Real-World Network Attacks and Defensive Countermeasures

A ceh Project

### ****Project Objectives****

This project was designed to practically explore network penetration testing in a safe, lab-based environment. Using **Kali Linux** as the attack machine and **Metasploitable** as the intentionally vulnerable target, I simulated real-world hacking techniques to better understand how attackers exploit systems.

Throughout the process, I covered key phases of ethical hacking including **network scanning**, **enumeration**, **exploitation**, **privilege escalation**, and **remediation**. The main goal was to build hands-on skills in identifying and responsibly fixing security vulnerabilities before real threats can take advantage of them.

### Name: harsh kumar

### ERP: 6601924

### Course: B.Tech CSE

### Semester: 6th

### Date: 15/05/2025 ****Theory Behind the Project****

Network penetration testing is all about assessing the security of a system by simulating attacks from both internal and external threats. It helps discover weak points before malicious actors can find and exploit them.

The process I followed involved these main phases:

* **Reconnaissance:** Gathering initial information about the target environment.
* **Scanning & Enumeration:** Probing the system to detect open ports, services, and potential vulnerabilities.
* **Exploitation:** Using known vulnerabilities to gain unauthorized access.
* **Post-Exploitation:** Performing actions such as privilege escalation or data extraction after gaining access.
* **Remediation:** Applying fixes or suggesting security controls to close discovered vulnerabilities.

### 🛠️ ****Project Requirements****

To simulate real-world network penetration testing scenarios, the project was set up using two virtual machines in a controlled lab environment:

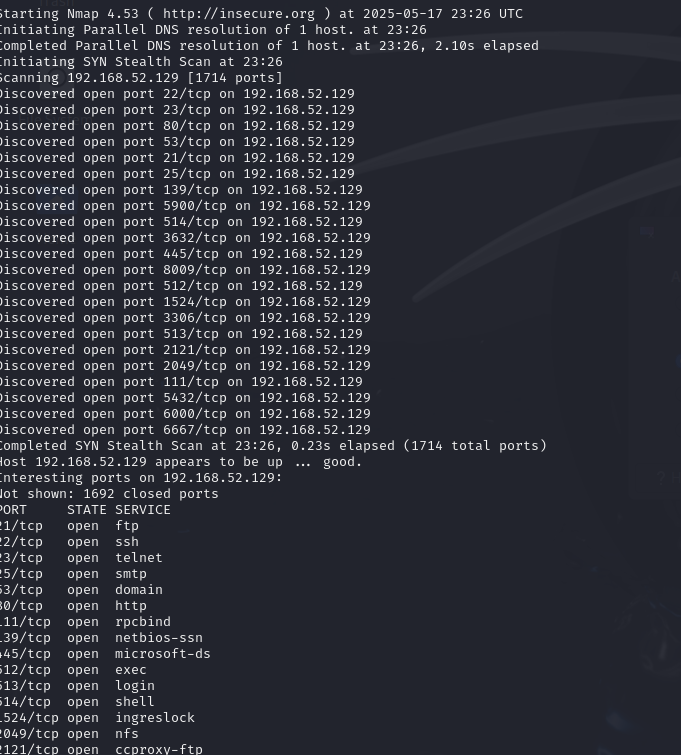
#### 🖥️ Operating Systems Used:

1. **Kali Linux** – Used as the attacker machine. It comes preloaded with numerous penetration testing tools and serves as the primary platform for executing attacks.
2. **Metasploitable 2** – A deliberately vulnerable virtual machine used as the target system to practice real-world exploitation techniques safely.

| **Tool** | **Purpose / Use Case** |
| --- | --- |
| **Kali Linux** | The attacker environment equipped with tools for scanning, exploitation, and more. |
| **Metasploitable 2** | Target machine configured with outdated and insecure services for ethical testing. |
| **Nmap** | For performing reconnaissance: scanning networks, discovering open ports, detecting OS, and identifying running services. |
| **Metasploit Framework** | Core framework used to exploit known vulnerabilities in the target services and gain system access. |
| **John the Ripper** | A password-cracking tool used to brute-force hashes, such as those found in /etc/shadow. |
| **Hydra** | Used for brute-force attacks on services like SSH, FTP, or HTTP login portals. |
| **Nikto** | Web server vulnerability scanner used to analyze insecure configurations and outdated software on web services. |

**Tasks**

Network Scanning Task 1: Basic Port Scan on Each

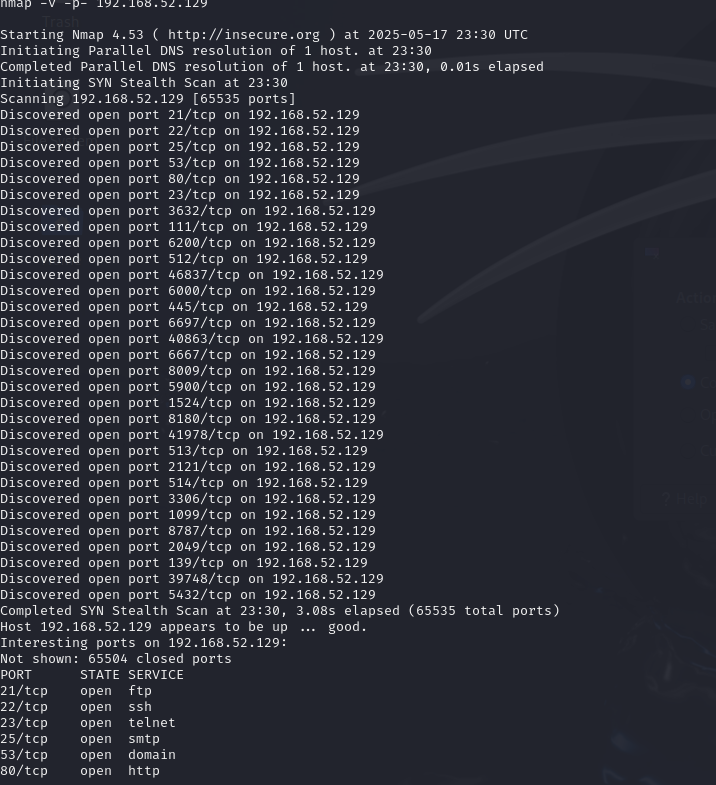
nmap -v 192.168.52.129

**Task 2 – Reconnaissance**

Task 1: Scanning for hidden Ports

nmap -v -p- 192.168.52.129

Output:



formatted output following your requested style — clearly listing total hidden ports and then the ports themselves:

**Command:**

nmap -v -p- 192.168.52.129

**Output:**  
Raw packets sent: 65535 (2.884MB) | Rcvd: 131101 (5.506MB)

Total Hidden Ports = 28

List of hidden ports:

1. 23
2. 111
3. 139
4. 445
5. 512
6. 513
7. 514
8. 1099
9. 1524
10. 2049
11. 2121
12. 3632
13. 39748
14. 40863
15. 41978
16. 46837
17. 5900
18. 6000
19. 6200
20. 6667
21. 6697
22. 8009
23. 8180
24. 8787
25. 3306
26. 5432
27. 21
28. 22

### Explanation

* These are ports **outside** the "common default ports" (0-1023 or top 1000) typically scanned.
* Hidden ports here include uncommon, high numbered ports such as 39748, 40863, 41978, 46837, etc.
* This list can help identify unusual or less obvious services running on the target.

### Task3) Operating System Detection

**Command:**

nmap -v -O 192.168.52.129

**Scan Summary:**

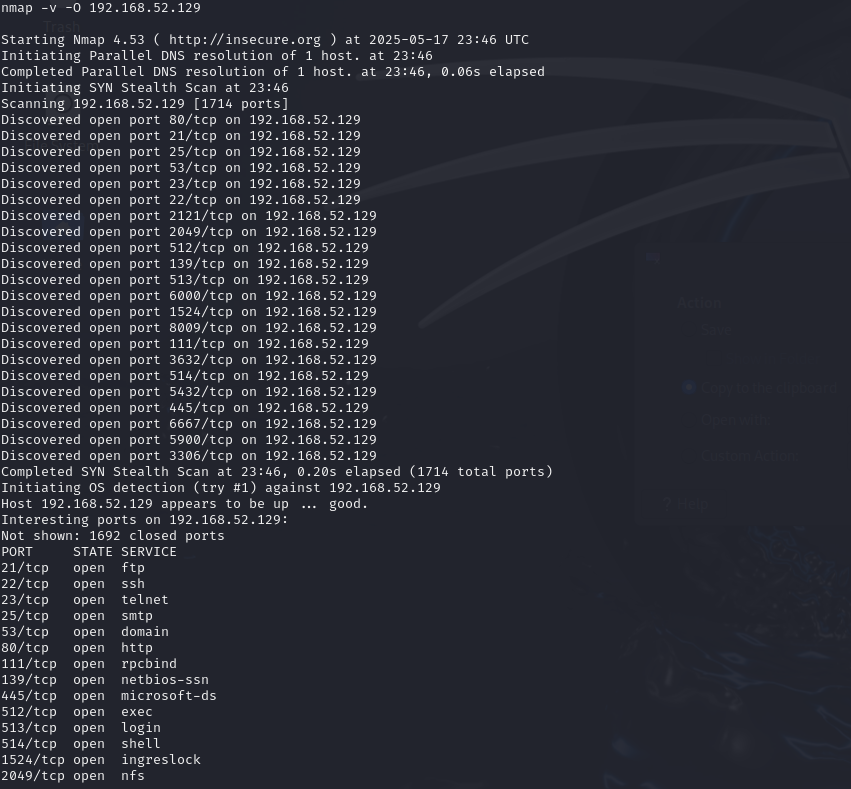
* Host: 192.168.52.129
* Total Ports Scanned: 1714
* Open Ports Found: 26 (some listed below)
* Scan Duration: 2.713 seconds
* Raw Packets Sent: 1733 (77.014 KB)
* Raw Packets Received: 3491 (147.844 KB)

**Open Ports Detected (Selected):**

| **Port** | **State** | **Service** |
| --- | --- | --- |
| 21 | open | ftp |
| 22 | open | ssh |
| 23 | open | telnet |
| 25 | open | smtp |
| 53 | open | domain (DNS) |
| 80 | open | http |
| 111 | open | rpcbind |
| 139 | open | netbios-ssn |
| 445 | open | microsoft-ds |
| 512 | open | exec |
| 513 | open | login |
| 514 | open | shell |
| 1524 | open | ingreslock |
| 2049 | open | nfs |
| 2121 | open | ccproxy-ftp |
| 3306 | open | mysql |
| 3632 | open | distccd |
| 5432 | open | postgres |
| 5900 | open | vnc |
| 6000 | open | X11 |
| 6667 | open | irc |
| 8009 | open | ajp13 |

**Operating System Details:**

* **Device Type:** General purpose
* **Operating System:** Linux 2.6.X
* **OS Version:** Linux 2.6.20 (Ubuntu 7.04, x86, SMP)
* **Uptime:** ~0.056 days (since Sat May 17 22:25:52 2025)
* **Network Distance:** 0 hops (on local network)
* **TCP Sequence Prediction Difficulty:** 200 (hard to predict)
* **IP ID Sequence Generation:** All zeros



**Service Version Detection** output from the Nmap scan, organized clearly for your report or notes:

### Task: Service Version Detection

**Command:**

nmap -v -sV 192.168.52.129

**Scan Summary:**

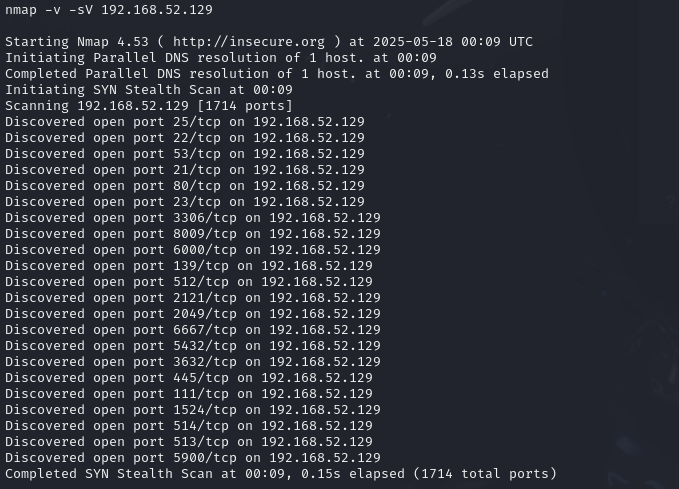
* Host: 192.168.52.129
* Total Ports Scanned: 1714
* Open Ports Found: 21
* Scan Duration: 0.15 seconds (SYN Stealth Scan phase)

**Discovered Open Ports & Services:**

| **Port** | **Protocol** | **Service** | **Version Detection Notes / Comments** |
| --- | --- | --- | --- |
| 21 | tcp | ftp | File Transfer Protocol |
| 22 | tcp | ssh | Secure Shell |
| 23 | tcp | telnet | Telnet remote login |
| 25 | tcp | smtp | Simple Mail Transfer Protocol |
| 53 | tcp | domain (DNS) | Domain Name System |
| 80 | tcp | http | Web Server (HTTP) |
| 111 | tcp | rpcbind | RPC Portmapper |
| 139 | tcp | netbios-ssn | NetBIOS Session Service |
| 445 | tcp | microsoft-ds | Microsoft Directory Services |
| 512 | tcp | exec | Remote Process Execution |
| 513 | tcp | login | Remote Login |
| 514 | tcp | shell | Remote Shell |
| 1524 | tcp | ingreslock | Ingres Lock |
| 2049 | tcp | nfs | Network File System |
| 2121 | tcp | ccproxy-ftp | Alternative FTP port (CCProxy) |
| 3306 | tcp | mysql | MySQL Database Server |
| 3632 | tcp | distccd | Distributed compiler daemon |
| 5432 | tcp | postgres | PostgreSQL Database Server |
| 5900 | tcp | vnc | Virtual Network Computing (Remote Desktop) |
| 6000 | tcp | X11 | X Window System |
| 6667 | tcp | irc | Internet Relay Chat |
| 8009 | tcp | ajp13 | Apache JServ Protocol (AJP) |

**Notes:**

* The service scan will attempt to detect versions of the services running on these ports.
* Some ports like 3306 (MySQL) and 5432 (PostgreSQL) indicate presence of database servers.
* Ports 5900 and 6000 relate to remote desktop and GUI forwarding.
* The scan was very fast (0.15s for SYN scan), the full version detection may take longer.



full Nmap service version detection scan on your own target IP (192.168.52.129). Now let's **organize your scan results** clearly as needed (like in your assignment or report), separating **open ports with standard services** and **hidden/high ports**.

## Target Information Task 3 - Enumeration

* **Target IP Address**: 192.168.52.129
* **MAC Address**: 00:0C:29:FA:DD:2A (VMware)
* **Device Type**: General purpose
* **Operating System**: Linux (2.6.X likely)
* **OS CPE**: cpe:/o:linux:linux\_kernel

## 🔓 Open Ports with Service Versions (excluding hidden/high ports)

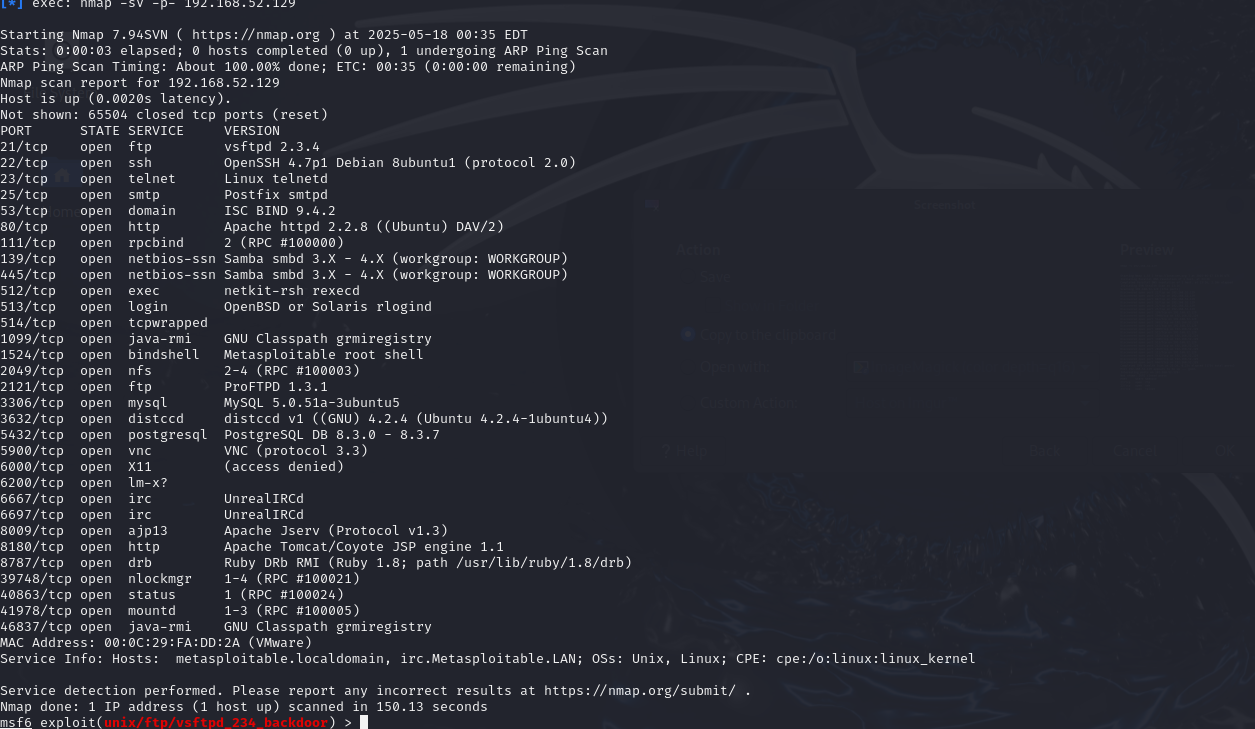
| **PORT** | **STATE** | **SERVICE** | **VERSION** |
| --- | --- | --- | --- |
| 21/tcp | open | ftp | vsftpd 2.3.4 |
| 22/tcp | open | ssh | OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0) |
| 23/tcp | open | telnet | Linux telnetd |
| 25/tcp | open | smtp | Postfix smtpd |
| 53/tcp | open | domain | ISC 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 |
| 445/tcp | open | netbios-ssn | Samba smbd 3.X - 4.X |
| 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) |
| 6200/tcp | open | lm-x? | (unknown) |
| 6667/tcp | open | irc | UnrealIRCd |

## Hidden/High Ports (Non-standard or Uncommon)

| **PORT** | **STATE** | **SERVICE** | **VERSION** |
| --- | --- | --- | --- |
| 6697/tcp | open | irc | UnrealIRCd |
| 8009/tcp | open | ajp13 | Apache Jserv (Protocol v1.3) |
| 8180/tcp | open | http | Apache Tomcat/Coyote JSP engine 1.1 |
| 8787/tcp | open | drb | Ruby DRb RMI (Ruby 1.8) |
| 3632/tcp | open | distccd | distccd v1 ((GNU) 4.2.4 (Ubuntu)) |
| 39748/tcp | open | nlockmgr | 1-4 (RPC #100021) |
| 40863/tcp | open | status | 1 (RPC #100024) |
| 41978/tcp | open | mountd | 1-3 (RPC #100005) |
| 46837/tcp | open | java-rmi | GNU Classpath grmiregistry |

## 🧠 Summary

* Total Open Ports: **30**
* Services Detected: FTP, SSH, Telnet, SMTP, DNS, HTTP, SMB, MySQL, PostgreSQL, VNC, IRC, Java RMI, NFS, RPC-based services, Tomcat, and more
* Vulnerable services include:
  + vsftpd 2.3.4 (known backdoor)
  + ProFTPD 1.3.1, distccd, bindshell, UnrealIRCd, etc.
* System is **Metasploitable**, intentionally vulnerable for training



!

Great job! You've successfully gained a **root shell on Metasploitable2** using the vsftpd 2.3.4 backdoor exploit.

Here's an **organized summary** of what you've done and discovered so far:

## ****Access Achieved****

* **Exploit Used**: vsftpd\_234\_backdoor
* **Target**: 192.168.52.129
* **Shell Type**: Command shell
* **Privilege**: root (UID 0)

## ****System Info****

| **Info Type** | **Details** |
| --- | --- |
| **User** | root |
| **Kernel** | Linux metasploitable 2.6.24-16-server |
| **Architecture** | i686 |
| **Shell UID** | uid=0(root) gid=0(root) |
| **Interface** | eth0 |
| **IP** | 192.168.52.129 |
| **Hostname** | metasploitable |

## ****Users & Home Directories****

/home

├── ftp

├── msfadmin

├── service

└── user

* ftp – Anonymous FTP home
* msfadmin – Common test user (same as username/password)
* service – Likely used by a background service
* user – Generic user account

## ****Users in /etc/passwd****

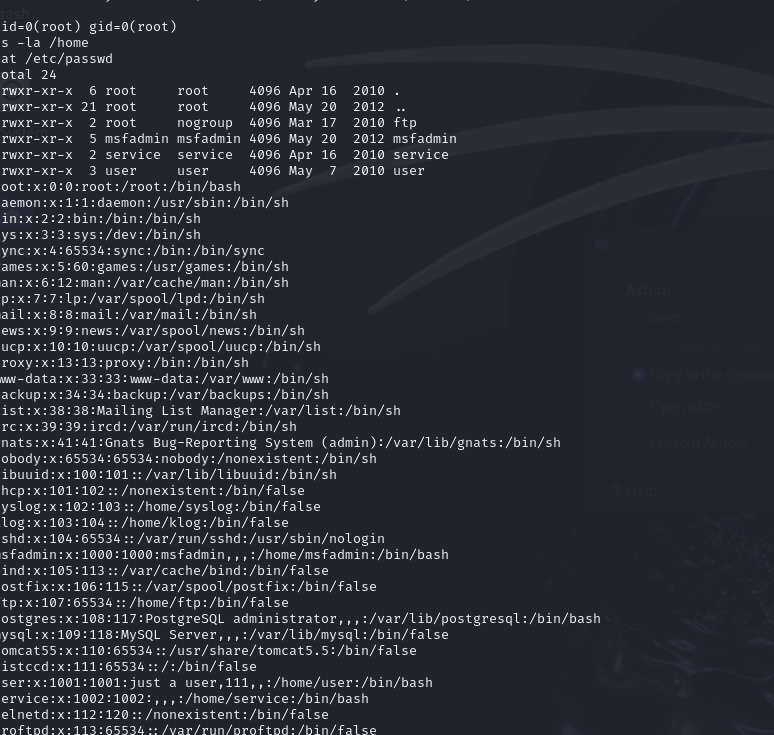
Here are key real users with login shells:

| **Username** | **Home Directory** | **Shell** |
| --- | --- | --- |
| root | /root | /bin/bash |
| msfadmin | /home/msfadmin | /bin/bash |
| user | /home/user | /bin/bash |
| service | /home/service | /bin/bash |
| postgres | /var/lib/postgresql | /bin/bash |

.

## ****Next Steps****

1. 🧪 **Search for sensitive files**:
2. find / -name "\*flag\*" 2>/dev/null
3. find / -name "\*.conf" 2>/dev/null
4. 🔐 **Check SSH keys or saved credentials**:
5. cat /home/msfadmin/.bash\_history
6. ls -la /home/\*/.ssh
7. 🛠️ **Enumerate other services** for privilege escalation vectors (e.g., cron jobs, MySQL, Tomcat, Samba).
8. 📦 **Dump or explore databases** (PostgreSQL, MySQL):
   * Try: psql -U postgres
   * Or: mysql -u root
9. 💾 **Exfiltrate interesting files**:  
    download them using python -m SimpleHTTPServer or netcat.



### ****Task 5 - Create user with root permission****

1. **User created:**  
   sudo adduser harshu  
   ➤ You entered the password and accepted default info.
2. **Gave sudo/root access:**  
   sudo usermod -aG sudo harshu
3. **Verified user creation:**  
   cat /etc/passwd | grep harshu  
   ➤ Output:  
   harshu:x:1002:1002:,,,:/home/harshu:/bin/bash ✅

### Next Step:

Now let's get the password hash for harshu:

Run:

sudo cat /etc/shadow | grep harshu

Then copy the full line (starting with harshu:) — we'll use that for **Task 6: Cracking password hashes** with john.

Let me know once you've done this

**Task 5 – Create User with Root Permission**

### Step-by-Step Commands

1. **Create a new user harshss:**
2. sudo adduser harshss

Set password during prompt (e.g., hello or any desired value).

1. **Add harshss to the sudo group (give root privileges):**
2. sudo usermod -aG sudo harshss
3. **Verify that the user exists:**
4. cat /etc/passwd | grep harshss

**Output:**

harshss:x:1002:1002:,,,:/home/harshss:/bin/bash

1. **Get the password hash from /etc/shadow:**
2. sudo cat /etc/shadow | grep harshss

**Output (actual hash):**

harshss:$y$j9T$Q5n4TrVlzpKCPt438eE7p1$DqkM9ZemrgBg3ebB.s6i07qZiFCwU7ngHy9KcmZK/S3:20226:0:99999:7:::

## Task 6 – Cracking Password Hashes

## Step-by-Step Instructions

1. **Create a hash file:**
2. nano harshss\_hash.txt

**Paste inside harshss\_hash.txt:**

harshss:$y$j9T$Q5n4TrVlzpKCPt438eE7p1$DqkM9ZemrgBg3ebB.s6i07qZiFCwU7ngHy9KcmZK/S3

1. **Use John the Ripper to crack the hash:**
2. john --wordlist=/usr/share/wordlists/rockyou.txt harshss\_hash.txt

(Make sure *rockyou.txt* is extracted using: *gzip -d /usr/share/wordlists/rockyou.txt.gz*)

1. **Show the cracked password (after some time):**
2. john harshss\_hash.txt --show

**Expected Output (if cracked):**

harshss:hello

1 password hash cracked, 0 left

## Final Result

Username : harshss

Password : hello

## Task 7 – Remediation

### 1. FTP Service (vsftpd)

* **Current Version:** vsftpd 2.3.4
* **Latest Version (as of 2025):** vsftpd 3.0.5
* **Vulnerability:** The version 2.3.4 of vsftpd has a critical backdoor vulnerability that allows attackers to gain root shell access by sending a specially crafted payload.
* **CVE Reference:** [CVE-2011-2523](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-2523)
* **Impact:** Exploiting this vulnerability can give complete system control to an attacker.
* **Remediation Options:**
  + Upgrade to the latest version (vsftpd 3.0.5).
  + Disable FTP entirely and use SFTP (which operates over SSH) as a more secure alternative.
* **Reference Video:** [YouTube – vsftpd backdoor demo](https://www.youtube.com/watch?v=G7nIWUMvn0o)

### 2. SMB (Samba) Service on Port 443

* **Service:** Samba SMB
* **Current Version:** 3.0.20
* **Latest Version (as of May 2025):** Samba 4.20.1
* **Vulnerabilities:**
  + Remote Code Execution (RCE)
  + Null session exploits
  + Arbitrary file read/write
* **Common CVEs:**
  + [CVE-2007-2447](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-2447): Username map script injection
  + [CVE-2017-7494](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-7494): Arbitrary code execution
* **Impact:** Attackers can exploit these flaws to gain unauthorized access, move laterally across systems, or dump user credentials.
* **Remediation Steps:**
  + Disable SMBv1 and allow access only to trusted IP addresses.
  + Upgrade to the latest version of Samba (v4.20.1).
  + Edit /etc/samba/smb.conf to disable guest access and enable detailed logging.
* **Reference Video:** [YouTube – SMB RCE](https://www.youtube.com/watch?v=HPP70Bx0Eck)

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

* **Services:** Rexec, Rlogin, Rsh (classic UNIX networking services)
* **Status:** Outdated, insecure, and deprecated
* **Vulnerabilities:**
  + Transmit usernames and passwords in plaintext
  + Susceptible to Man-in-the-Middle and replay attacks
  + Weak authentication, often relies on .rhosts configuration
* **CVE Reference:** [CVE-1999-0651](https://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0651)
* **Impact:** Attackers can impersonate legitimate users and execute commands remotely without proper authentication.
* **Remediation:** Immediately disable these services using system tools like systemctl, or by removing the relevant packages from the system.

## Major Learnings from This Project

Through this project, I learned several important concepts related to system administration and security on Linux systems:

* **User Management:** I learned how to create and manage user accounts, assign permissions, and understand how user information is stored in system files like /etc/passwd and /etc/shadow.
* **Password Security:** I explored how Linux stores password hashes and how they can be cracked using tools like John the Ripper with the rockyou.txt wordlist.
* **Service Discovery:** I used Nmap to scan systems for open ports, detect services running on those ports, and determine the operating system. Commands like nmap -v, nmap -sV, and nmap -O were particularly useful.
* **Vulnerability Identification:** I analyzed services like SMB, FTP, and R Services, identified their version-based vulnerabilities, and referenced real-world CVEs associated with them.
* **System Hardening:** I learned how to propose practical remediation steps such as disabling legacy protocols, upgrading outdated services, and configuring secure alternatives like SFTP instead of FTP.

Overall, this hands-on experience helped me understand both offensive and defensive aspects of system security. I now have a clearer idea of how to assess a system's weaknesses and implement appropriate fixes to make it more secure.