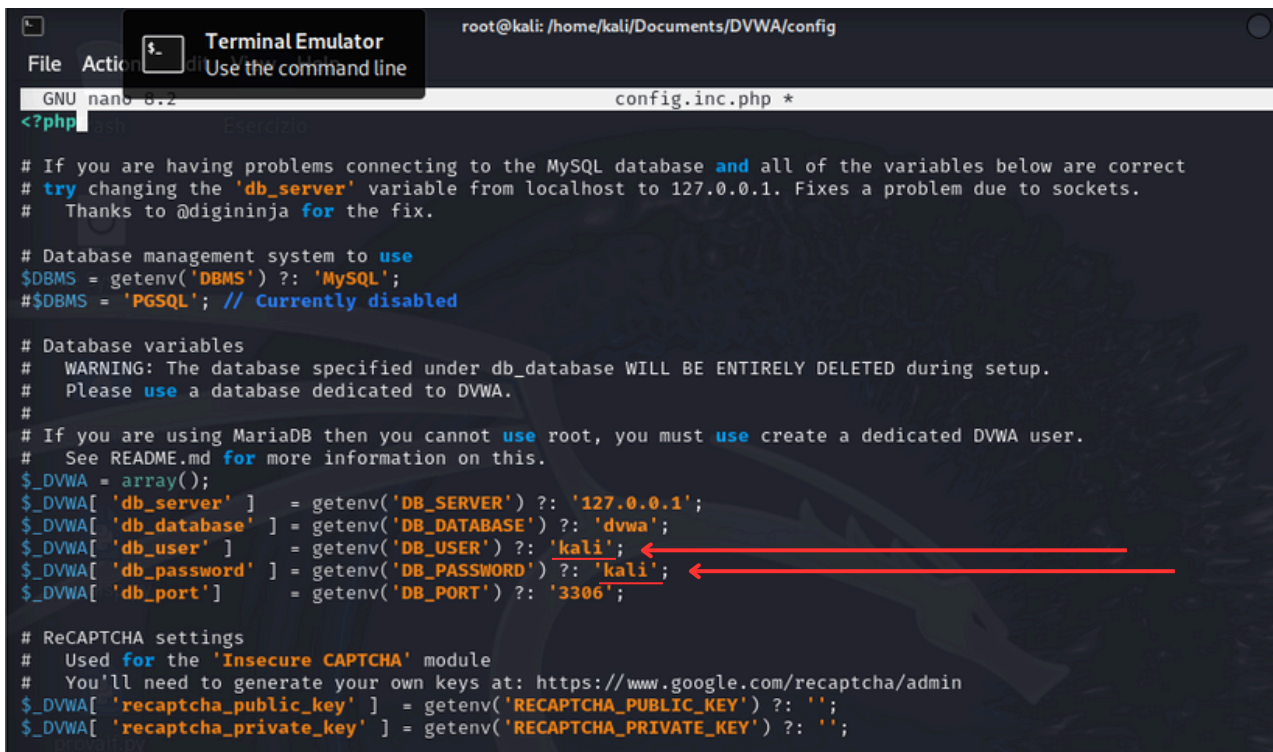


Esercizio S3/L3

L'esercizio di oggi richiedeva la configurazione di una DVWA, ovvero di una Damn Vulnerable Web Application, su Kali Linux. Per iniziare bisogna eseguire i comandi con utenza di root (**sudo su**), precedentemente proposti nelle slide, ovvero:

- `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`

Una volta eseguiti bisogna modificare il file “**config.inc.php**” e, sulle voci `db_user` e `db_password` inserire “**kali**”



```
GNU nano 0.2 config.inc.php *
<?php
# If you are having problems connecting to the MySQL database and all of the variables below are correct
# try changing the 'db_server' variable from localhost to 127.0.0.1. Fixes a problem due to sockets.
# Thanks to @digininja for the fix.

# Database management system to use
$DBMS = getenv('DBMS') ?: 'MySQL';
#$DBMS = 'PGSQL'; // Currently disabled

# Database variables
# WARNING: The database specified under db_database WILL BE ENTIRELY DELETED during setup.
# Please use a database dedicated to DVWA.
#
# If you are using MariaDB then you cannot use root, you must use create a dedicated DVWA user.
# 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' ] = getenv('DB_USER') ?: 'kali';
$_DVWA[ 'db_password' ] = getenv('DB_PASSWORD') ?: 'kali';
$_DVWA[ 'db_port' ] = getenv('DB_PORT') ?: '3306';

# ReCAPTCHA settings
# Used for the 'Insecure CAPTCHA' module
# You'll need to generate your own keys at: https://www.google.com/recaptcha/admin
$_DVWA[ 'recaptcha_public_key' ] = getenv('RECAPTCHA_PUBLIC_KEY') ?: '';
$_DVWA[ 'recaptcha_private_key' ] = getenv('RECAPTCHA_PRIVATE_KEY') ?: '';
```

Successivamente bisogna far partire il servizio mysql con il comando `mysql start` e connettersi al database con utenza di root attraverso il comando `mysql -u root -p`.

```
(root@kali)-[/home/kali/Documents/DVWA/config]
# service mysql start

(root@kali)-[/home/kali/Documents/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)]> █
```

Ora bisogna creare un utenza sul database e assegnare i privilegi, rispettivamente con i comandi:

- `create user 'kali'@'127.0.0.1' identified by 'kali' ;`
- `grant all privileges on dvwa.* to 'kali'@'127.0.0.1' identified by 'kali' ;`

E infine uscire.

```
(root@kali)-[/home/kali/Documents/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';
Query OK, 0 rows affected (0.007 sec)

MariaDB [(none)]> grant all privileges on dvwa.* to 'kali'@'127.0.0.1' identified by 'kali';
Query OK, 0 rows affected (0.004 sec)

MariaDB [(none)]> exit
Bye

(root@kali)-[/home/kali/Documents/DVWA/config]
# █
```

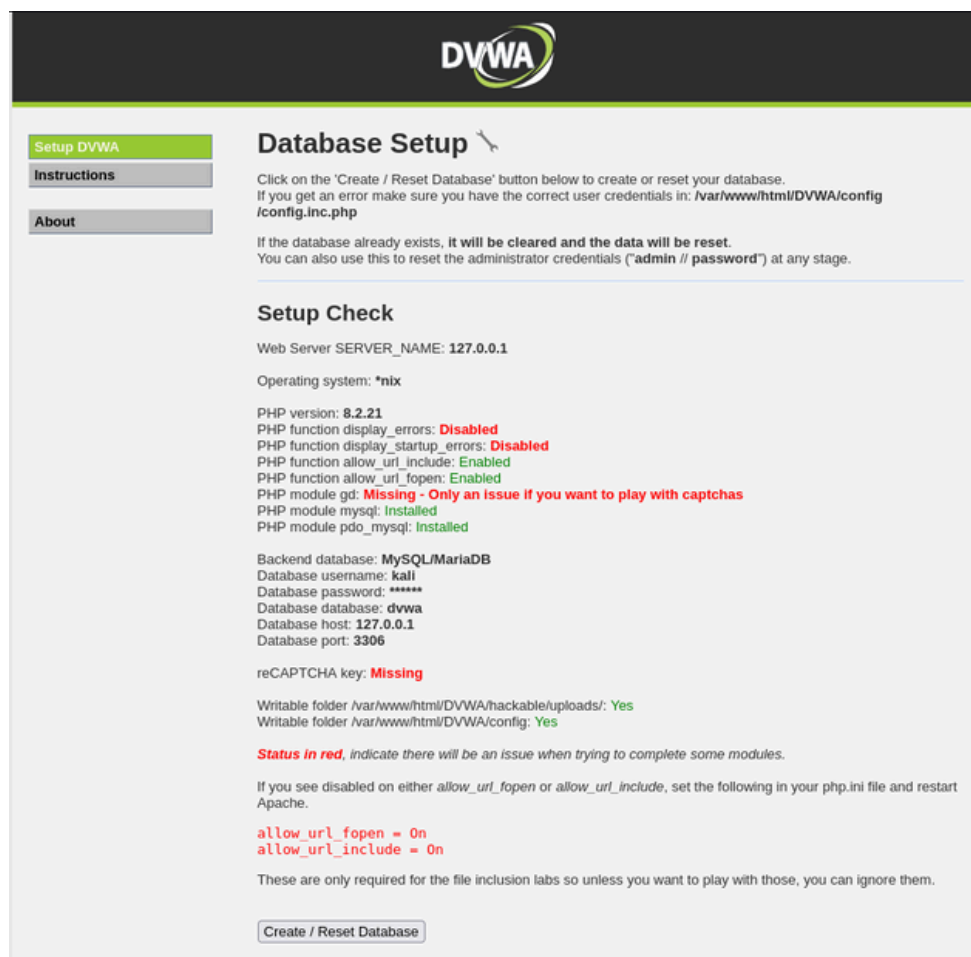
A questo punto bisogna attivare il servizio di web server apache con il comando `apache2 start`, e con il successivo comando `apache2 status` si può verificare se è attivo.

```
(kali@kali) [~]
$ service apache2 start

(kali@kali) [~]
$ service apache2 status
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; disabled; preset: disabled)
   Active: active (running) since Wed 2024-12-11 09:22:04 EST; 1h 9min ago
 Invocation: 0df9d3b82f91442abe587826e11c2978
    Docs: https://httpd.apache.org/docs/2.4/
   Process: 8100 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 8116 (apache2)
    Tasks: 11 (limit: 2269)
   Memory: 19.5M (peak: 27.8M swap: 10.4M swap peak: 15.1M)
      CPU: 639ms
   CGroup: /system.slice/apache2.service
           └─8116 /usr/sbin/apache2 -k start
             8119 /usr/sbin/apache2 -k start
             8120 /usr/sbin/apache2 -k start
             8121 /usr/sbin/apache2 -k start
             8122 /usr/sbin/apache2 -k start
             8123 /usr/sbin/apache2 -k start
           └─9835 /usr/sbin/apache2 -k start
             9844 /usr/sbin/apache2 -k start
             9845 /usr/sbin/apache2 -k start
             9846 /usr/sbin/apache2 -k start
             9847 /usr/sbin/apache2 -k start

Dec 11 09:22:03 kali systemd[1]: Starting apache2.service - The Apache HTTP Server ...
Dec 11 09:22:04 kali apachectl[8115]: AH00558: apache2: Could not reliably determine the server's full
Dec 11 09:22:04 kali systemd[1]: Started apache2.service - The Apache HTTP Server.
```

Provando poi a inserire nell'URL di ricerca del browser Firefox `127.0.0.1/DVWA/setup.php` ecco come apparirà. Clicchiamo poi sul pulsante "Create/reset Database"




Potremo accedere con utenza “admin” e “password”. Dopodichè si aprirà la Web Application

← → ↺ 🏠

🔒 📄 🔗 127.0.0.1/DVWA/login.php

☆ 📧 📌 ☰

🐧 Kali Linux 🗄️ Kali Tools 📄 Kali Docs 🐧 Kali Forums 🚩 Kali NetHunter 🔥 Exploit-DB 🔥 Google Hacking DB 🛡️ OffSec



Username


admin

Password

••••••••

Login

🐧 Kali Linux 🗄️ Kali Tools 📄 Kali Docs 🐧 Kali Forums 🚩 Kali NetHunter 🔥 Exploit-DB 🔥 Google Hacking DB 🛡️ OffSec



Home

Instructions

Setup / Reset DB

Brute Force

Command Injection

CSRF

File Inclusion

File Upload

Insecure CAPTCHA

SQL Injection

SQL Injection (Blind)

Weak Session IDs

XSS (DOM)

XSS (Reflected)

XSS (Stored)

CSP Bypass

JavaScript

Authorisation Bypass

Open HTTP Redirect

Cryptography

DVWA Security

PHP Info

About

Logout

Welcome to Damn Vulnerable Web Application!

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goal is to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and to aid both students & teachers to learn about web application security in a controlled class room environment.

The aim of DVWA is to **practice some of the most common web vulnerabilities**, with **various levels of difficulty**, with a simple straightforward interface.

General Instructions

It is up to the user how they approach DVWA. Either by working through every module at a fixed level, or selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module; however users should feel that they have successfully exploited the system as best as they possible could by using that particular vulnerability.

Please note, there are **both documented and undocumented vulnerabilities** with this software. This is intentional. You are encouraged to try and discover as many issues as possible.

There is a help button at the bottom of each page, which allows you to view hints & tips for that vulnerability. There are also additional links for further background reading, which relates to that security issue.

WARNING!

Damn Vulnerable Web Application is damn vulnerable! **Do not upload it to your hosting provider's public html folder or any Internet facing servers**, as they will be compromised. It is recommend using a virtual machine (such as [VirtualBox](#) or [VMware](#)), which is set to NAT networking mode. Inside a guest machine, you can download and install [XAMPP](#) for the web server and database.

Disclaimer

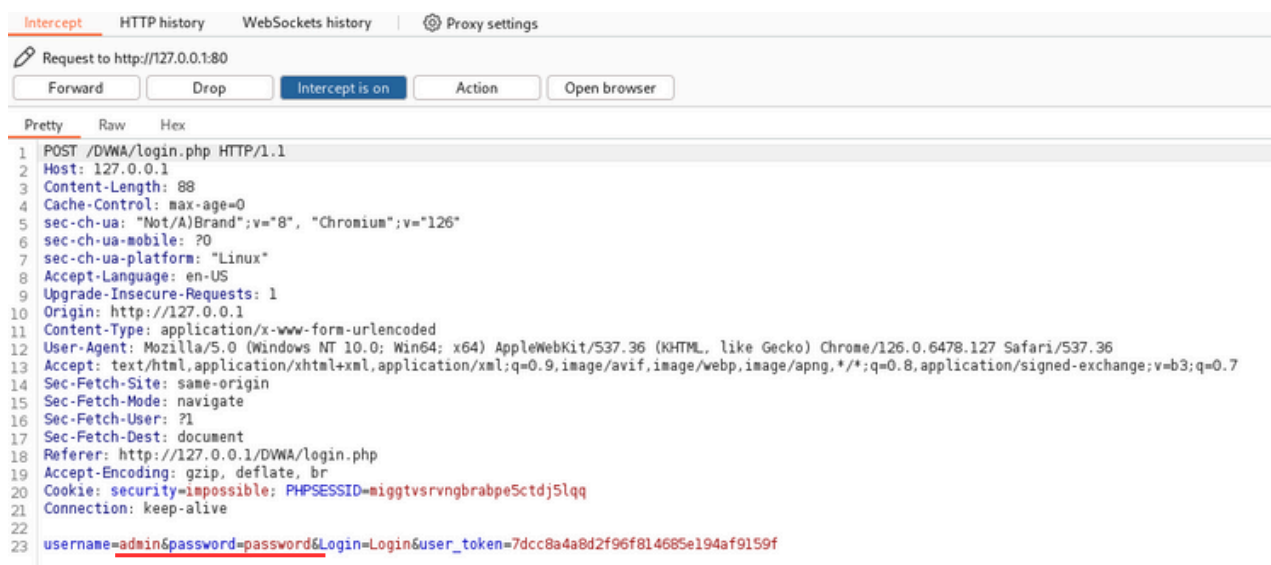
We do not take responsibility for the way in which any one uses this application (DVWA). We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

More Training Resources

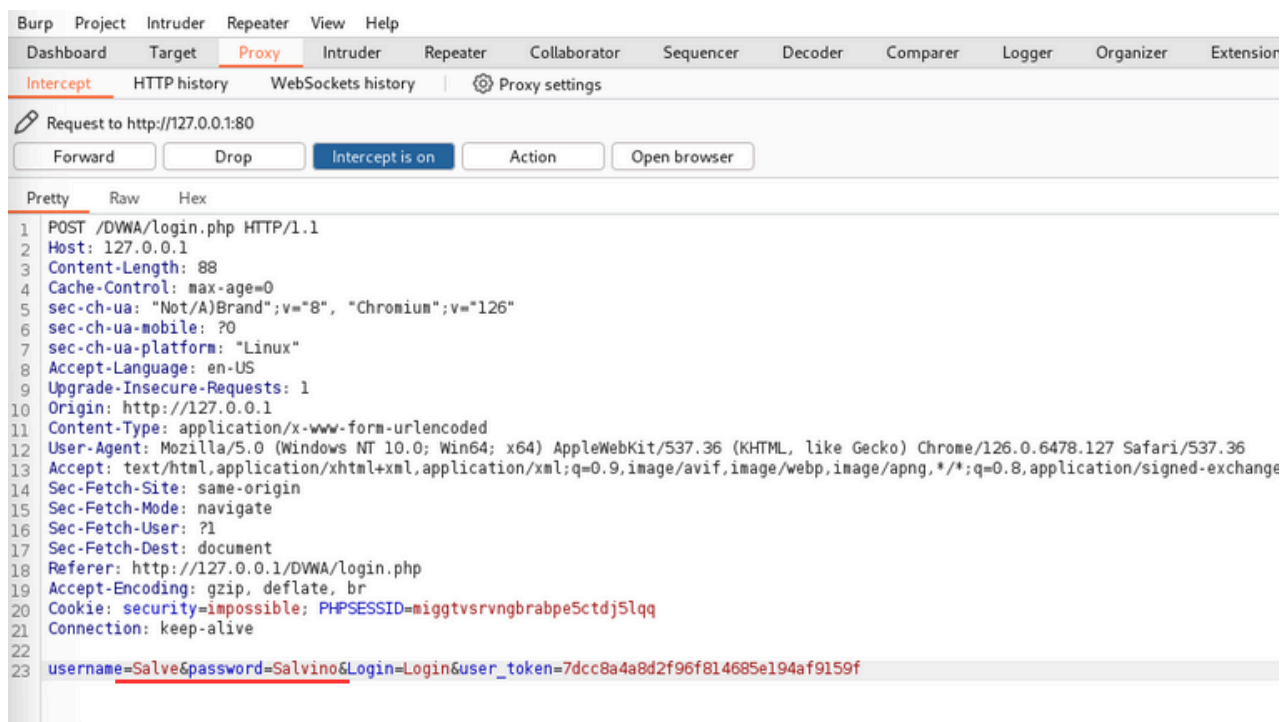
DVWA aims to cover the most commonly seen vulnerabilities found in today's web applications. However there are plenty of other issues with web applications. Should you wish to explore any additional attack vectors, or want more difficult challenges, you may wish to look into the following other projects:

- [Mutillidae](#)
- [OWASP Vulnerable Web Applications Directory](#)

Apprendo adesso Burpsuite potremo intercettare la richiesta al sito e, facendo il login, potremo osservare username e password completamente in chiaro.



Proviamo ora a fare la richiesta inversa alla Web Application ma inserendo un username e una password diversa.



Ecco come vedremo la richiesta dal programma.

The screenshot displays the Burp Suite interface with the 'Repeater' tab selected. The 'Request' pane on the left shows an HTTP 1.1 GET request to /DVWA/login.php. The 'Response' pane on the right shows the corresponding HTML response, which is an HTML form for login. The request includes various headers such as Host, Cache-Control, User-Agent, and cookies. The response is an HTML document with a form containing username and password input fields and a submit button.

Request

```
1 GET /DVWA/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/DVWA/login.php
17 Accept-Encoding: gzip, deflate, br
18 Cookie: security=impossible; PHPSESSID=niggtvsrvngbrabpe5ctdj5lqq
19 Connection: keep-alive
20
21
```

Response

```
36 <!-->
37 <br />
38
39 </div>
40 <!--<div id="header">-->
41
42 <div id="content">
43
44 <form action="login.php" method="post">
45
46 <fieldset>
47
48 <label for="user">
49 Username
50 </label>
51 <input type="text" class="loginInput" size="20" name="username">
52 <br />
53
54 <label for="pass">
55 Password
56 </label>
57 <input type="password" class="loginInput" AUTOCOMPLETE="off" size="20"
    name="password">
58 <br />
59
60 <br />
61 <p class="submit">
62 <input type="submit" value="Login" name="Login">
63 </p>
64
65 </fieldset>
66
67 <input type="hidden" name="user_token" value="
    2665d65669d24ee072b421802147e0a6" />
68
69 </form>
70
71
```