

LABORATORIO 11/12/2024 S3-L3

Traccia: Nella lezione pratica di oggi vedremo come configurare una DVWA – ovvero damn vulnerable web application in Kali Linux.

Per farlo seguiremo tutte le direttive presenti nelle slide e i relativi comandi:

Per installare la DVWA abbiamo bisogno di 3 componenti:

- Kali Linux.
- Database MySQL
- Web Server Apache

Apriamo un terminale su Kali, utilizzando l'utenza di root, eseguendo il comando «sudo su» e poi eseguiamo i comandi:

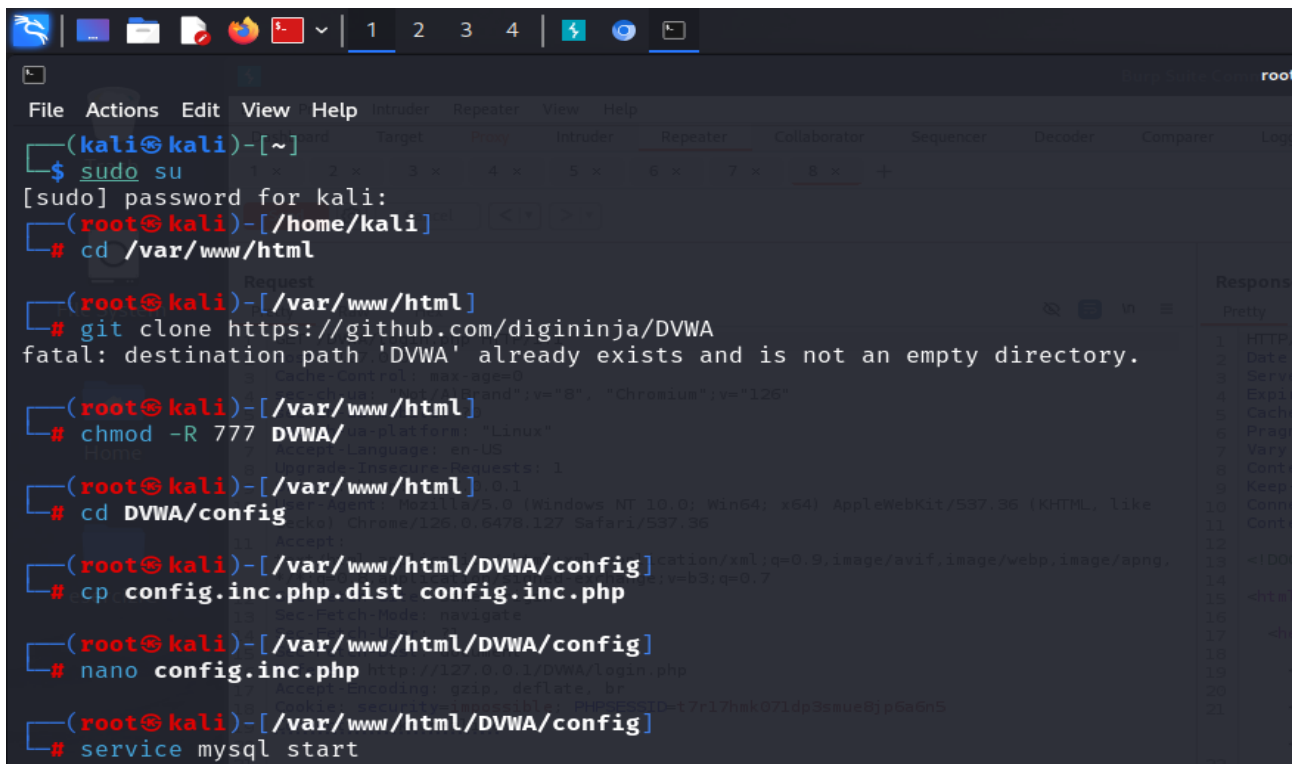
- `cd /var/www/html`
- `git clone https://github.com/digininja/DVWA`
- `chmod -R 777 DVWA/`
- `cd DVWA/config`

- `cp config.inc.php.dist config.inc.php`
- `nano config.inc.php`

All'interno del file `config.inc.php` cambiamo utente e password di default (inserendo, `user:kali, password:kali.`)

Sempre con utenza di root su Kali, facciamo partire il servizio `mysql` con il comando:

`service mysql start`



```
(kali@kali)-[~]
└─$ sudo su
[sudo] password for kali:
(kali@kali)-[/home/kali]
└─# cd /var/www/html

(kali@kali)-[/var/www/html]
└─# git clone https://github.com/digininja/DVWA
fatal: destination path 'DVWA' already exists and is not an empty directory.

(kali@kali)-[/var/www/html]
└─# chmod -R 777 DVWA/

(kali@kali)-[/var/www/html]
└─# cd DVWA/config

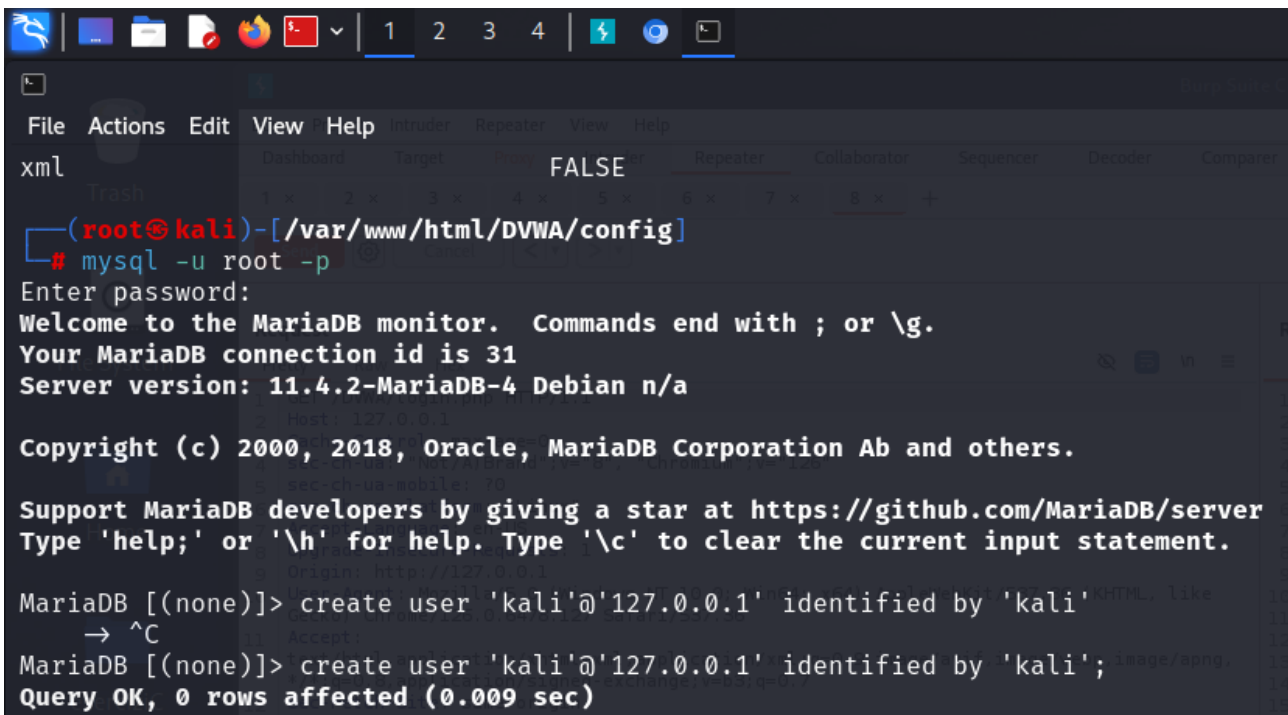
(kali@kali)-[/var/www/html/DVWA/config]
└─# cp config.inc.php.dist config.inc.php

(kali@kali)-[/var/www/html/DVWA/config]
└─# nano config.inc.php

(kali@kali)-[/var/www/html/DVWA/config]
└─# service mysql start
```

poi connettiamoci al db con utenza di root con il comando seguente:

mysql -u root -p



```
(root@kali)-[/var/www/html/DVWA/config]
# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 11.4.2-MariaDB-4 Debian n/a

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

Support MariaDB developers by giving a star at https://github.com/MariaDB/server
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'
→ ^C
MariaDB [(none)]> create user 'kali'@'127.0.0.1' identified by 'kali';
Query OK, 0 rows affected (0.009 sec)
```

Creiamo un'utenza sul db con il seguente comando:

create user 'kali'@'127.0.0.1' identified by 'kali' ;

successivamente assegniamo i privilegi all'utente kali con il seguente comando:

grant all privileges on dvwa.* to 'kali'@'127.0.0.1'

identified by 'kali' ;

ed usciamo utilizzando "exit".

Ora che il servizio mysql è configurato, passiamo al servizio apache (il web server).

Facciamo partire il servizio con il comando:

service apache2 start

```
MariaDB [(none)]> grant all privileges on dvwa.* to 'kali'@'127.0.0.1' identified by 'kali';
Query OK, 0 rows affected (0.003 sec)

MariaDB [(none)]> exit
Bye

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

(root@kali)-[/var/www/html/DVWA/config]
# cd /etc/php/8.2/apache2

(root@kali)-[/etc/php/8.2/apache2]
# service apache2 start
```

Spostiamoci nella cartella /etc/php/8.1/apache2 con il comando:

cd /etc/php/8.1/apache2

Utilizziamo nano per modificare il file php.ini all'interno della cartella apache2.

Modifichiamo le voci allow_url_fopen e allow_url_include

```
root@kali: /etc/php/8.2/apache2
File Actions Edit View Help
GNU nano 8.2 php.ini *
; Temporary directory for HTTP uploaded files (will use system default if not
; specified).
; https://php.net/upload-tmp-dir
upload_tmp_dir =

; Maximum allowed size for uploaded files.
; https://php.net/upload-max-filesize
upload_max_filesize = 2M

; Maximum number of files that can be uploaded via a single request
max_file_uploads = 20

;;;;;;;;;;;;;;;;;;;;;;;;;
; Fopen wrappers ;
;;;;;;;;;;;;;;;;;;;;;;;;;

; 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

; Define the anonymous ftp password (your email address). PHP's default setting
; for this is empty.
; https://php.net/from
from="john@doe.com"

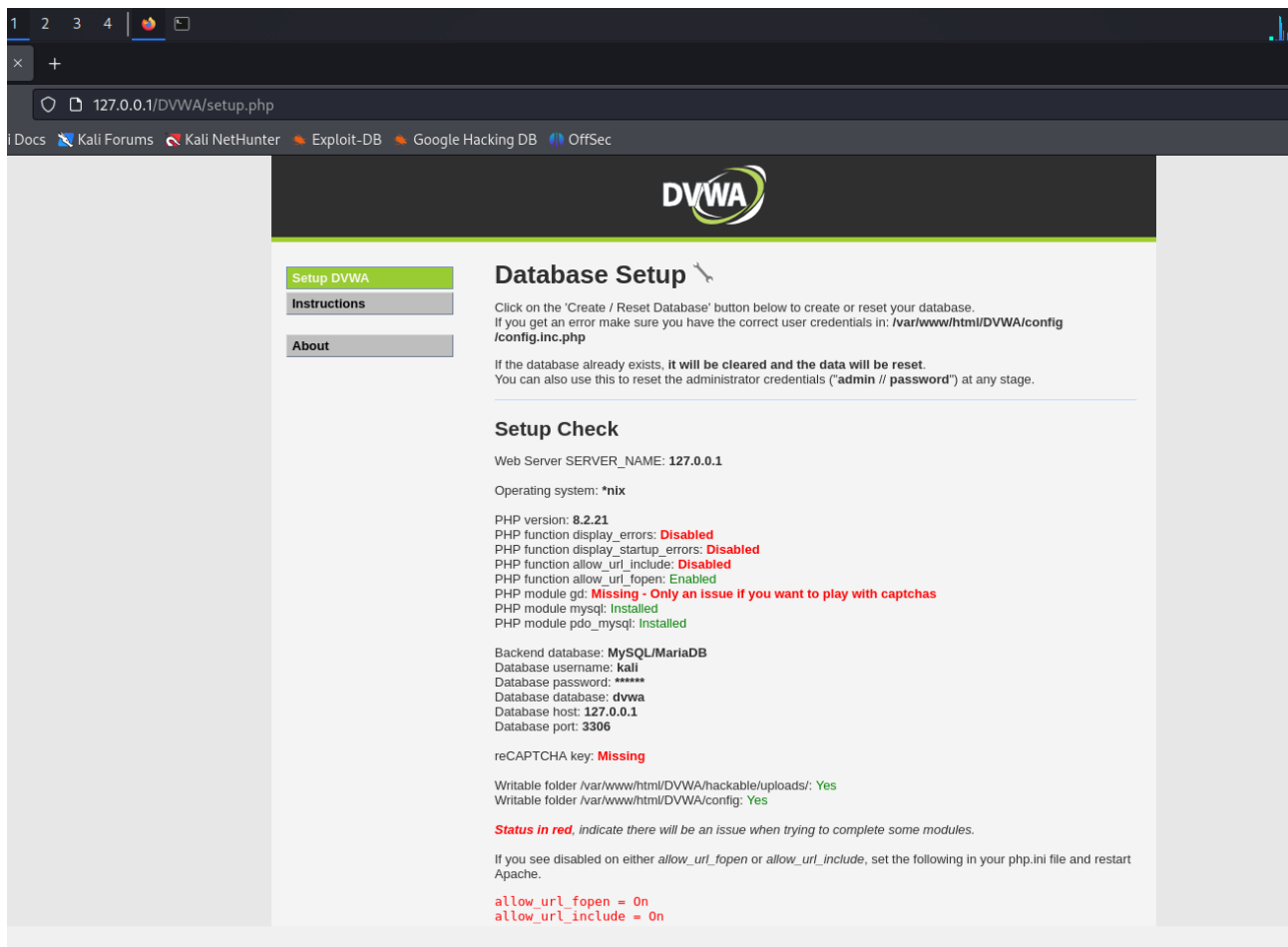
; Define the User-Agent string. PHP's default setting for this is empty.
; https://php.net/user-agent
user_agent="PHP"

; Default timeout for socket based streams (seconds)
```

Eseguiamo nuovamente il comando `service apache2 start`.

**A questo punto apriamo una sessione del vostro browser
e scriviamo nella barra degli indirizzi:**

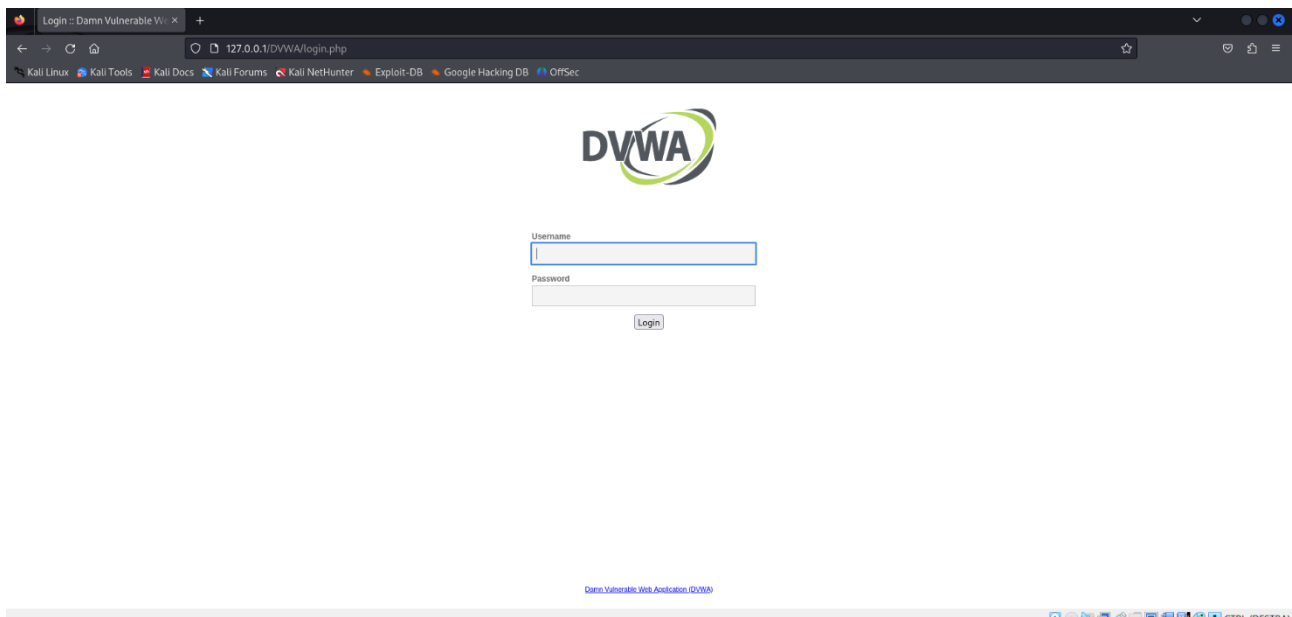
127.0.0.1/DVWA/setup.php



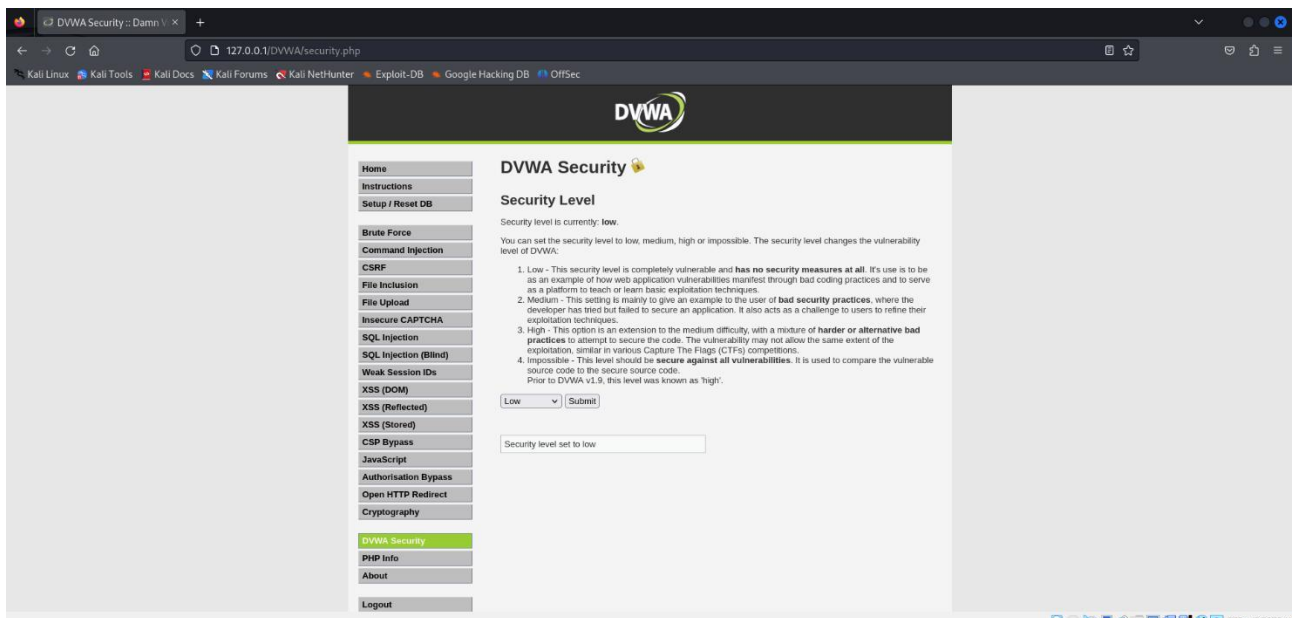
Clicchiamo su «Create / Reset Database» nella parte bassa della pagina. Appena creato il database veniamo rediretti su una pagina di login, dove possiamo entrare inserendo le credenziali di admin di default.

User: admin

Password: password

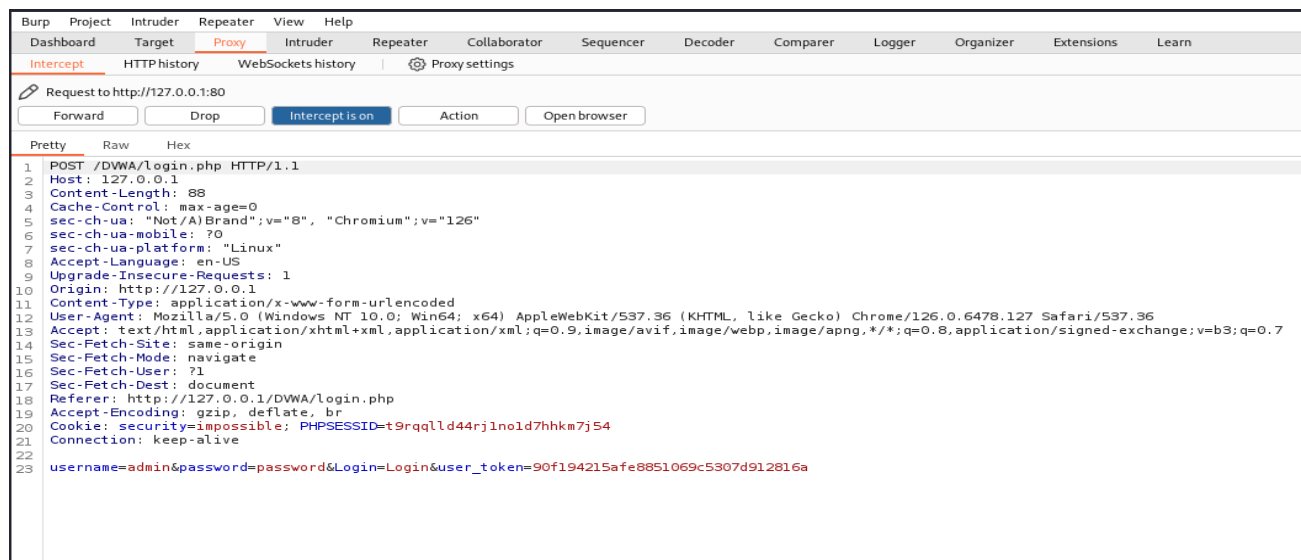


Una volta entrati nell'app, clicchiamo sulla scheda DVWA Security. Qui possiamo scegliere il livello di sicurezza dell'APP. Selezioneremo il livello più basso.



Lanciamo Burpsuite, scegliamo un progetto temporaneo ed apriamo un browser, inserendo l'indirizzo della nostra DVWA: 127.0.0.1/DVWA e inseriamo nei campi login e password i valori «admin» e «password» rispettivamente.

Intercettiamo la richiesta con burp e vediamo come possiamo modificarla. Guardiamo i parametri di login, possiamo modificarli a nostro piacimento prima di inviare la richiesta all'app.



Proviamo a modificare i campi, ed inviare la richiesta inserendo delle credenziali sicuramente errate. Prima di inviare la richiesta, clicchiamo con il tasto destro e selezioniamo «send to repeater»

Clicchiamo su send per inviare la richiesta di login ed e poi su follow redirection.

The screenshot displays the Burp Suite interface with the 'Repeater' tab selected. The 'Send' button is highlighted in red. The 'Request' pane on the left shows an HTTP GET request to /DWA/login.php. The 'Response' pane on the right shows the corresponding HTML response, which includes a title 'Login :: Damn Vulnerable Web Application (DWA)' and a login form. The 'Inspector' pane on the far right shows the request and response headers and body.

Request

```
1 GET /DWA/login.php HTTP/1.1
2 Host: 127.0.0.1
3 Cache-Control: max-age=0
4 sec-ch-ua: "Not(A)Brand";v="8", "Chromium";v="126"
5 sec-ch-ua-mobile: ?0
6 sec-ch-ua-platform: "Linux"
7 Accept-Language: en-US
8 Upgrade-Insecure-Requests: 1
9 Origin: http://127.0.0.1
10 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36
11 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: navigate
14 Sec-Fetch-User: ?1
15 Sec-Fetch-Dest: document
16 Referer: http://127.0.0.1/DWA/login.php
17 Accept-Encoding: gzip, deflate, br
18 Cookie: security=impossible; PHPSESSID=t7r17hak071dp3smue8jp6a6n5
19 Connection: keep-alive
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Wed, 11 Dec 2024 15:44:35 GMT
3 Server: Apache/2.4.62 (Debian)
4 Expires: Tue, 23 Jun 2009 12:00:00 GMT
5 Cache-Control: no-cache, must-revalidate
6 Pragma: no-cache
7 Vary: Accept-Encoding
8 Content-Length: 1942
9 Keep-Alive: timeout=5, max=100
10 Connection: Keep-Alive
11 Content-Type: text/html; charset=utf-8
12
13 <!DOCTYPE html>
14
15 <html lang="en-GB">
16
17 <head>
18
19 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
20
21 <title>
22 Login :: Damn Vulnerable Web Application (DWA)
23 </title>
24
25 <link rel="stylesheet" type="text/css" href="dwa/css/login.css" />
26
27 </head>
28
29 <body>
30
31 <div id="wrapper">
32
33 <div id="header">
34
35 <br />
36
37 <p>
38 
39 </p>
40
41 <br />
42
43 </div>
```

Inspector

- Request attributes: 2
- Request query parameters: 0
- Request body parameters: 0
- Request cookies: 2
- Request headers: 18
- Response headers: 10

1,670 bytes | 1,006 millis