Revista Digital del Departamento de
Ingeniería e Investigaciones Tecnológicas
de la Universidad Nacional de La Matanza
ISSN: 2525-1333
Vol.:10-Nro.1 (julio-2025)

Sección de alumnos: artículo original

Detección y explotación de vulnerabilidades de tipo SQL Injection: Union Select

SQL Injection vulnerability detection and exploitation: Union Select

Integrantes:

Agustín Federico
Ingeniería en Informática, Universidad Nacional de La Matanza
afederico@alumno.unlam.edu.ar

Alejo Agasi
Ingeniería en Informática, Universidad Nacional de La Matanza
aagasi@alumno.unlam.edu.ar

Referentes institucionales:

Martín Zeballos
Universidad Nacional de la Matanza
mzeballos@unlam.edu.ar

Maximiliano Alejandro Downar
Universidad Nacional de la Matanza
adownar@unlam.edu.ar

Aníbal Pose
Universidad Nacional de la Matanza
apose@unlam.edu.ar

Revista Digital del Departamento de
Ingeniería e Investigaciones Tecnológicas
de la Universidad Nacional de La Matanza
ISSN: 2525-1333
Vol.:10-Nro.1 (julio-2025)

Resumen:

La inyección SQL (SQLi) es una de las vulnerabilidades más críticas y frecuentes en aplicaciones web. Este trabajo analiza detalladamente su funcionamiento, impacto y formas de explotación, a partir de la realización práctica de un laboratorio del sitio PortSwigger. En él, se logró recuperar credenciales de acceso mediante una técnica de inyección basada en la cláusula UNION SELECT, demostrando el compromiso de la confidencialidad e integridad de los datos. Se analizó también la severidad bajo CVSS v3, obteniendo un puntaje de 9.1. Finalmente, se presentan recomendaciones para mitigar este tipo de fallas a través de buenas prácticas de desarrollo seguro.

Abstract:

SQL Injection (SQLi) is one of the most critical and frequent vulnerabilities in web applications. This paper presents a detailed analysis of its functioning, impact, and exploitation techniques through a hands-on lab from PortSwigger. In this scenario, we successfully retrieved login credentials using a UNION SELECT-based injection, proving how confidentiality and integrity can be compromised. The severity was evaluated using the CVSS v3 metric, resulting in a score of 9.1. Mitigation recommendations are provided, highlighting secure coding practices and input validation mechanisms.

Palabras Clave: *Vulnerabilidad, inyección SQL, PortSwigger, CVSS, UNION*

Key Words: *Vulnerability, SQL injection, PortSwigger, CVSS, UNION*

Revista Digital del Departamento de
Ingeniería e Investigaciones Tecnológicas
de la Universidad Nacional de La Matanza
ISSN: 2525-1333
Vol.:10-Nro.1 (julio-2025)

I. CONTEXTO

Este trabajo se inscribe en el área de conocimiento de la seguridad informática, específicamente en el estudio y detección de vulnerabilidades en aplicaciones web. El proyecto fue desarrollado en el marco de la asignatura Tecnologías en Seguridad, dentro de la carrera Ingeniería en Informática de la Universidad Nacional de La Matanza. El estudio tiene como finalidad comprender a fondo la naturaleza de la vulnerabilidad SQLi y aplicar herramientas prácticas para su detección y mitigación.

II. INTRODUCCIÓN

La inyección SQL (SQLi) es una técnica de ataque que explota fallas en la forma en que las aplicaciones manejan las entradas de los usuarios en las consultas SQL. Esta falla se encuentra listada como una de las principales amenazas en OWASP Top 10 [1]. Mediante la inserción de código malicioso, un atacante puede manipular consultas, acceder a datos sensibles e incluso comprometer sistemas completos. Existen múltiples variantes como las basadas en errores, tiempo, booleanas o canales secundarios [2].

En este trabajo se analiza específicamente un caso de tipo Union-based SQLi, donde se usa la cláusula UNION SELECT para recuperar datos de otras tablas. La explotación se realizó sobre un entorno controlado brindado por PortSwigger Web Security Academy [3]. El análisis incluyó la evaluación del riesgo con el vector CVSS v3.1 [4], la exposición paso a paso de la explotación, y la presentación de medidas preventivas recomendadas por la OWASP Cheat Sheet Series [5].

III. MÉTODOS

El estudio se basó en la realización del laboratorio “SQL injection UNION attack, retrieving multiple values in a single column”, disponible en la plataforma PortSwigger Web Security Academy [3]. La metodología se dividió en dos partes: el reconocimiento del sistema vulnerable y la explotación efectiva de la falla.

A. Entorno de prueba:

Se accedió al laboratorio desde el siguiente enlace:

https://portswigger.net/web-security/sql-injection/union-attacks/lab-retrieve-multiple-values-in-single-column

El entorno es un sitio web con una funcionalidad de filtrado por categoría de productos.

Revista Digital del Departamento de
Ingeniería e Investigaciones Tecnológicas
de la Universidad Nacional de La Matanza
ISSN: 2525-1333
Vol.:10-Nro.1 (julio-2025)

B. Explotación práctica:

El procedimiento de explotación fue el siguiente:

1. *Acceso al sitio*: Click en "ACCESS THE LAB". Luego, navegación a una categoría (por ejemplo, *Gifts*).

2. *Identificación del parámetro vulnerable*: Se observó que la URL incluía el parámetro category, como en: `https://<site>.web-security-academy.net/filter?category=Gifts`

3. *Detección del número de columnas*: Se probó con distintas combinaciones de NULL mediante inyecciones como:

    - Gifts' UNION SELECT NULL--
    - Gifts' UNION SELECT NULL,NULL--
    - Gifts' UNION SELECT NULL,NULL,NULL--

    Hasta que el sitio cargó sin errores, indicando coincidencia con el número correcto de columnas.

4. *Identificación de columna con soporte de texto*: Se reemplazó cada NULL por 'test' para verificar qué columna permitía mostrar texto visible en el sitio.

5. *Recuperación de datos sensibles:* Se utilizó la siguiente inyección para concatenar username y password: Gifts' UNION SELECT NULL, username || '~' || password FROM users-- Esto permitió visualizar credenciales en pantalla, como: administrator~s3cr3tpass

6. *Acceso con privilegios elevados:* Con esas credenciales, se ingresó a la sección "My account" como administrator.

Este procedimiento permitió comprobar de forma práctica la existencia y explotación exitosa de una vulnerabilidad crítica de tipo **Union-based SQLi**.

Revista Digital del Departamento de
Ingeniería e Investigaciones Tecnológicas
de la Universidad Nacional de La Matanza
ISSN: 2525-1333
Vol.:10-Nro.1 (julio-2025)

IV. RESULTADOS Y OBJETIVOS

Resultados alcanzados:

- Se obtuvo acceso no autorizado como administrador mediante SQLi.
- Se evidenció la falta de sanitización en la entrada del parámetro category.
- Se determinó que el sistema era vulnerable a una inyección UNION SELECT.
- Se estableció una puntuación CVSS de 9.1, categoría **Crítica.**
- Se confirmó la posibilidad de comprometer la confidencialidad e integridad sin afectar la disponibilidad del sistema.

Objetivos en curso o futuros:

- Incorporar pruebas automatizadas para detección temprana de SQLi.
- Aplicar contramedidas como consultas preparadas (plantillas para consultas a sistemas de bases de datos en lenguaje SQL, cuyos parámetros están desprovistos de valores) y ORMs (técnica de programación que facilita la interacción entre aplicaciones orientadas a objetos y bases de datos relacionales)
- Evaluar la misma técnica en otros entornos, para comparar comportamientos.

V. CONCLUSIONES

La inyección SQL continúa siendo una de las amenazas más significativas dentro del panorama de la seguridad en aplicaciones web. Su persistencia obedece, en gran parte, a errores comunes en la validación de entradas, al uso inadecuado de concatenación de strings en consultas SQL, y a la falta de implementación de mecanismos de defensa adecuados por parte de los desarrolladores.

El presente trabajo permitió evidenciar, a través de un entorno controlado, cómo una vulnerabilidad de tipo Union-based SQL Injection puede ser explotada sin privilegios previos y sin interacción del usuario. A partir del uso de la cláusula UNION SELECT, fue posible enumerar columnas, identificar tipos de datos aceptados y finalmente acceder a información sensible almacenada en la base de datos, como credenciales de administrador. El éxito de esta explotación remarca la criticidad de este tipo de vulnerabilidad, la cual fue correctamente clasificada con una severidad crítica (CVSS: 9.1), comprometiendo la confidencialidad e integridad de la aplicación.

Desde el punto de vista metodológico, el enfoque práctico utilizado permitió comprender en profundidad tanto el impacto técnico como la lógica subyacente al ataque. Asimismo, este tipo de ejercicios resulta fundamental para formar profesionales con pensamiento ofensivo-defensivo, capaces de identificar fallas y aplicar medidas de mitigación proactivas.

Entre las contramedidas analizadas, se destacan las consultas preparadas, que permiten separar el código SQL de los datos ingresados por los usuarios, y el uso de ORMs, que abstraen la interacción con la base de datos mediante estructuras seguras. Estas prácticas, junto con una validación robusta del lado del servidor, monitoreo de errores y políticas de mínimo privilegio, constituyen pilares fundamentales para mitigar los riesgos asociados a SQLi.

En definitiva, este estudio demuestra no solo la peligrosidad de las inyecciones SQL, sino también la necesidad urgente de adoptar buenas prácticas de desarrollo seguro, desde las etapas iniciales del ciclo de vida del software.

VI. REFERENCIAS Y BIBLIOGRAFÍA

A. Referencias bibliográficas:

[1] OWASP Foundation, "SQL Injection," [en línea]. Disponible en: https://owasp.org/www-community/attacks/SQL_Injection. [Consulta: julio 2025].

[2] DataCamp, "SQL Injection: Qué es y cómo prevenirla," [en línea]. Disponible en: https://www.datacamp.com/es/tutorial/sql-injection. [Consulta: julio 2025].

[3] PortSwigger Web Security Academy, "SQL Injection," [en línea]. Disponible en: https://portswigger.net/web-security/sql-injection. [Consulta: julio 2025].

[4] FIRST, "Common Vulnerability Scoring System (CVSS)," HackerOne, [en línea]. Disponible en: https://www.first.org/cvss/. [Consulta: julio 2025].

[5] OWASP Cheat Sheet Series, "SQL Injection Prevention Cheat Sheet," [en línea]. Disponible en: https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html. [Consulta: julio 2025].

B. Bibliografía:

P. Documentation. *PREPARE*. PostgreSQL Global Development Group, versión actual. [en línea] Disponible en: https://www.postgresql.org/docs/current/sql-prepare.html. [Consulta: julio 2025].

Recibido: 2025-06-30
Aprobado: 2025-07-18
Hipervínculo Permanente: https://doi.org/10.54789/reddi.10.1.4
Datos de edición: Vol. 10 - Nro. 1 – S.A. Art. 1
Fecha de edición: 2025-07-31