Informe de Incidente – Vulnerabilidad de Inyección SQL en DVWA

Introducción

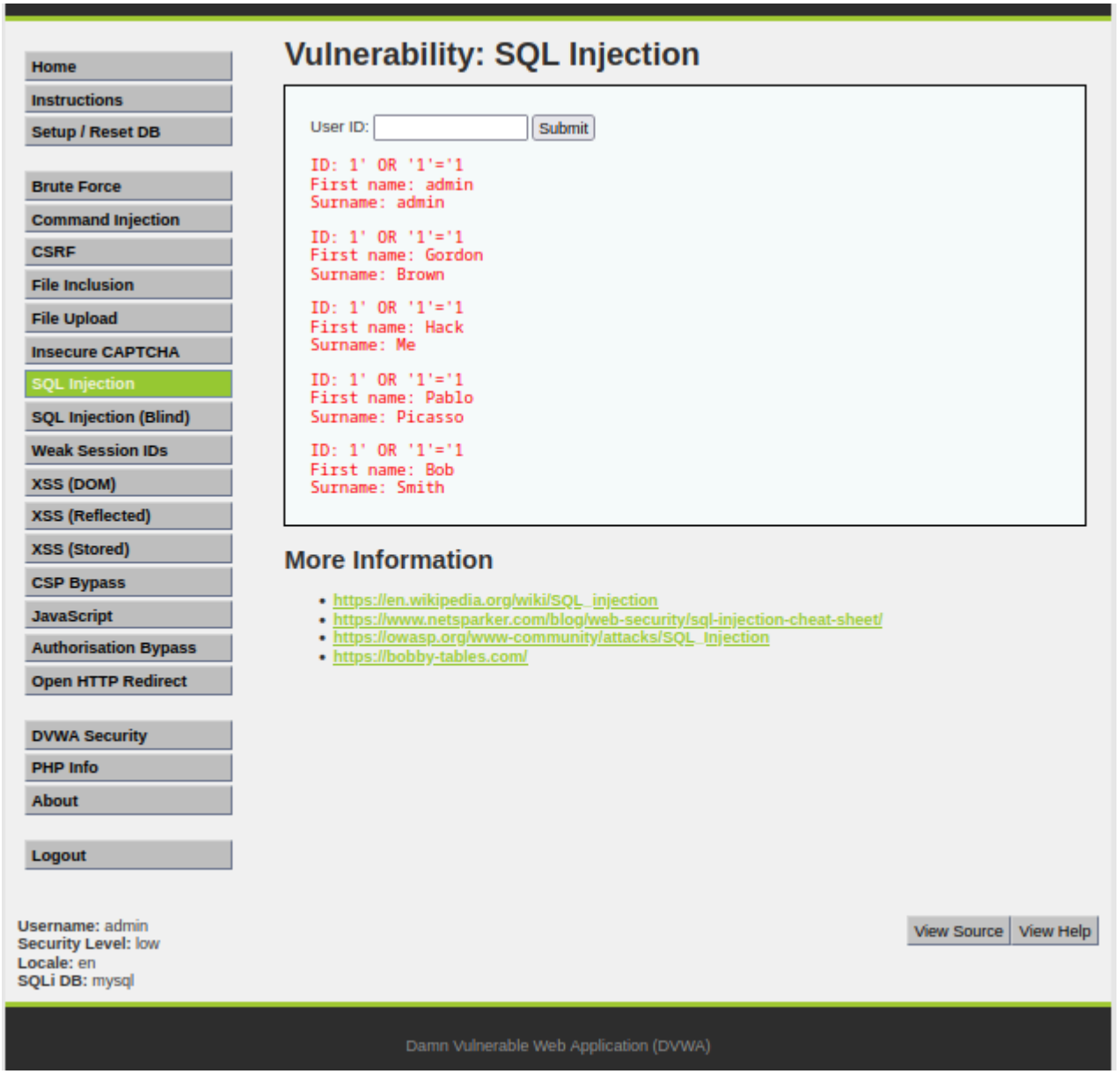
Este informe de incidentes describe la detección y explotación de una vulnerabilidad de **Inyección SQL** en la aplicación web **DVWA**.

La prueba se realizó en una **máquina virtual de Debian** para analizar el impacto de una validación inadecuada de entradas en la seguridad de aplicaciones web.

Descripción del Incidente

Durante la evaluación de seguridad de DVWA, utilizamos el módulo **SQL Injection**, configurando la simulación a nivel de seguridad bajo.

Generamos el incidente en la pantalla de SQL Injection, que nos permite concatenar lo introducido en una query SQL (la inyección). En el caso de la práctica nos pedía utilizar el USER ID.



Método de Inyección SQL Utilizado

Para reproducir y demostrar la vulnerabilidad, se utilizó el siguiente *payload* SQL en el campo **User ID**:

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

De este modo inyectamos en la query normal, la condición de que el ID sea igual a sí mismo. Quedando la siguiente query:

```
SELECT first_name, last_name FROM users WHERE user_id = '1' OR '1'='1';
```

Así hemos introducido esta condición lógica que siempre se evalúa como verdadera y en consecuencia, nos devolverá ve todos los registros de la tabla `users`, independientemente del identificador proporcionado por el usuario.

Impacto del Incidente

La explotación de esta vulnerabilidad podría permitir a un atacante :

- Acceder y extraer información confidencial almacenada en la base de datos.
- Listar todos los usuarios sin necesidad de autorización.
- Modificar, eliminar o comprometer datos sensibles de la aplicación.

Esto representa un riesgo significativo para los tres pilares que hemos revisado en este curso sobre los que nos tenemos que basar en nuestras actuaciones, **confidencialidad**, **integridad** y **disponibilidad**.

Recomendaciones

En base a los resultados de la evaluación, recomiendo las siguientes medidas:

1. **Implementar una validación de entradas** más estricta para todos los datos proporcionados por el usuario.
2. **Utilizar consultas parametrizadas** para prevenir la inyección de código SQ, y revisar dichas consultas periódicamente por si surgiera algún nuevo exploit.
3. **Realizar auditorías de seguridad** de manera regular para prepararse mejor.
4. **Capacitar y concienciar** al personal técnico y no técnico en prácticas de desarrollo seguro.

Conclusiones

La identificación y explotación exitosa de esta vulnerabilidad de Inyección SQL en esta simulación de DVWA pone de manifiesto la importancia de la proactividad en cuanto a la seguridad durante el desarrollo y mantenimiento de aplicaciones web.

El implementar controles de seguridad periódicos y el cumplimiento de buenas prácticas de ciberseguridad son esenciales para proteger de manera actualizada y constante nuestros activos críticos y garantizar la continuidad del servicio.