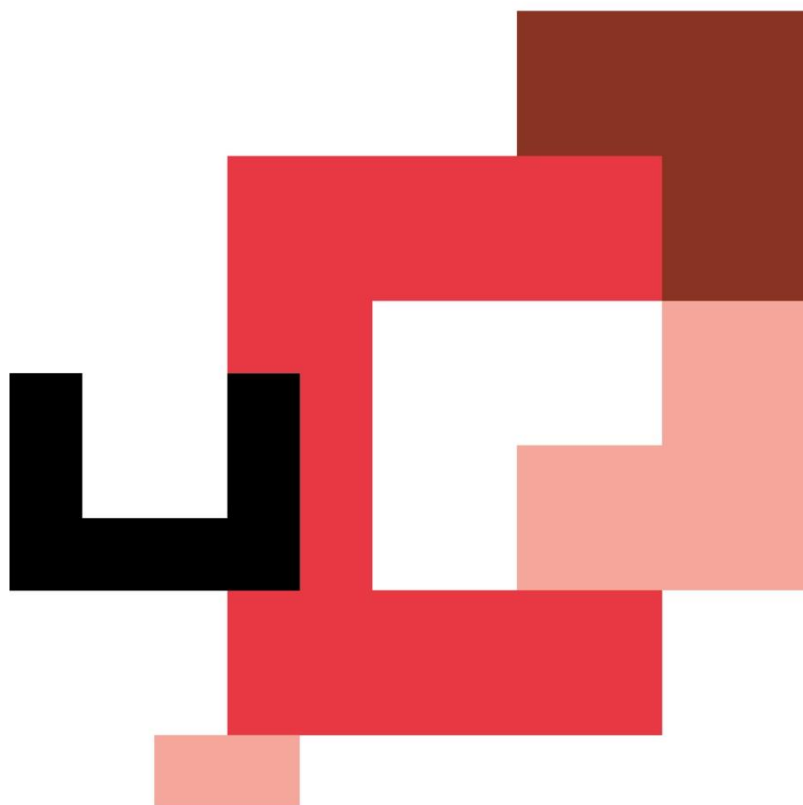




BOOTCAMP

Ciberseguridad en formato online



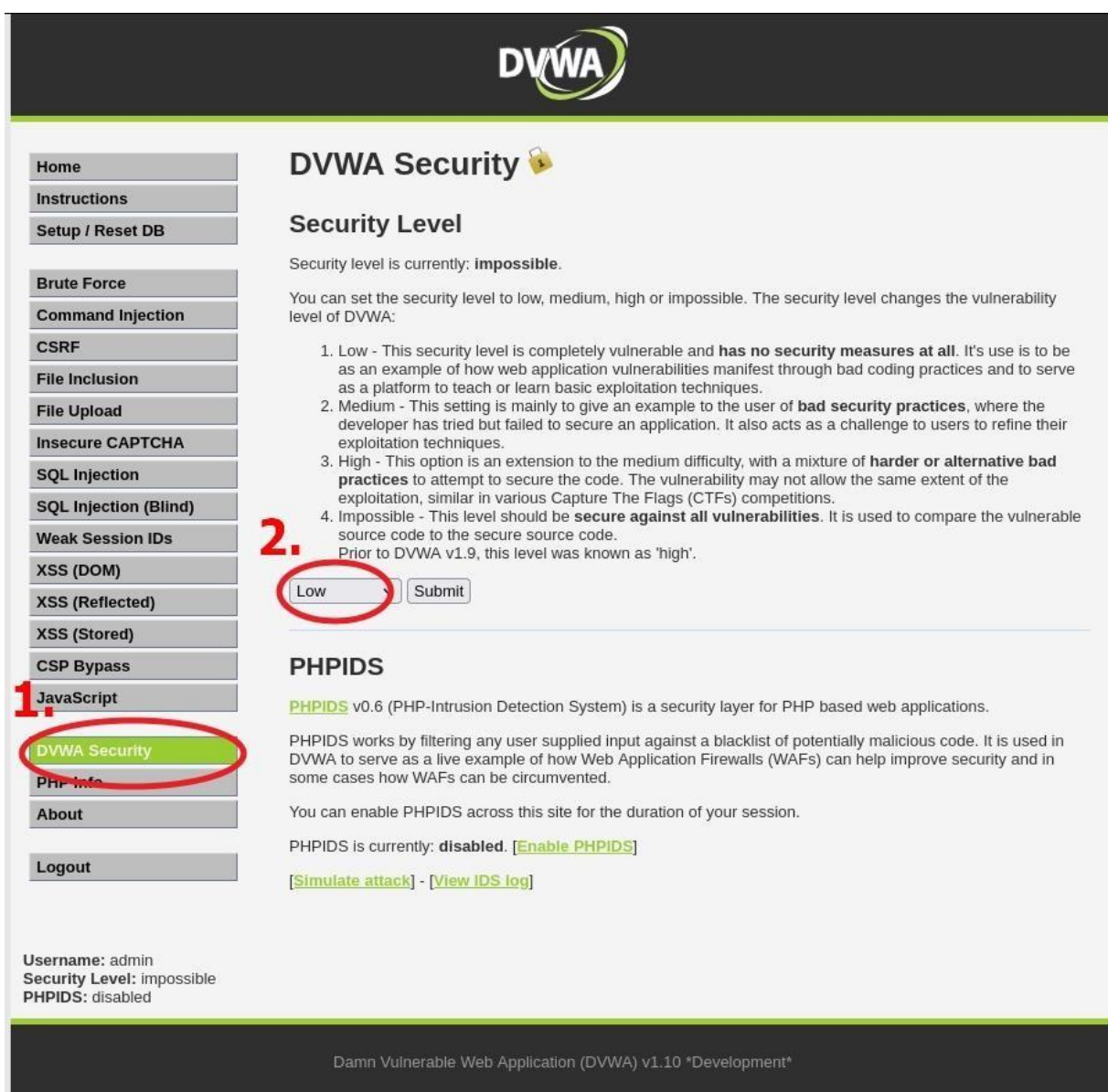
EJERCICIOS SQL Injection

Prerrequisitos


Para esta actividad lo primero que vamos a necesitar es colocar la máquina **vm** y la **Kali** en la misma red, para ello vamos a colocarlas en Red Nat.

Una vez tengamos establecida la configuración de red debemos conectarnos a través de nuestra **Kali** al recurso levantado por la máquina **vm** en el puerto 80.

Una vez accedido al recurso nos dirigimos a cambiar el nivel de la máquina y lo establecemos en **Low**



The screenshot shows the DVWA Security page. On the left sidebar, the 'DVWA Security' menu item is highlighted with a red circle and labeled '1.'. In the main content area, the 'Security Level' section shows the current level as 'Impossible'. A list of four security levels is provided, with the second item, 'Medium', circled in red and labeled '2.'. Below the list, the 'Low' option in the dropdown menu is also circled in red. The 'Submit' button is visible next to the dropdown. At the bottom of the page, the status bar shows 'Username: admin', 'Security Level: impossible', and 'PHPIDS: disabled'.

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

PHPIDS

PHPIDS v0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

PHPIDS works by filtering any user supplied input against a blacklist of potentially malicious code. It is used in DVWA to serve as a live example of how Web Application Firewalls (WAFs) can help improve security and in some cases how WAFs can be circumvented.

You can enable PHPIDS across this site for the duration of your session.


PHPIDS is currently: **disabled**. [\[Enable PHPIDS\]](#)

[\[Simulate attack\]](#) - [\[View IDS log\]](#)

Username: admin
Security Level: impossible
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.10 *Development*

Después nos dirigimos al recurso de **XSS (Reflected)**



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

DVWA Security

PHP Info

About

Logout

Username: admin

Security Level: impossible

PHPIDS: disabled

Vulnerability: SQL Injection

User ID:

More Information

- <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
- https://en.wikipedia.org/wiki/SQL_injection
- <http://feruh.mavituna.com/sql-injection-cheatsheet-oku/>
- <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>
- https://www.owasp.org/index.php/SQL_injection
- <http://bobby-tables.com/>

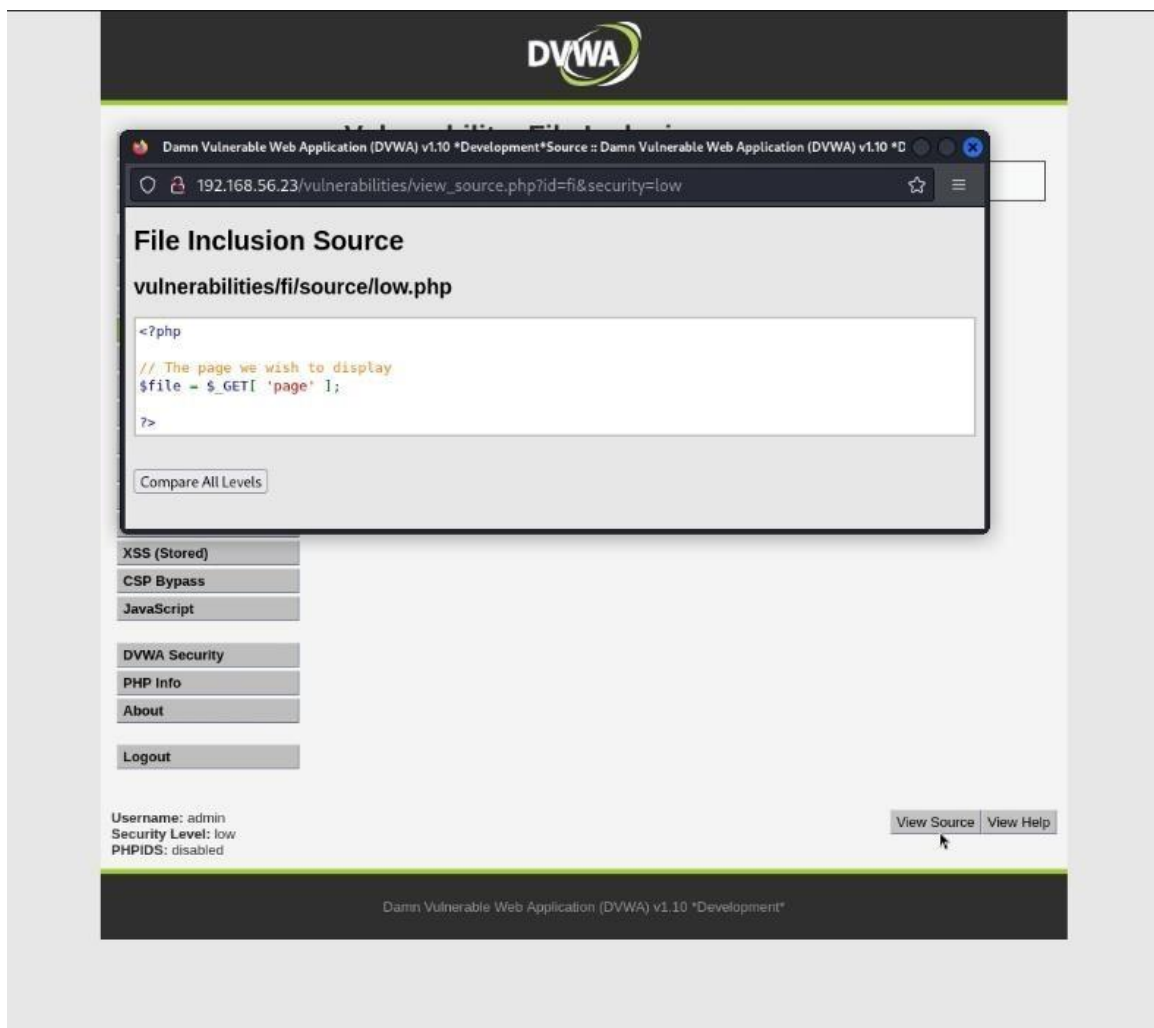
Damn Vulnerable Web Application (DVWA) v1.10 "Development"

Y ya estaríamos listos para comenzar el challenge.

Ejercicio

Para esta actividad debes realizar un breve documento explicando la vulnerabilidad presentada en el nivel **Low y Medium**, explicando cómo es que se produce la vulnerabilidad y las recomendaciones pertinentes para solucionar la vulnerabilidad.

Como pista recordarte que puedes ver el código que hay por detrás de la aplicación y así analizar la vulnerabilidad en particular pulsando, como se muestra en la imagen, en el botón **View Source**.



El informe a presentar debe constar; a parte de la explicación y subsanación de la vulnerabilidad, de aportaciones visuales que permitan revelar la explotación de dicha vulnerabilidad.



**THE
BRIDGE**

