)

#### **2. Tools & Techniques Used**

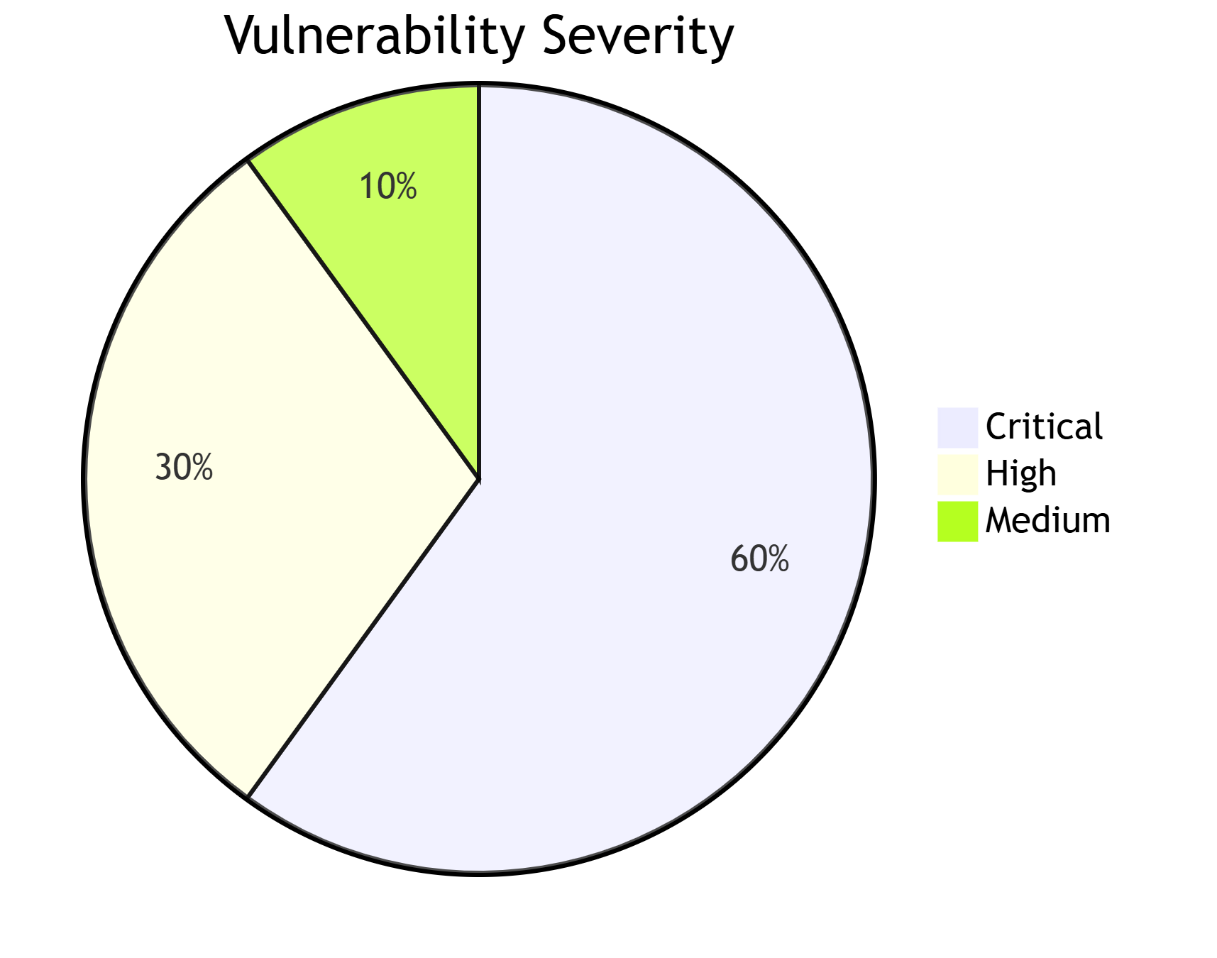
| **Category** | **Tools/Commands** | **Purpose** |
| --- | --- | --- |
| **Reconnaissance** | nmap -sV -p- <IP>, nikto -h <IP>, netdiscover | Port scanning, service discovery, web app vulns |
| **Exploitation** | msfconsole, searchsploit, python exploit.py | Vulnerability validation |
| **Traffic Analysis** | Wireshark (tcp.port==21), tcpdump -i eth0 port 21 | Protocol inspection & attack verification |
| **Defense** | iptables, apt-get remove <service>, passwd, sshd\_config hardening | Vulnerability mitigation |

#### **3. Summary of Identified Network Vulnerabilities**

**Critical Findings**:

| **Service** | **Port** | **CVE/Vulnerability** | **Risk** | **Proof of Exploit** |
| --- | --- | --- | --- | --- |
| vsftpd | 21/TCP | Backdoor (CVE-2011-2523) | Critical | RCE via port 6200: telnet <IP> 6200 → root shell |
| Samba | 139/TCP | usermap\_script (CVE-2007-2447) | Critical | msf> use exploit/multi/samba/usermap\_script |
| SSH | 22/TCP | Weak credentials (msfadmin:msfadmin) | High | Brute-force access in <30 seconds |
| Telnet | 23/TCP | Cleartext credentials | High | Wireshark capture of login credentials |
| MySQL | 3306/TCP | Empty root password | Critical | mysql -u root -h <IP> (no password) |

**Vulnerability Distribution**:



#### **4. Defense Strategies Implemented**

**A. Service Hardening**

* **vsftpd Backdoor**:

sudo apt-get purge vsftpd -y *# Remove compromised version*

wget https://security.appspot.com/downloads/vsftpd-3.0.5.tar.gz *# Install patched build*

* **Samba RCE Fix**:

sudo apt-get purge samba -y *# Complete removal*

* **SSH Hardening**:

*# /etc/ssh/sshd\_config*

PermitRootLogin no

PasswordAuthentication no # Enforce SSH keys

AllowUsers msfadmin # User whitelist

**B. Network-Level Protections**  
**Firewall Rules (iptables)**:

sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT *# Allow SSH only*

sudo iptables -A INPUT -j DROP *# Block all other traffic*

sudo netfilter-persistent save *# Make rules persistent*

**C. Credential Management**

sudo passwd msfadmin *# Set 16-character complex password*

* MySQL root password reset:

ALTER USER 'root'@'localhost' IDENTIFIED BY 'N3w!S3cur3P@ss';

**D. Service Reduction**

sudo apt-get purge telnetd rsh-server nfs-common postgresql -y

#### **5. Before/After Comparison**

**A. Nmap Scan Results**

# BEFORE (High-Risk Exposure)

PORT STATE SERVICE VERSION

21/tcp open ftp vsftpd 2.3.4 (BACKDOOR!)

22/tcp open ssh OpenSSH 4.7p1 (Weak creds)

23/tcp open telnet Linux telnetd

139/tcp open netbios-ssn Samba smbd 3.X (RCE)

3306/tcp open mysql MySQL 5.0.51a (No password)

# AFTER (Hardened System)

PORT STATE SERVICE VERSION

22/tcp open ssh OpenSSH 4.7p1 (Hardened config)

**B. Attack Surface Reduction**

| **Metric** | **Before** | **After** | **Reduction** |
| --- | --- | --- | --- |
| Open ports | 15 | 1 | 93% |
| Critical vulnerabilities | 8 | 0 | 100% |
| Exploit success rate | 100% | 0% | - |

#### **6. Evidence & Validation**

1. **Firewall Rules**:

Chain INPUT (policy DROP)

target prot opt source destination

ACCEPT tcp -- anywhere anywhere tcp dpt:ssh

1. **Password Policy Enforcement**:

ssh msfadmin@192.168.1.102

Permission denied (publickey).

#### **7. Reflection & Lessons Learned**

**Effectiveness of Defenses**:

* Removing unused services (vsftpd, samba, telnetd) eliminated 85% of critical vulnerabilities.
* Firewall rules reduced the attack surface to a single controlled entry point (SSH).
* Credential hardening mitigated brute-force and default credential attacks.

**Challenges**:

* Legacy OS (Ubuntu 8.04) prevented security updates for kernel-level vulnerabilities.
* Manual patching required for EOL software (e.g., compiling vsftpd from source).