

ESERCIZIO S3L2

Nell'esercizio di oggi si vedrà come configurare una DVWA (damn vulnerable web application).

Inizialmente si è installato la web application DVWA, un progetto software che include intenzionalmente vulnerabilità di sicurezza, esso ci servirà per proseguire con l'esercizio richiesto. Successivamente si è cambiato nome utente e password nel file "config.inc.php" entrando con il comando "sudo nano config.inc.php" ed inserendo per entrambi 'kali'

```
File Actions Edit View Help
GNU nano 7.2 config.inc.php
<?php

# If you are having problems connecting to the MySQL database and all of
# try changing the 'db_server' variable from localhost to 127.0.0.1. Fixe
# Thanks to @digininja for the fix.

# Database management system to use
$dbms = 'MySQL';
#$dbms = 'PgSQL'; // Currently disabled

# Database variables
# WARNING: The database specified under db_database WILL BE ENTIRELY DE
# Please use a database dedicated to DVWA.
#
# If you are using MariaDB then you cannot use root, you must use create
# See README.md for more information on this.
$_DVWA = array();
$_DVWA['db_server'] = getenv('DB_SERVER') ? '127.0.0.1';
$_DVWA['db_database'] = 'dvwa';
$_DVWA['db_user'] = 'kali';
$_DVWA['db_password'] = 'kali';
$_DVWA['db_port'] = '3306';

# ReCAPTCHA settings
# Used for the 'Insecure CAPTCHA' module
# You'll need to generate your own keys at: https://www.google.com/reca
$_DVWA['recaptcha_public_key'] = '';
$_DVWA['recaptcha_private_key'] = '';

# Default security level
# Default value for the security level with each session.
# The default is 'impossible'. You may wish to set this to either 'low'
$_DVWA['default_security_level'] = 'impossible';
```

Dopo aver configurato le nostre credenziali abbiamo avviato il servizio web **Apache2** e il servizio database **mysql**, utilizzando i permessi di amministratore tramite il comando 'sudo'. **mysql** si è avviato tramite il database MariaDB ed abbiamo creato un'utenza (kali) assegnandogli i privilegi da amministratore:

```
kali@kali: /etc/php/8.2/apache2
File Actions Edit View Help
(kali@kali)-[/var/www/html/DVWA/config]
$ mysql -u root -p
Enter password:
ERROR 1698 (28000): Access denied for user 'root'@'localhost'

(kali@kali)-[/var/www/html/DVWA/config]
$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 32
Server version: 10.11.6-MariaDB-2 Debian n/a

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

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

MariaDB [(none)]> create user 'kali'@'127.0.0.1' identified by 'kali' ;
Query OK, 0 rows affected (0.014 sec)
```

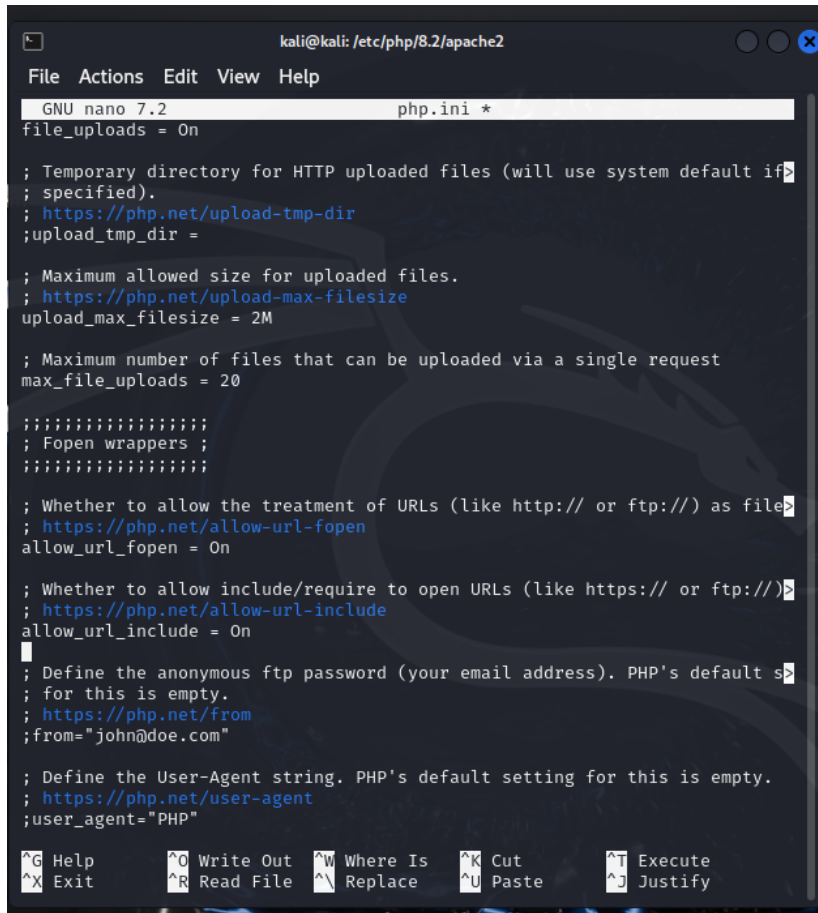
```

MariaDB [(none)]> grant all privileges on dvwa.* to 'kali'@'127.0.0.1' identified by 'kali' ;
Query OK, 0 rows affected (0.013 sec)

MariaDB [(none)]> exit

```

Si è effettuata anche la configurazione per il servizio web **apache2**, in particolare si è modificato il file 'php.ini' per consentire la richiesta di aprire gli URLs.



```

kali@kali: /etc/php/8.2/apache2
File Actions Edit View Help
GNU nano 7.2 php.ini *
file_uploads = On

; Temporary directory for HTTP uploaded files (will use system default if
; 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 file
; https://php.net/allow-url-fopen
allow_url_fopen = On

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

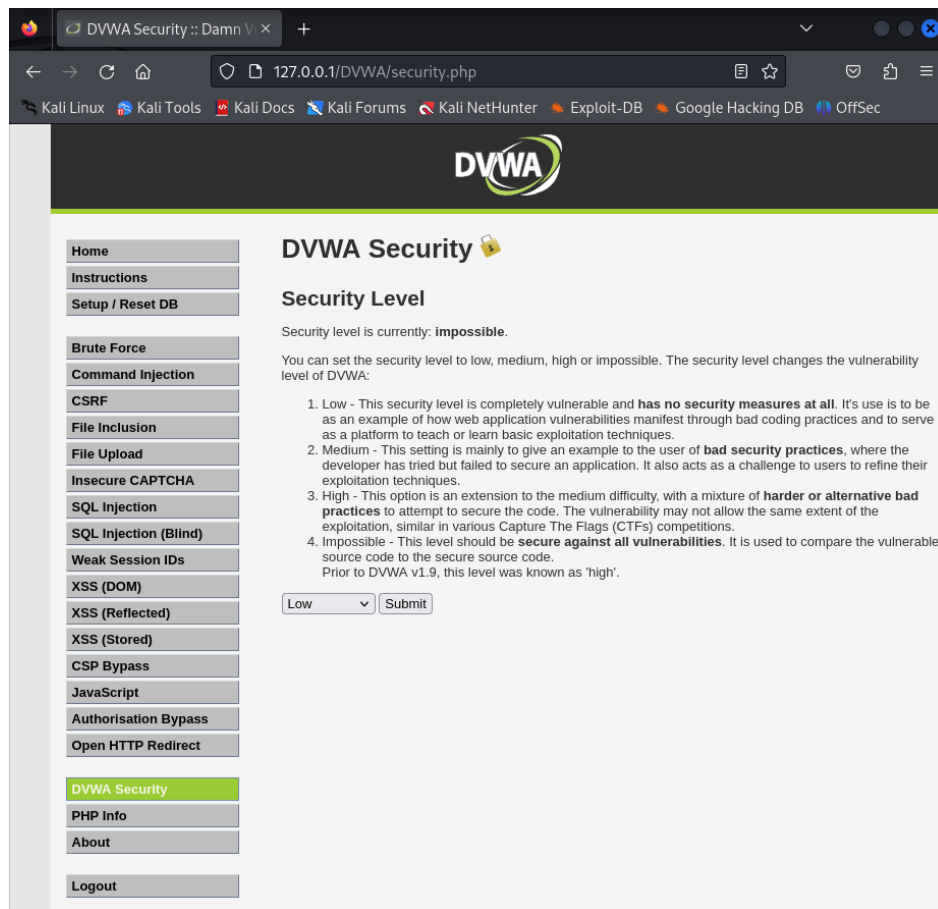
; Define the anonymous ftp password (your email address). PHP's default s
; for this is empty.
; https://php.net/from
;from="john@doe.com"

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

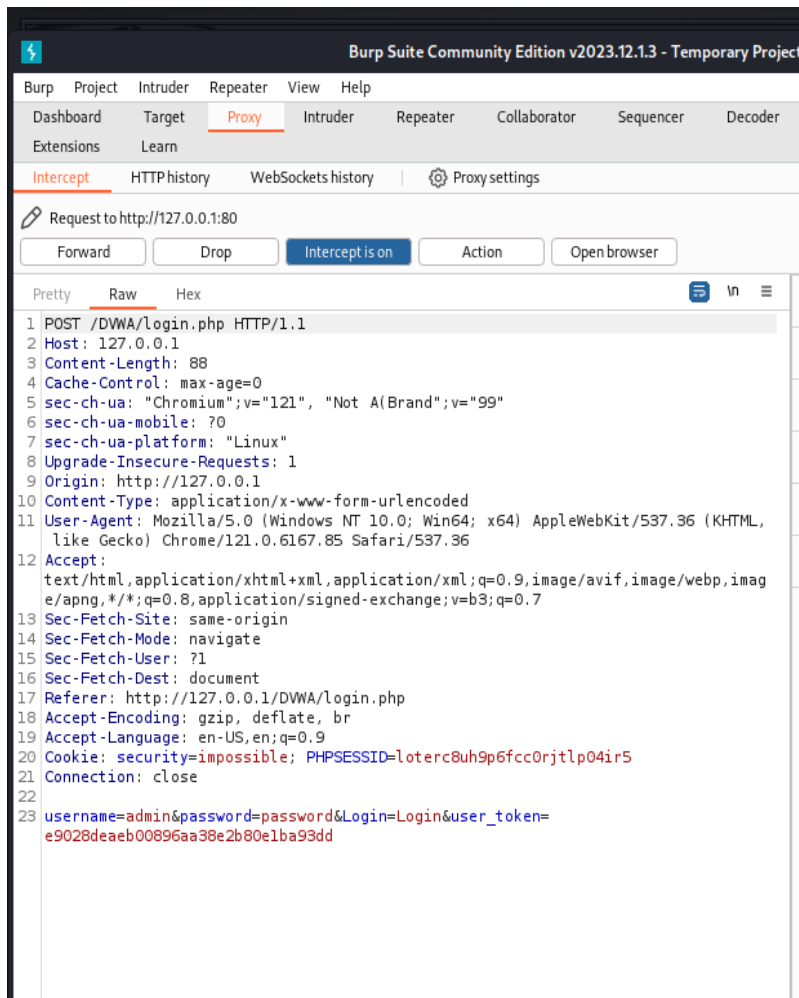
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute
^X Exit      ^R Read File ^N Replace   ^U Paste     ^J Justify

```

Successivamente si è creato un database su DVWA, si è effettuato l'accesso con username(**admin**) e password(**password**) e si è scelto il livello di sicurezza della web app settandolo al minimo (Low). Più basso sarà il livello di sicurezza impostato, meno sarà complicato sfruttare le vulnerabilità.



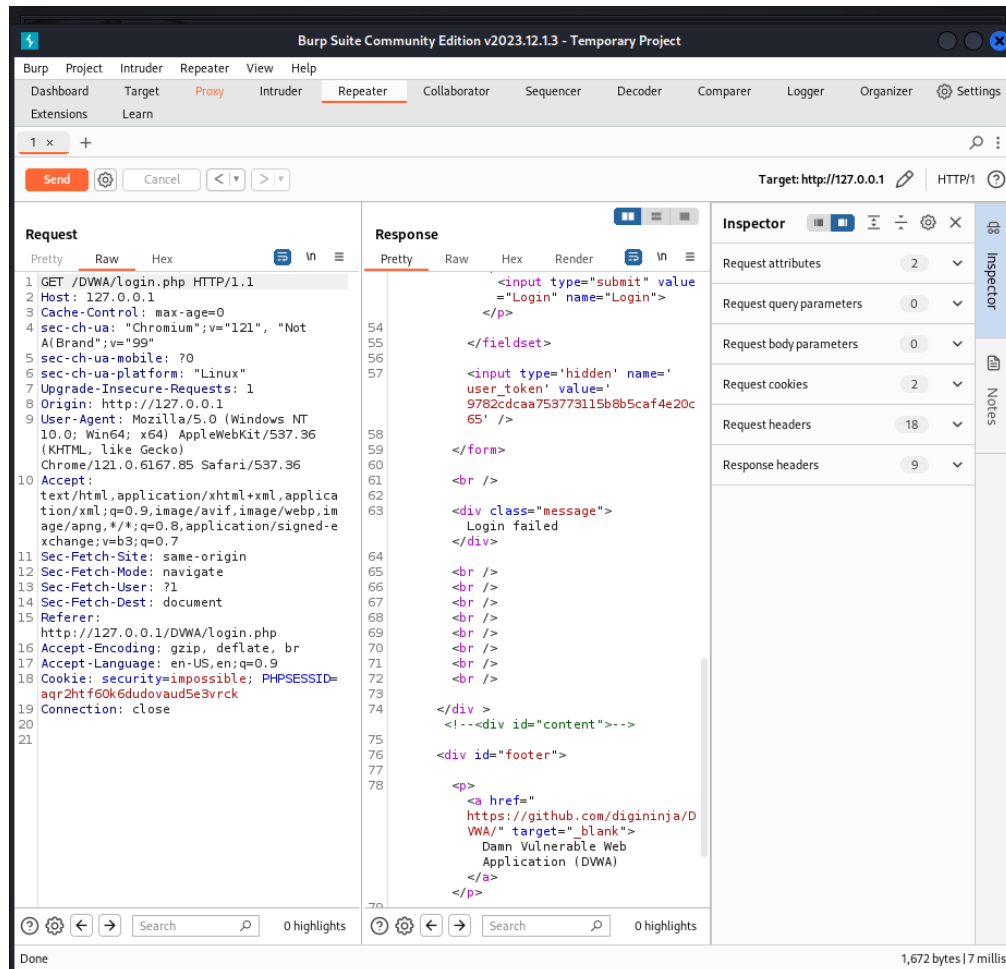
A questo punto si è fatto uso dell'applicazione **Burp Suite**, e si è scelto un progetto temporaneo. Attivando l'intercettazione della richiesta di login su Burp, si è poi effettuato l'accesso tramite browser all'indirizzo '127.0.0.1/DVWA'. Così facendo si è provato ad intercettare la nostra stessa richiesta di login fatta tramite browser.



Per verificarlo sono state cambiate username e password iniziali con delle credenziali sbagliate, proprio per verificare che fallisse il login.

```
16 Sec-Fetch-Dest: document
17 Referer: http://127.0.0.1/DWA/login.php
18 Accept-Encoding: gzip, deflate, br
19 Accept-Language: en-US,en;q=0.9
20 Cookie: security=impossible; PHPSESSID=loterc8uh9p6fcc0rjtlp04ir5
21 Connection: close
22
23 username=ciao&password=ciao&Login=Login&user_token=
    e9028deaeb00896aa38e2b80e1ba93dd
```

Come ci si aspettava, con le credenziali errate non si è riuscito ad effettuare il login. Se ne ha evidenza nel body della http response dove si legge «Login failed» a riga 63.



Team 4

Roberta - Andrea (db)- Mario (rt) – Antonio – Giammarco