

Inizio con l'installazione della DVWA (Damn Vulnerable Web Application)

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

(root@kali)-[/var/www/html]
└─# git clone https://github.com/digininja/DVWA
Cloning into 'DVWA' ...
remote: Enumerating objects: 4590, done.
remote: Counting objects: 100% (140/140), done.
remote: Compressing objects: 100% (103/103), done.
remote: Total 4590 (delta 58), reused 101 (delta 36), pack-reused 4450
Receiving objects: 100% (4590/4590), 2.34 MiB | 7.34 MiB/s, done.
Resolving deltas: 100% (2153/2153), done.

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

(root@kali)-[/var/www/html]
└─# cd DVWA/CONFIG
cd: no such file or directory: DVWA/CONFIG

(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]
└─# 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: 10.11.8-MariaDB-1 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)]>
```

```
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 (0.009 sec)

MariaDB [(none)]> grant all privileges on dvwa.* to 'kali'@'127.0.0.1' identified by 'kali';
Query OK, 0 rows affected (0.005 sec)

MariaDB [(none)]> exit
Bye

(root@kali)-[/var/www/html/DVWA/config]
└─# server apache2 start
Command 'server' not found, did you mean:
  command 'cserver' from deb freewnn-cserver
  command 'jserver' from deb freewnn-jserver
  command 'semver' from deb node-semver
  command 'kserver' from deb freewnn-kserver
Try: apt install <deb name>

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

(root@kali)-[/var/www/html/DVWA/config]
└─# cd /etc/php/8.1/apache2
cd: no such file or directory: /etc/php/8.1/apache2

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

(root@kali)-[/etc/php]
└─# ls
8.2

(root@kali)-[/etc/php]
└─# cd /etc/php/8.2/apache2

(root@kali)-[/etc/php/8.2/apache2]
└─# nano php.ini

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

(root@kali)-[/etc/php/8.2/apache2]
└─#
```

```
;;;;;;;;;;  
  
; 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
```

Setup DVWA

Instructions

About

Database Setup

Click on the 'Create / Reset Database' button below to create or reset your database.
If you get an error make sure you have the correct user credentials in: `/var/www/html/DVWA/config/config.inc.php`

If the database already exists, **it will be cleared and the data will be reset**.
You can also use this to reset the administrator credentials ("**admin** // **password**") at any stage.

Setup Check

Web Server SERVER_NAME: **127.0.0.1**

Operating system: ***nix**

PHP version: **8.2.18**
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**

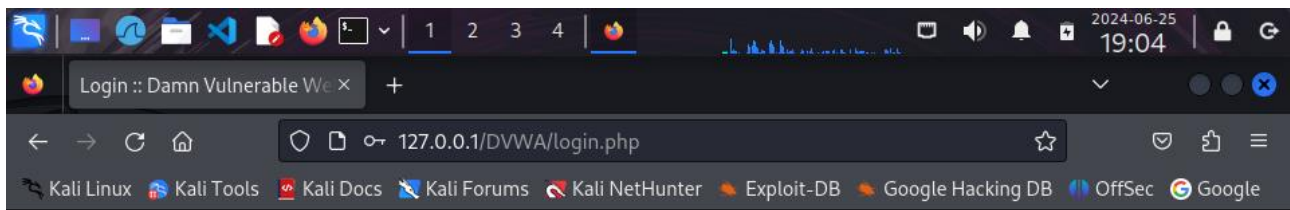
reCAPTCHA key: **Missing**

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
```



Username

kali

Password

••••

Login

[Damn Vulnerable Web Application \(DVWA\)](#)

127.0.0.1/DVWA/index.php


Kali LinuxKali ToolsKali DocsKali ForumsKali NetHunterExploit-DBGoogle Hacking DBOffSecGoogle

HomeInstructionsSetup / Reset DB

Brute ForceCommand InjectionCSRFFile InclusionFile UploadInsecure CAPTCHASQL InjectionSQL Injection (Blind)Weak Session IDsXSS (DOM)XSS (Reflected)XSS (Stored)CSP BypassJavaScriptAuthorisation BypassOpen HTTP Redirect

DVWA SecurityPHP InfoAbout

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

1

2

3

4

127.0.0.1/DVWA/security.php

Kali LinuxKali ToolsKali DocsKali ForumsKali NetHunterExploit-DBGoogle Hacking DBOffSecGoogle

DVWA

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

DVWA Security

PHP Info

About

Logout

DVWA Security

Security Level

Security level is currently: impossible.

You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

1. Low - This security level is completely vulnerable and **has no security measures at all**. It's use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.

2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.

3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.

4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code.

Prior to DVWA v1.9, this level was known as 'high'.

Low

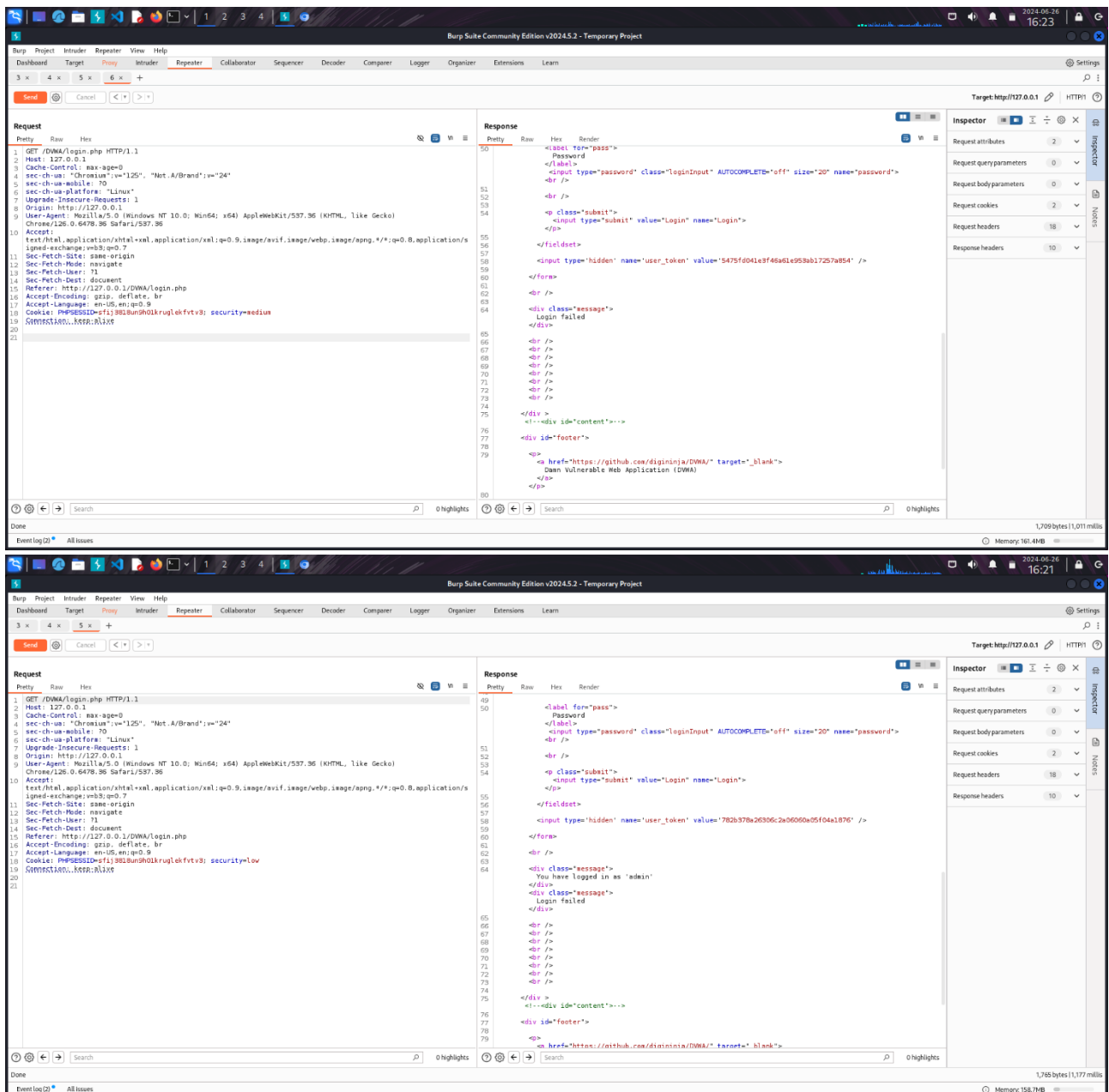
Submit

Dopo la configurazione della DVWA procedo con l'installazione di Burpsuite e inizio ad intercettare le richieste del sito web <http://127.0.0.1/DVWA/login.php> come specificato nell'esercizio.

The image displays two screenshots of the Burp Suite Community Edition v2024.5.2 interface, showing the HTTP history and response for a login attempt on the DVWA (Damn Vulnerable Web Application).

Top Screenshot: Shows the initial login request and response. The request is a GET to `/DVWA/login.php` with various headers including `Host: 127.0.0.1`, `Cache-Control: max-age=0`, `sec-ch-ua: "Chromium";v="125", "Not A/Brand";v="24"`, `sec-ch-ua-mobile: ?0`, `sec-ch-ua-platform: "Linux"`, `Upgrade-Insecure-Requests: 1`, `Origin: http://127.0.0.1`, `User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.96 Safari/537.36`, and `Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7`. The response is an HTML page with a login form containing fields for username and password, and a submit button labeled "Login".

Bottom Screenshot: Shows the response after a login attempt. The response is an HTML page with a message indicating "Login failed". The response headers include `Content-Type: text/html; charset=UTF-8`, `Server: Apache/2.4.18 (Ubuntu)`, and `Set-Cookie: PHPSESSID=01bfa2b117ab040b34359e0`. The response body contains a message "Login failed" and a link to the DVWA GitHub repository.



Nell'analisi dei 4 response HTTP provenienti dalla DVWA, possiamo notare come le misure di sicurezza e le configurazioni variano significativamente tra i livelli "low", "medium", "high", e "impossible".

Tutti i response condividono alcune caratteristiche base: Headers http, per esempio, include direttive come no-cache, must-revalidate che indicano una politica di non memorizzazione nella cache per prevenire la conservazione di dati sensibili, o i Token CSRF, che sono presenti in tutti i livelli, dimostrano l'importanza della protezione contro attacchi CSRF attraverso la verifica dell'autenticità delle richieste.

Livello Low: La presenza del token CSRF indica una protezione di base contro gli attacchi CSRF, ma al livello low, i token potrebbero non essere gestiti o controllati con rigore.

Il messaggio di errore dato dal sistema è "You have logged in as 'admin'" seguito da "Login failed". Questo mostra una gestione blanda dei messaggi di stato che può confondere l'utente o fornire un feedback inappropriato, indicativo di misure di sicurezza più deboli e meno sofisticate.

Livello Medium: La sicurezza sarà simile al livello low, ma con possibili miglioramenti nella verifica dei token. Il messaggio "Login failed" persiste, indicando che i tentativi di login non autorizzati sono bloccati, ma senza dettagli aggiuntivi.

Livello High: A questo livello i token CSRF sono gestiti con controlli molto più stringenti. Il messaggio "Login failed" indica che il sistema è configurato per respingere gli accessi non autorizzati in modo efficace e che potrebbero esserci ulteriori misure di protezione non evidenti solo dai messaggi di errore. Il server respinge i tentativi di accesso senza fornire dettagli o feedback che potrebbero aiutare un attaccante.

Livello Impossible: Adesso si presume che tutti i controlli di sicurezza siano ottimizzati al massimo. La presenza del token CSRF, che è ancora diverso e unico rispetto ai precedenti response, indica un controllo rigoroso e probabilmente una verifica crittografica per assicurarsi che la richiesta provenga dalla sessione corretta e non sia manipolata. L'assenza completa di messaggi di errore o di successo nel response suggerisce un livello di sicurezza dove non si forniscono indicazioni che potrebbero essere sfruttate per scoprire vulnerabilità nel sistema.

In conclusione, si nota che la progressione dal livello low al livello impossible presenta un incremento significativo nella gestione della sicurezza, evidente sia nella gestione dei token CSRF, sia nei messaggi di risposta del server. Ogni livello incrementa sicuramente la rigidità dei controlli e riduce le informazioni disponibili che potrebbero essere utilizzate per attacchi. Questa analisi va a dimostrare l'efficacia crescente delle misure di sicurezza implementate in DVWA man mano che si aumenta il livello di difficoltà.