**Departamento de Ciencias de la Computación**

**Carrera de Ingeniería de Software**

**Ingeniería de Seguridad del Software.**

**Práctica de Laboratorio No. 4:**

**Tema: Exploración de Vulnerabilidades SQL Injection con SQLmap.**

**Objetivo:** Explorar las Vulnerabilidades SQL Injection (SQLi) utilizando SQLmap en un entorno virtual de red controlado:

**Requerimientos:**

* **Ubuntu Desktop** – Cliente con navegador para interacción manual (opcional).
* **Ubuntu Server** – Servidor web con aplicación vulnerable (p. ej., DVWA o Mutillidae) y base de datos (MySQL/MariaDB).
* **Kali Linux** – Máquina atacante con SQLmap y herramientas ofensivas.
* **VirtualBox** o cualquier otra herramienta de virtualización para gestionar las máquinas virtuales.
* **Acceso a Internet** para descarga de herramientas y recursos.
* **OWASP top ten attacks:** una lista de los 10 principales ataques de ciberseguridad a las aplicaciones web y una guía metodológica que proporciona herramientas, documentación y estándares que ayuden a desarrolladores, organizaciones y profesionales de seguridad a crear aplicaciones web seguras.

**Objetivos de Aprendizaje:**

* Aprender a identificar y explotar vulnerabilidades de inyección SQL utilizando herramientas automatizadas como SQLmap.
* Familiarizarse con el análisis de aplicaciones web vulnerables a ataques SQL Injection.
* Comprender las implicaciones de la inyección SQL para la seguridad de las bases de datos y cómo prevenirla.
* Desarrollar la capacidad de implementar pruebas de penetración y análisis de seguridad en aplicaciones web.
* Comprender el alcance de OWASP.

**Marco Teórico:**

* **Inyección SQL (SQL Injection):**  
  La inyección SQL es una técnica de ataque cibernético que permite a un atacante ejecutar código SQL malicioso en una aplicación web. Este ataque se produce cuando una aplicación web no valida adecuadamente los datos introducidos por el usuario, permitiendo que estos datos se incluyan directamente en una consulta SQL. Esto puede permitir que un atacante obtenga acceso no autorizado a bases de datos, manipule datos y ejecute comandos maliciosos.
* **OWASP**

(Open Worldwide Application Security Project) es una comunidad abierta y sin fines de lucro que se dedica a mejorar la seguridad del software. Su propósito principal es proporcionar herramientas, documentación y estándares que ayuden a desarrolladores, organizaciones y profesionales de seguridad a crear aplicaciones seguras. Uno de sus proyectos más conocidos es el OWASP Top Ten, una lista de las diez principales vulnerabilidades de seguridad en aplicaciones web, actualizada periódicamente según las amenazas más críticas observadas en la industria (OWASP, 2021).

* **SQLmap:** es una herramienta de prueba de penetración automatizada de código abierto diseñada para detectar y explotar vulnerabilidades de inyección SQL en aplicaciones web. SQLmap realiza pruebas de seguridad a sitios web con el objetivo de identificar debilidades y simula ataques para demostrar cómo un atacante podría comprometer la seguridad de una base de datos.

**Desarrollo de la Práctica:**

1. **Instalación de SQLmap:**
   * Instrucciones para instalar SQLmap en la máquina local:
     1. Para **Linux/Mac**:
     2. $ sudo apt-get update
     3. $ sudo apt-get install sqlmap
2. **Exploración de un Sitio Web Vulnerable:**
   * Los estudiantes usarán un entorno de laboratorio con una aplicación web vulnerable a inyección SQL. Puede utilizarse un entorno como [Damn Vulnerable Web Application (DVWA)](https://github.com/ethicalhack3r/DVWA) y una aplicación simple configurada con una base de datos vulnerable.
   * **Activar la vulnerabilidad de SQL Injection en DVWA**:
     1. Acceder a DVWA, instalarlo de no existir. Acceder desde Ubuntu Desktop: http://[IP-SERVER]/DVWA/
     2. Iniciar sesión con las credenciales por defecto (usuario: admin, contraseña: password).
     3. Configurar el nivel de seguridad en "Low" para habilitar la vulnerabilidad de inyección SQL.
     4. Acceder a la página de "SQL Injection".
     5. Probar URL vulnerable: http://[IP]/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit
   * **Activar la vulnerabilidad de SQL Injection en DVWA**
3. **Uso de SQLmap:**
   * Una vez que los estudiantes tengan un sitio web vulnerable en funcionamiento, deberán utilizar SQLmap para detectar la vulnerabilidad. El siguiente comando es el ejemplo básico:
   * $ sqlmap -u “http://localhost/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" --batch --threads=10
     1. Descripción del comando:
        + **-u**: La URL del sitio web vulnerable.
        + **--batch**: Ejecuta SQLmap sin pedir confirmaciones al usuario.
        + **--threads=10**: Aumenta la velocidad de las pruebas utilizando 10 hilos concurrentes.
   * Los estudiantes deben analizar las respuestas de SQLmap para identificar si la aplicación es vulnerable y qué tipo de base de datos está usando.
4. **Explotación de la Vulnerabilidad:**
   * Una vez identificado que la aplicación es vulnerable, se puede proceder a obtener información de la base de datos.
   * Por ejemplo, el siguiente comando permite obtener los nombres de las bases de datos:
   * sqlmap -u "http://localhost/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" --dbs
   * Los estudiantes deben intentar obtener otras informaciones valiosas como las tablas (--tables) o los datos de una tabla específica (--dump).
5. **Simular SQL Injection con un sitio web desarrollado por los estudiantes**
   * **Crear base de datos vulnerable (en MySQL)**

CREATE DATABASE estudianteDB;

USE estudianteDB;

CREATE TABLE empleados (

id INT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(50),

cargo VARCHAR(50)

);

INSERT INTO empleados (nombre, cargo) VALUES ('Ana', 'Analista'), ('Luis', 'Desarrollador');

* + **Desarrollar una aplicación web vulnerable**

Archivo: /var/www/html/sitiovuln/index.php

<?php

$conn = mysqli\_connect("localhost", "root", "", "estudianteDB");

$id = $\_GET['id'];

$sql = "SELECT \* FROM empleados WHERE id = $id";

$result = mysqli\_query($conn, $sql);

while($row = mysqli\_fetch\_assoc($result)) {

echo "Nombre: " . $row['nombre'] . " - Cargo: " . $row['cargo'] . "<br>";

}

?>

* + **Acceder al sitio web vulnerable**

Desde el navegador o Kali

<http://[IP-SERVER]/sitiovuln/index.php?id=1>

?id=1 OR 1=1

* + **Explotar la** **vulnerabilidad con sqlmap (desde Kali)**

$ sqlmap -u "http://[IP]/sitiovuln/index.php?id=1" –dbs

1. **Prevención de SQL Injection.**

* Al final de la práctica, se les debe presentar a los estudiantes las mejores prácticas para prevenir inyecciones SQL, que incluyen:
  + Validación y sanitización de las entradas del usuario.
  + Uso de consultas preparadas (prepared statements).
  + Implementación de control de acceso adecuado en la base de datos.
  + Uso de ORM (Object Relational Mapping) que automáticamente previene inyecciones SQL.
  + Identificar el error.
  + Corregirlo usando consultas preparadas con mysqli\_prepare().
  + Comparar comportamiento con la versión vulnerable.

**Conclusión:**

Esta práctica proporcionará a los estudiantes una comprensión práctica de cómo funcionan los ataques SQL Injection y cómo pueden ser detectados y explotados utilizando herramientas como SQLmap. También les permitirá entender las implicaciones de seguridad y la importancia de implementar buenas prácticas de desarrollo para proteger las aplicaciones web de estos ataques.

**Referencias bibliográficas**