Ataques por inyección SQL

Jonathan Ordoñez Cubero

- ¿Qué es SQL injection?
- Tipos de ataques
- Asegurar nuestra aplicación web
- Visión del atacante
- Demostración en un “entorno seguro” (DVWA)
- Cuestiones

¿Qué es SQL injection?

- Se conoce como Inyección **SQL** al hecho de incrustar código **SQL** intruso y a la porción de código incrustado.

- Estos ataques generalmente se realizan a través de formularios.

- El origen de la vulnerabilidad reside en el incorrecto **chequeo** de las variables utilizadas en un programa que contiene código **SQL**.

Tipos de ataques

- In-band SQL injection
  - Error-based SQL injection
  - Union-based SQL injection

- Blind SQL injection
  - Boolean-based Blind SQL injection
  - Time-based Blind SQL injection

Tipos de ataques: in-band SQL injection

Es el ataque más común, se produce cuando un atacante es capaz de usar el mismo canal de comunicación para lanzar el ataque y obtener resultados.

- **Error-based SQL injection.** Se basa en mensajes de error lanzados por el servidor de base de datos para obtener información sobre la estructura de la base de datos.

- **Union-based SQL injection.** Este ataque se aprovecha del operador **UNION** de **SQL** para combinar los resultados de varias instrucciones **SELECT** en un solo resultado, el cual es devuelto como parte de la respuesta **HTTP.**

Tipos de ataques: Blind SQL injection

Es un ataque en el que no se muestran mensajes de error al no producirse resultados correctos ante una consulta a la base de datos, mostrándose siempre el mismo contenido.

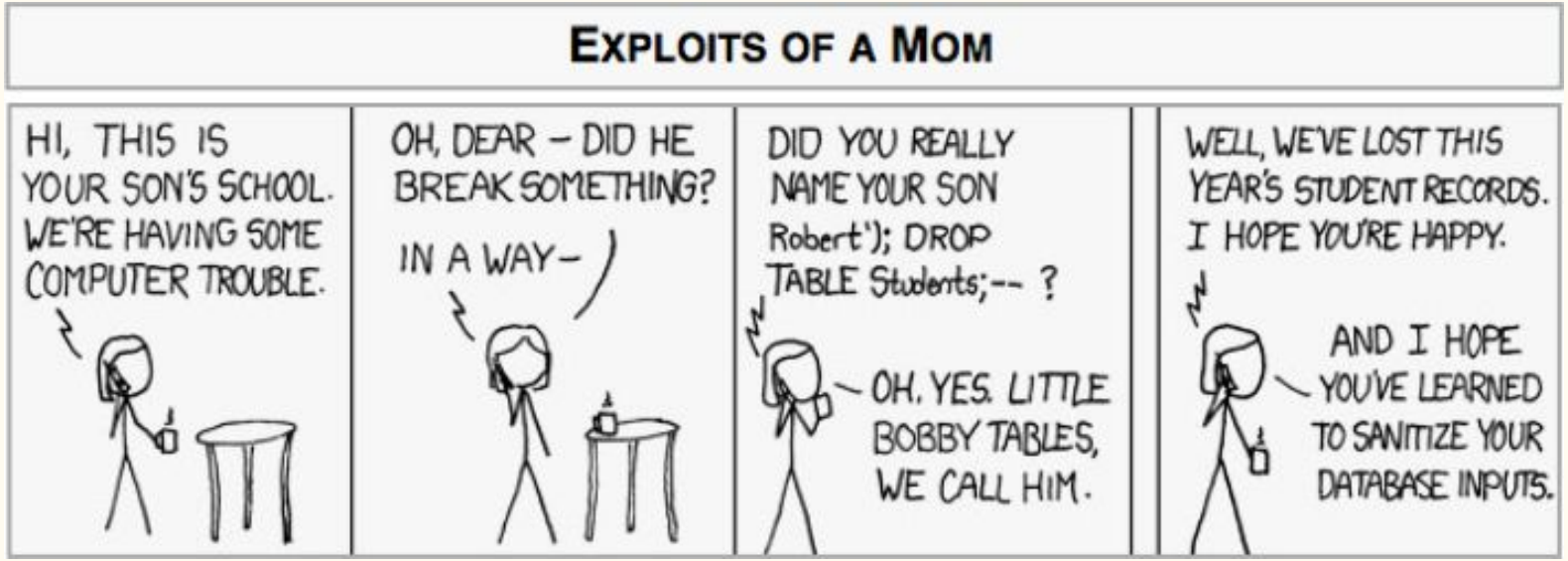
- **Boolean-based Blind SQL injection.** El atacante puede inyectar una consulta que devuelva verdadero (**http://newspaper.com/items.php?id=2 and 1=1**) o falso (**http://newspaper.com/items.php?id=2 and 1=2**) para intentar obtener información de la página.

- **Time-based Blind SQL injection.** Son técnicas que automatizan la extracción de información a ciegas usando retardos de tiempo y han ido especializándose en diferentes tecnologías de bases de datos para generar retardos de tiempo.

Asegurar nuestra aplicación web

- Evitar que el usuario pueda introducir libremente caracteres, ya que podrían aprovecharse de las comillas simples y dobles.
- Dar al usuario los privilegios mínimos para acceder al contenido que necesite.
- Proteger los formularios, por ejemplo, usar <input type="password"> y no <input type="text"> para introducir contraseñas.
- Verificar **SIEMPRE** los datos que introduce el usuario.
- Programar bien.

Asegurar nuestra aplicación web



Visión del atacante

- Hay distintos tipos de atacantes: Pentester, auditores y ciberdelincuentes.
- Se debería tener bastantes conocimientos del sistema que queremos perpetrar.
- La actitud es importante.
- El uso de herramientas para automatizar o facilitar los ataques: Havij, FOCA PRO, SQLInjector, ...
- No uses tu propia conexión a internet, o tendrás visitas no deseadas.

Demostración en un “entorno seguro”

He montando una aplicación web que permite probar vulnerabilidades, sin riesgos, ya que es tu propia máquina virtual, se llama DAMN VULNERABLE WEB APPLICATION.

Se puede descargar de: https://github.com/ethicalhack3r/DVWA/

Su instalación es sencilla y está bastante completo para empezar a prácticar.


¿Preguntas?