

REPORT S3/L3

Traccia: Nella lezione pratica di oggi vedremo come configurare una DVWA – ovvero damn vulnerable web application in Kali Linux. La DVWA ci sarà molto utile per i nostri test.

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

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

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

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

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

Andiamo ad editare i file config.inc.php, inserendo “kali” “kali” come “user” e “password”.

```
$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';
```

Sempre con utenza di root su Kali, facciamo partire il servizio mysql con il comando: **service mysql start** poi connettiamoci al db con utenza di root con il comando seguente (ricordatevi che la password standard per utente di root è «kali»): **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' ;
Query OK, 0 rows affected (0.003 sec)

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
```

Creiamo un'utenza sul db con il 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”**.

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

(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)]> |
```

Avviamo il servizio apache con il comando **“service apache2 start”** e editiamo il file php.ini.

Infine dopo aver eseguito i vari passaggi inserire nel browser l'indirizzo 127.0.0.1/DVWA

Setup Check

Web Server SERVER_NAME: 127.0.0.1

Operating system: *nix

PHP version: 8.2.21

PHP function display_errors: Disabled

PHP function display_startup_errors: Disabled

PHP function allow_url_include: Disabled

PHP function allow_url_fopen: Enabled

PHP module gd: Missing - Only an issue if you want to play with captchas

PHP module mysql: Installed

PHP module pdo_mysql: Installed

Backend database: MySQL/MariaDB

Database username: kali

Database password: *****

Database database: dvwa

Database host: 127.0.0.1

Database port: 3306

reCAPTCHA key: Missing

Writable folder /var/www/html/DVWA/hackable/uploads/: Yes

Writable folder /var/www/html/DVWA/config: Yes

Status in red, indicate there will be an issue when trying to complete some modules.

If you see disabled on either `allow_url_fopen` or `allow_url_include`, set the following in your php.ini file and restart Apache.

`allow_url_fopen = On`

`allow_url_include = On`

These are only required for the file inclusion labs so unless you want to play with those, you can ignore them.

Create / Reset Database

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](#)
- [QWASP Vulnerable Web Applications Directory](#)

Avviare ora **BURPSUITE** andando a intercettare la richiesta di login facendo poi varie prove. Andando a modificare i dati in rosso si possono notare i differenti esiti.

Project

Intruder

Repeater

View

Help

Dashboard

Target

Proxy

Intruder

Repeater

Collaborator

Sequencer

Decoder

Comparer

Logger

Organizer

Extensions

Learn

Intercept

HTTP history

WebSockets history

Match and replace

Proxy settings

On Intercept on

Forward

Drop

Request to http://127.0.0.1:80

Time	Type	Direction	Method	URL
15:52:41.11 D...	HTTP	→ Request	POST	http://127.0.0.1/DVWA/login.php

Request

Inspector

Pretty

Raw

Hex

Content-Type: application/x-www-form-urlencoded

Upgrade-Insecure-Requests: 1

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36

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

Sec-Fetch-Site: same-origin

Sec-Fetch-Mode: navigate

Sec-Fetch-User: 11

Sec-Fetch-Dest: document

Referer: https://127.0.0.1/DVWA/login.php

Accept-Encoding: gzip, deflate, br

Cookie: securityimpossible; PHPSESSID=4ba1updd21vab9np23f vk 7h70a

Connection: keep-alive

username=admin&password=password>Login=Login&user_token=ecdf39bbf2463db96a518d43ca3ad7be

Inspector

Request attributes

Request query paramet

Request body paramet

Request cookies

Request headers

Event log (1)

All issues