

Esercizio S3_L3

Consegna:

Avviare un server apache con database Mysql e tentare di intercettare il traffico tramite la Burp Suite.

Prerequisiti:

- Kali VirtualMachine
- Scheda di rete con accesso ad internet
- VM aggiornata `sudo apt update && sudo apt upgrade`
- Burp Suite installato

Svolgimento:

1. Download repository

Come prima cosa, mi sono spostato nella directory root del server web e ho clonato il repository di DVWA da GitHub:

```
cd /var/www/html
```

```
sudo git clone https://github.com/digininja/DVWA
```



```
(kali㉿kali)-[~]  
$ cd /var/www/html  
  
(kali㉿kali)-[/var/www/html]  
$ sudo git clone https://github.com/digininja/DVWA  
Cloning into 'DVWA' ...  
remote: Enumerating objects: 5373, done.  
remote: Total 5373 (delta 0), reused 0 (delta 0), pack-reused 5373 (from 1)  
Receiving objects: 100% (5373/5373), 2.58 MiB | 3.62 MiB/s, done.  
Resolving deltas: 100% (2667/2667), done.
```

2. Permessi e file di configurazione

Ho poi assegnato i permessi di scrittura alla cartella:

```
sudo chmod -R 777 DVWA/
```

Mi sono successivamente spostato nella cartella di configurazione:

```
cd DVWA/config
```

E ho copiato il file di configurazione di esempio:

```
cp config.inc.php.dist config.inc.php
```

```
(kali㉿kali)-[/var/www/html]
$ sudo chmod -R 777 DVWA/

(kali㉿kali)-[/var/www/html]
$ cd DVWA/config

(kali㉿kali)-[/var/www/html/DVWA/config]
$ cp config.inc.php.dist config.inc.php

(kali㉿kali)-[/var/www/html/DVWA/config]
$ nano config.inc.php
```

Ho quindi modificato il file appena copiato con l'editor di testo nano:

```
nano config.inc.php
```

Impostando i seguenti parametri di connessione al database:

```
$_DVWA['db_user']    = 'kali';
$_DVWA['db_password'] = 'kali';
$_DVWA['db_port']     = '3306';
```

```
# See README.md for more information on this.
$_DVWA = array();
$_DVWA[ 'db_server' ] = getenv('DB_SERVER') ?: '127.0.0.1';
$_DVWA[ 'db_database' ] = getenv('DB_DATABASE') ?: 'dvwa';
$_DVWA[ 'db_user' ] = 'kali';
$_DVWA[ 'db_password' ] = 'kali';
$_DVWA[ 'db_port' ] = '3306';
```

3. Configurazione del database (MariaDB)

Ho avviato il servizio MariaDB:

```
sudo service mysql start
```

Sono entrato nel prompt di MariaDB come root:

```
mysql -u root -p
```

Una volta dentro, ho eseguito i seguenti comandi:

```
CREATE USER 'kali'@'127.0.0.1' IDENTIFIED BY 'kali';
CREATE DATABASE dvwa;
GRANT ALL PRIVILEGES ON dvwa.* TO 'kali'@'127.0.0.1';
FLUSH PRIVILEGES;
```

```
(root@kali)-[/etc/php/8.4/apache2]
# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 44
Server version: 11.8.2-MariaDB-1 from Debian -- Please help get to 10k stars at https://github.com/MariaDB/Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create user 'kali'@'127.0.0.1' identified by 'kali';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> █
```

```
(kali@kali)-[~]
$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 54
Server version: 11.8.2-MariaDB-1 from Debian -- Please help get to 10k stars at https://github.com/MariaDB/Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE dvwa;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON dvwa.* TO 'kali'@'127.0.0.1' IDENTIFIED BY 'kali';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> █
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> EXIT
Bye
```

4. Configurazione php

Ho poi proceduto aprendo il file di configurazione PHP:

sudo nano /etc/php/8.4/apache2/php.ini

```
(root@kali)-[/etc/php/8.4/apache2]
# ls
conf.d  php.ini

(root@kali)-[/etc/php/8.4/apache2]
# nano php.ini
```

E mi sono assicurato che le seguenti voci fosse settate su "On":

```
; Whether to allow the treatment of URLs (like http:// or ftp://) as files.
; https://php.net/allow-url-fopen
allow_url_fopen = On

; Whether to allow include/require to open URLs (like https:// or ftp://) as files.
; https://php.net/allow-url-include
allow_url_include = On█
```

5. Avvio dei servizi

Ho avviato Apache:

```
sudo service apache2 start
```

```
(root@kali)-[/var/www/html/DVWA/config]
# service apache2 start
```

E abilitato il modulo mod_rewrite, utile per DVWA:

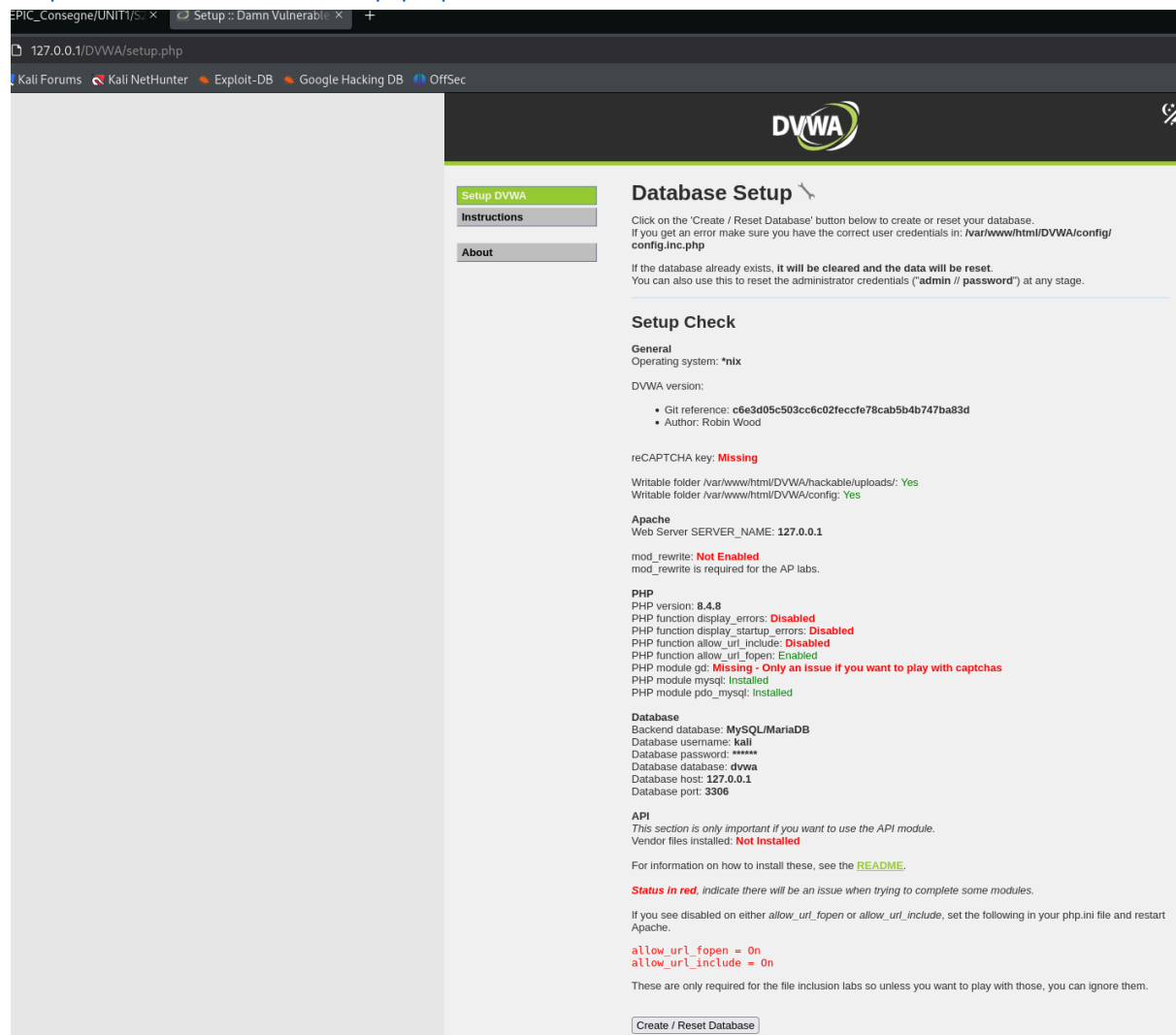
```
sudo a2enmod rewrite
```

```
sudo systemctl restart apache2
```

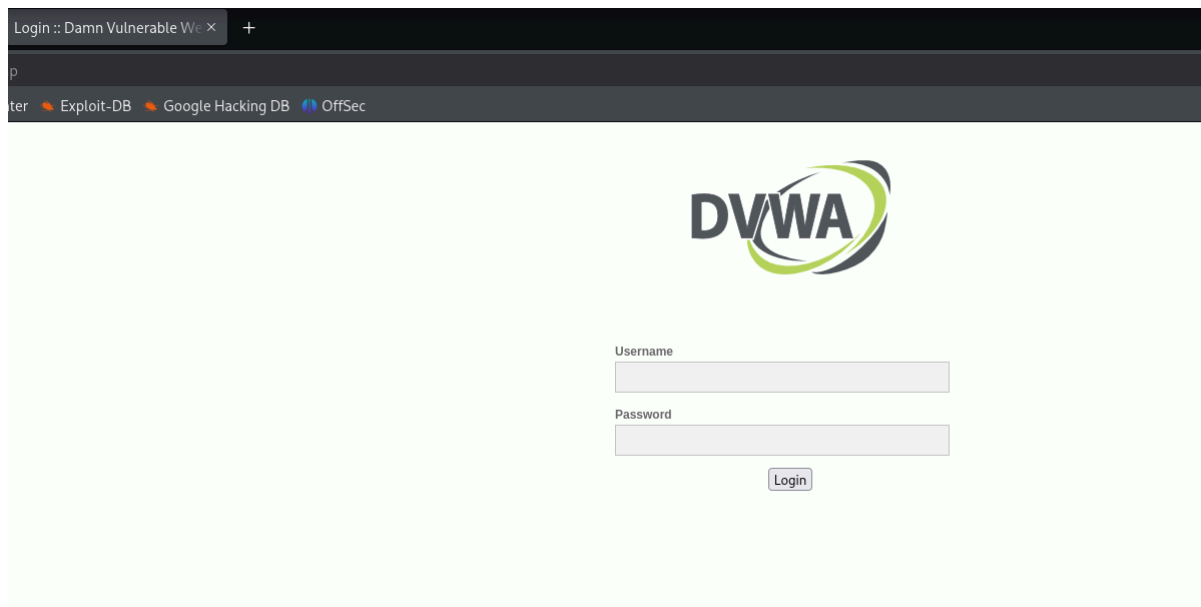
6. Setup finale di DVWA

Ho aperto il browser all'indirizzo:

<http://127.0.0.1/DVWA/setup.php>

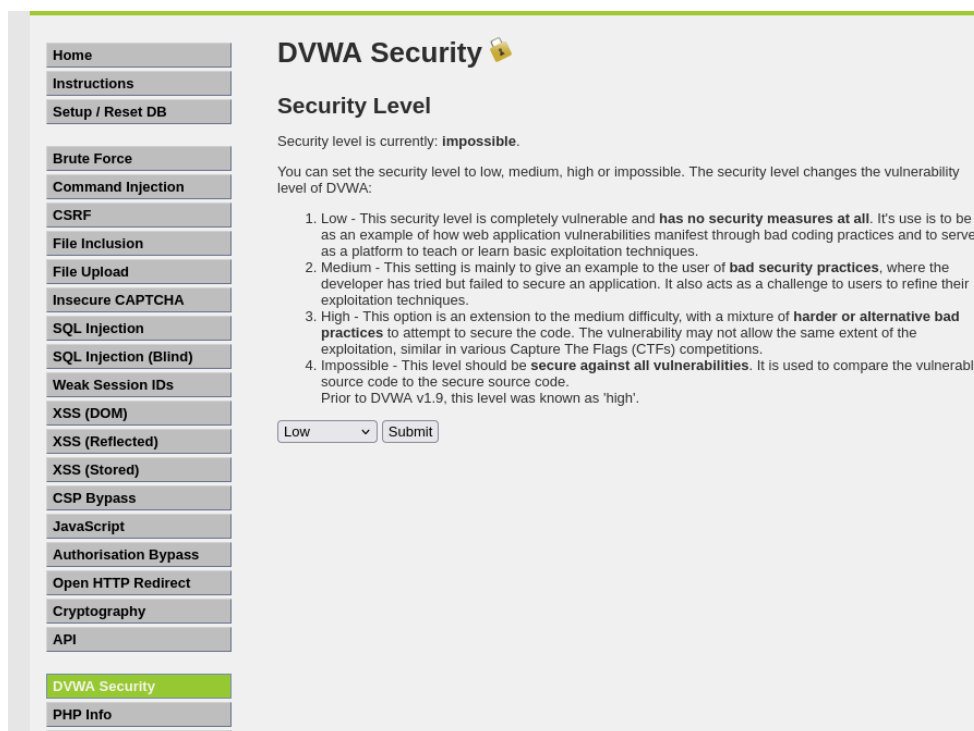


Nella pagina, ho cliccato su “Create / Reset Database” per inizializzare il database.



A questo punto DVWA è pronto per essere utilizzato. Posso accedere al pannello login con le credenziali *username: admin* & *psw: password* e iniziare i test su vulnerabilità web in ambiente controllato.

Una volta aperto mi sono trasferito nella sezione DVWA Security e settare il livello sicurezza su *Low*.



7. Intercettazione Traffico

A seguito di ciò è stato possibile avviare la burp suite per provare ad intercettare il traffico; ho effettuato il logout dal database ed ho avviato il browser di Burp.

Ho settato Burp in modalità **Proxy > Intercept on**

Sono tornato nella pagina di login del database ed ho inserito le credenziali; burp ha intercettato la richiesta e mi è stato possibile leggere in chiaro le credenziali che il client sta passando al server per effettuare il login.

The screenshot shows the Burp Suite interface with the 'Intercept' tab selected. The 'Intercept on' button is active, and the 'Forward' button is highlighted. A table of intercepted requests is visible, showing a POST request to 'http://127.0.0.1/DVWA/login.php' at 09:23:40. The 'Request' tab is open, displaying the raw HTTP request details. The request is a POST to 'http://127.0.0.1/DVWA/login.php' with a 'Content-Type' of 'application/x-www-form-urlencoded'. The body of the request contains the login credentials: 'username=admin&password=password&login=Login&user_token=9bdef30587364a839f22f701e9fb25fe'. The browser window on the right shows the DVWA login page with the username 'admin' and password 'password' entered.

Time	Type	Direction	Method	URL
09:23:40.16 J...	HTTP	→ Request	POST	http://127.0.0.1/DVWA/login.php

Request

Pretty Raw Hex

```
8 Accept-Language: en-us,en;q=0.9
9 Origin: http://127.0.0.1
10 Content-Type: application/x-www-form-urlencoded
11 Upgrade-Insecure-Requests: 1
12 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4431.24 Safari/537.36
13 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/svg+xml,*/*;q=0.8
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-User: ?1
17 Sec-Fetch-Dest: document
18 Referer: http://127.0.0.1/DVWA/login.php
19 Accept-Encoding: gzip, deflate, br
20 Cookie: security=impossible; PHPSESSID=1dc53be03b53a5ee0da528bacc68ad7c
21 Connection: keep-alive
22
23 username=admin&password=password&login=Login&user_token=9bdef30587364a839f22f701e9fb25fe
```

Per sperimentare ho poi cambiato quelle credenziali con *username:prova&password:prova*.

Ho quindi inviato la richiesta modificata al **Repeater**, dove ho potuto testare diverse varianti della richiesta e osservare la risposta del server **senza dover reinserire i dati dal browser**.

The screenshot shows the Burp Suite interface with the 'Repeater' tab selected. The 'Forward' button is highlighted. A table of intercepted requests is visible, showing a POST request to 'http://127.0.0.1/DVWA/login.php' at 09:23:40. The 'Request' tab is open, displaying the raw HTTP request details. The request is a POST to 'http://127.0.0.1/DVWA/login.php' with a 'Content-Type' of 'application/x-www-form-urlencoded'. The body of the request contains the modified login credentials: 'username=prova&password=prova&login=Login&user_token=9bdef30587364a839f22f701e9fb25fe'. The browser window on the right shows the DVWA login page with the username 'prova' and password 'prova' entered.

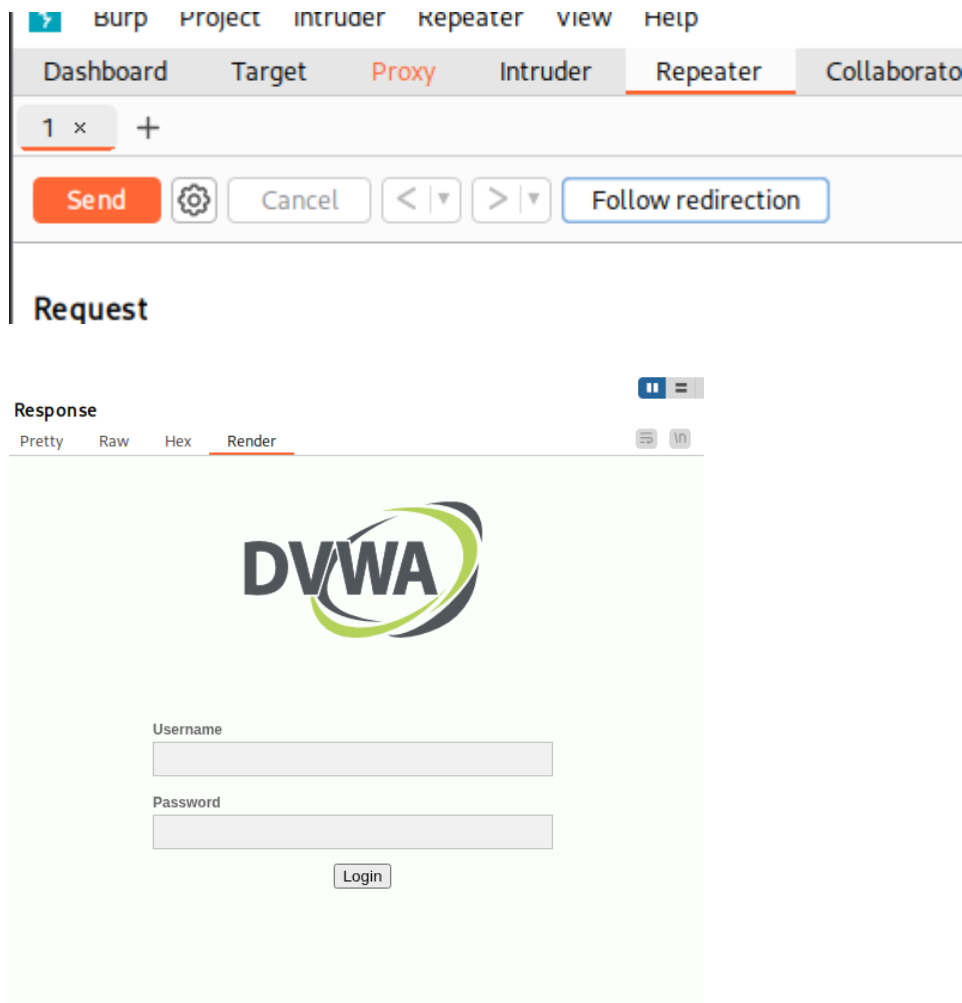
Time	Type	Direction	Method	URL
09:23:40.16 J...	HTTP	→ Request	POST	http://127.0.0.1/DVWA/login.php

Request

Pretty Raw Hex

```
8 Accept-Language: en-us,en;q=0.9
9 Origin: http://127.0.0.1
10 Content-Type: application/x-www-form-urlencoded
11 Upgrade-Insecure-Requests: 1
12 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4431.24 Safari/537.36
13 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/svg+xml,*/*;q=0.8
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-User: ?1
17 Sec-Fetch-Dest: document
18 Referer: http://127.0.0.1/DVWA/login.php
19 Accept-Encoding: gzip, deflate, br
20 Cookie: security=impossible; PHPSESSID=1dc53be03b53a5ee0da528bacc68ad7c
21 Connection: keep-alive
22
23 username=prova&password=prova&login=Login&user_token=9bdef30587364a839f22f701e9fb25fe
```

Dopo aver premuto su **Send** e poi **Follow redirection**, ho verificato che l'accesso non andava a buon fine, dimostrando così l'efficacia dell'intercettazione e manipolazione del traffico tra client e server.



Conclusione:

L'esercizio ha permesso di:

- Installare e configurare correttamente un ambiente DVWA
- Comprendere le interazioni tra Apache, MySQL e PHP
- Analizzare richieste HTTP tramite Burp Suite
- Visualizzare e manipolare credenziali di autenticazione in chiaro

Un setup ed un'attività estremamente utili per prendere confidenza con l'analisi del traffico web e lo studio delle vulnerabilità applicative.