Esercizio W8D1

DVWA e Burp Suite

Mungiovì Fabio

In questo report vedremo come configurare una DVWA (Damn Vulnerable Web Application) in Kali Linux.
La DVWA ci sarà molto utile per i nostri test, in cui vedremo da vicino le tecniche per sfruttare le vulnerabilità nella fase di exploit.

Dopodiché utilizzeremo lo strumento Burp Suite per intercettare e modificare richieste HTTP durante un login di prova su DVWA

CONFIGURAZIONE

Installazione dei componenti necessari

Per l'esecuzione è necessario installare su Kali i seguenti strumenti:

MySQL: Che fornisce la base dati necessaria per eseguire e testare le funzionalità di DVWA,
Apache Server: Che funge da server web per ospitare l'applicazione vulnerabile.

#Installazione di DVWA

Aprire un terminale e accedere come utente root con il comando `sudo su`.
Clonare il repository DVWA nella directory web di Apache con il comando:

```
git clone https://github.com/digininja/DVWA  /var/www/html/DVWA
```

Modificare i permessi della cartella DVWA per garantire l'accesso:

```
chmod -R 777 /var/www/html/DVWA
```

Configurare il file *config.inc.php* per impostare il database, l'utente e la password.

#Configurazione dei Servizi

MySQL
Avviare il servizio MySQL e creare un utente dedicato con privilegi per DVWA:

```
service mysql start
mysql -u root -p
create user 'kali'@'127.0.0.1' identified by 'kali';
grant all privileges on dvwa.* to 'kali'@'127.0.0.1';
```

Apache
Avviare il servizio Apache e modificare il file *php.ini* per abilitare `allow_url_fopen` e `allow_url_include`.

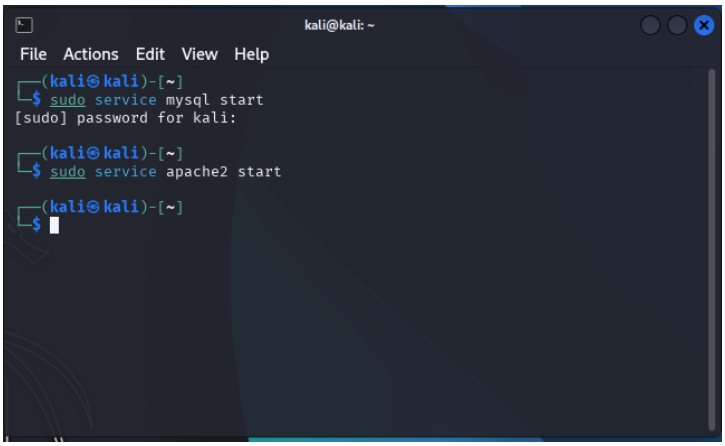
#Configurazione Finale di DVWA

Aprire un browser e accedere a 127.0.0.1/DVWA/setup.php.

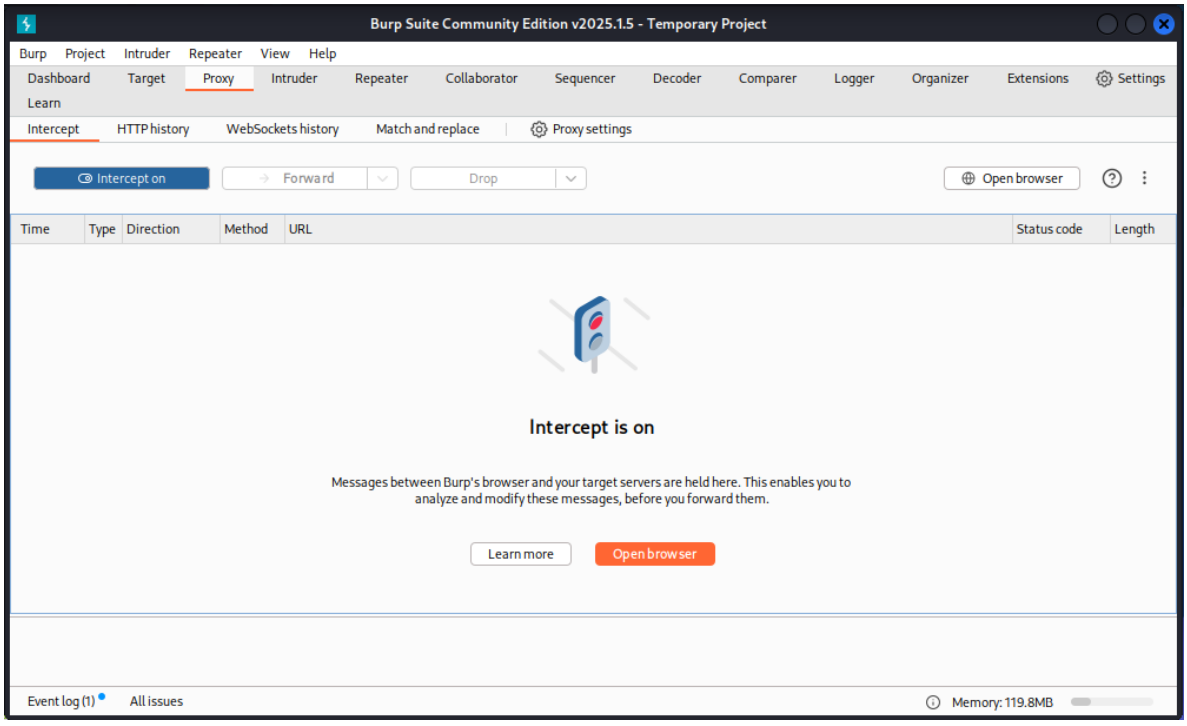
Cliccare su *Create / Reset Database* per finalizzare la configurazione e impostare le credenziali di amministratore.

ESECUZIONE

Ora che il laboratorio è configurato avviamo Apache e Mysql, con i seguenti comandi:



Da adesso la DVWA è attiva e utilizzabile, apriamo quindi Burp Suite, e nella tabella Proxy -> Intercept andiamo ad impostare l'intercept su ON, in questo modo ogni qual volta il Browser, invierà/riceverà qualsiasi tipo di richiesta, resterà in attesa, dandoci modo di intercettare e analizzare i dati, e decidere con il comando *Forward* quando far continuare lo scambio di richieste e risposte con il server.

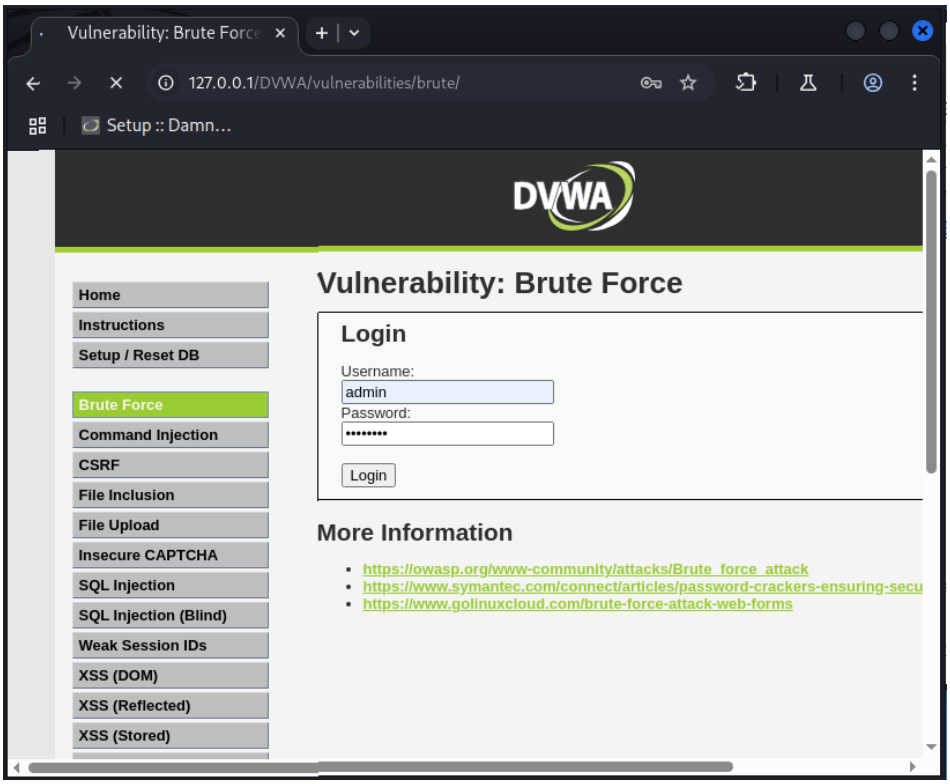


Sempre dalla stessa pagina avviamo il browser integrato di Burp Suite, e colleghiamoci al link 127.0.0.1/DVWA/setup.php.

Dopo aver effettuato l'accesso con le credenziali configurate in precedenza, andiamo nella tab *DVWA security* per settare il livello di sicurezza.
Sono presenti 4 livelli, da LOW a IMPOSSIBLE.
Per il nostro esercizio imposteremo il livello su LOW

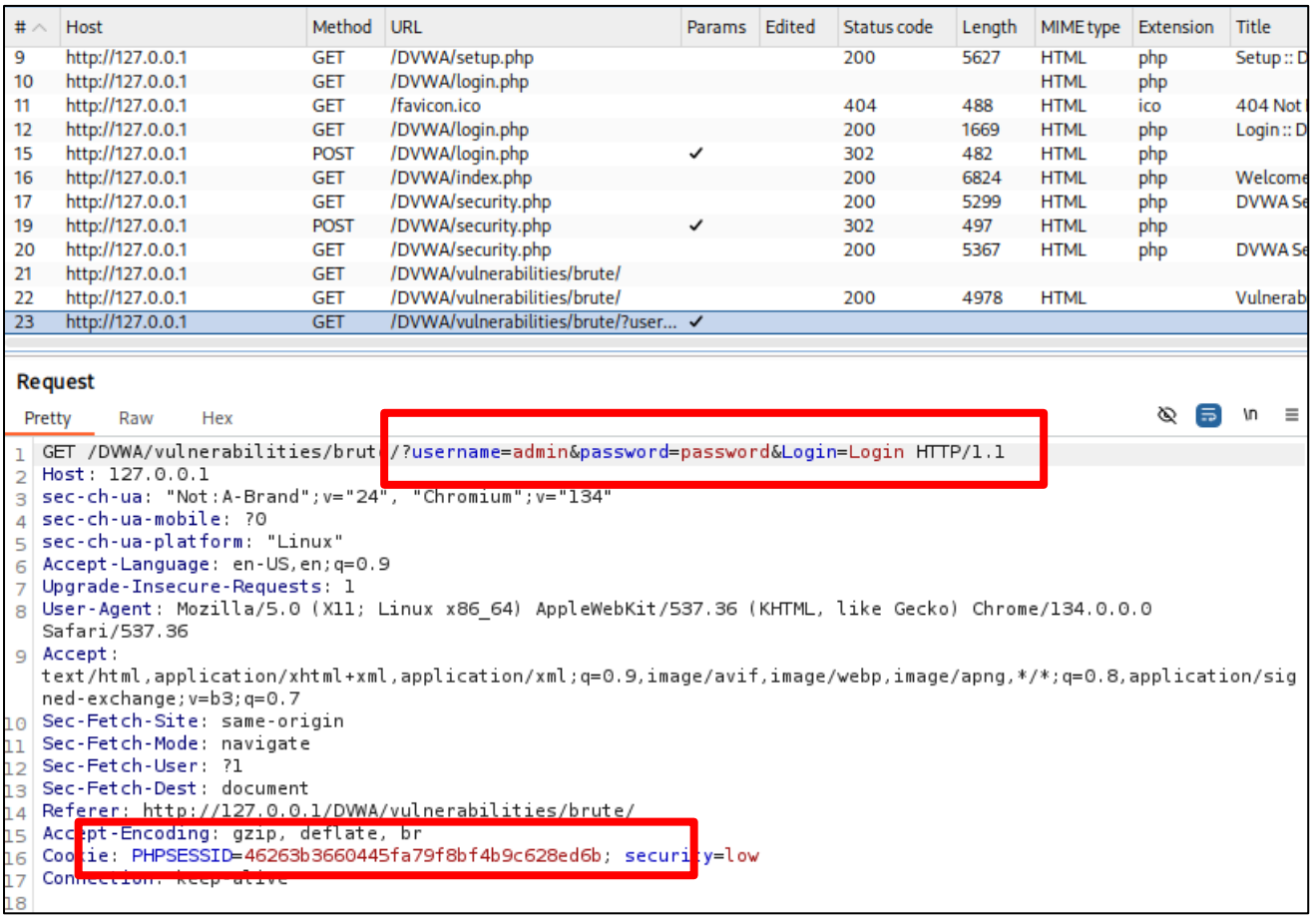
Sempre dal servizio DVWA, entriamo nella tab *Brute Force,* dove viene simulata una funzione di Login.

Il nostro scopo è intercettare i dati potenzialmente immessi da un utente, leggerli e modificarli



Una volta immessi Username e Password e premuto il tasto Login, il browser resterà in attesa per via dell'intercept attivato in precedenza.

Spostandoci ora su Burp Suite, nella tab *HTTP history,* avremo modo di visualizzare la richiesta di login intercettata



#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title
9	http://127.0.0.1	GET	/DVWA/setup.php			200	5627	HTML	php	Setup :: D
10	http://127.0.0.1	GET	/DVWA/login.php					HTML	php	
11	http://127.0.0.1	GET	/favicon.ico			404	488	HTML	ico	404 Not
12	http://127.0.0.1	GET	/DVWA/login.php			200	1669	HTML	php	Login :: D
15	http://127.0.0.1	POST	/DVWA/login.php	✓		302	482	HTML	php	
16	http://127.0.0.1	GET	/DVWA/index.php			200	6824	HTML	php	Welcome
17	http://127.0.0.1	GET	/DVWA/security.php			200	5299	HTML	php	DVWA Se
19	http://127.0.0.1	POST	/DVWA/security.php	✓		302	497	HTML	php	
20	http://127.0.0.1	GET	/DVWA/security.php			200	5367	HTML	php	DVWA Se
21	http://127.0.0.1	GET	/DVWA/vulnerabilities/brute/							
22	http://127.0.0.1	GET	/DVWA/vulnerabilities/brute/			200	4978	HTML		Vulnerab
23	http://127.0.0.1	GET	/DVWA/vulnerabilities/brute/?user...	✓						

Request

Pretty Raw Hex

```
1  GET /DVWA/vulnerabilities/brut /?username=admin&password=password&Login=Login HTTP/1.1
2  Host: 127.0.0.1
3  sec-ch-ua: "Not:A-Brand";v="24", "Chromium";v="134"
4  sec-ch-ua-mobile: ?0
5  sec-ch-ua-platform: "Linux"
6  Accept-Language: en-US,en;q=0.9
7  Upgrade-Insecure-Requests: 1
8  User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0
   Safari/537.36
9  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/sig
   ned-exchange;v=b3;q=0.7
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Referer: http://127.0.0.1/DVWA/vulnerabilities/brute/
15 Accept-Encoding: gzip, deflate, br
16 Cookie: PHPSESSID=46263b3660445fa79f8bf4b9c628ed6b; security=low
17 Connection: keep-alive
18
```

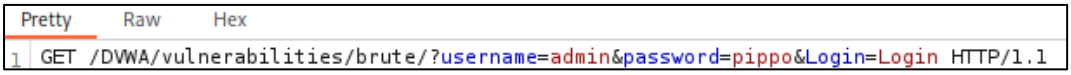
Nell'immagine della richiesta GET intercettata, possiamo notare alcune criticità di sicurezza, date dal fatto che il livello di sicurezza è stato volutamente impostato al livello più basso.

Come prima cosa si nota come Username e Password, utilizzate per il Login vengano inviate all'interno dell'URL, il che le rende facilmente visualizzabili da chiunque intercetti questa richiesta

Un'altra criticità è data dall'esposizione del cookie PHPSESSID, che permette l'identificazione univoca di un utente da parte del server, che quindi potrebbe essere intercettata e utilizzata per un furto di identità

Andiamo ora a modificare la richiesta intercettata per analizzare la risposta del server:

Modifico il campo password, sostituendo la password inserita dall'utente con la parola *pippo*



Facendo ora continuare il processo con la funzione Forward andiamo ad intercettare la risposta del server, che, come ci si aspettava risponderà con un accesso negato, essendo stata sostituita la password con una errata