Pontificia Universidad Católica de Chile
Departamento de Ciencia de la Computación
IIC2413 - Bases de Datos

Guía 7 - Acceso Programático e Inyecciones SQL

Profesores: Andrés Cádiz
Raimundo Herrera
Matías Toro

En este laboratorio usted hackeará el sistema de bases de datos del curso a través de una página web.

P1. Conéctese vía SSH al servidor `codd.ing.puc.cl`, usando el usuario `alumno` y contraseña `alumno123`. Luego conectese a la base de datos `profesor` (e.g. psql profesor) usando la misma contraseña, donde encontrará las tablas que usaremos. Puedes revisar los detalles de las tablas usando `\dt` y `\d+ TABLA`.

(a) En `peliculas` encontrará nombres, año y rating de varias películas.. Escriba una consulta SQL para obtener los datos de alguna(s) pelicula(s) escogida por usted (puede devolver las tuplas enteras) ordenadas por año (columna `anho`).

(b) En `puntos.iic2413` encontrará una tabla con puntos extras para el control bonus de IIC2413 de usted y sus compañeros (obviamente no tienen nada que ver con la realidad... ¿o sí?). Escriba una consulta SQL que obtenga solo **sus** puntos extras del control bonus (puede devolver la tupla entera; puede usar una condición sobre `nombre`).

(c) Si le parece injustos los puntos extras, puede intentar cambiarlos. Escriba una instrucción SQL que modifique **solamente sus punto extras**. Como es de esperar, la base de datos está preparada para este tipo de "ataques". Escriba el resultado de su instrucción.

(d) Tenga en cuenta que en Postgres se pueden hacer consultas de la forma siguiente:

- `SELECT table_name, table_schema FROM information_schema.tables;`
- ```
  SELECT column_name, data_type FROM information_schema.columns
  WHERE table_name='TABLA' AND table_schema='ESQUEMA';
  ```

Ejecuta la primera consulta para ver todas las tablas y sus esquemas. Después ejecute la segunda consulta para ver solo las columnas de la tabla `puntos.iic2413` y sus tipos. (Serán útiles estas consultas.)

P2. Adicionalmente usted se entera de que existe una página web (`http://codd.ing.puc.cl/~profesor/`) que se conecta a la misma base de datos, pero que extrae la información de películas. En dicha página, usted puede ingresar algún nombre, y se entregarán las 250 películas que comiencen con ese nombre, ordenadas por nombre.

¿Es está página segura ante inyecciones SQL? (*Spoiler*: No.) Es momento de ponerlo a prueba. Todo su poder se basa en la capacidad de escribir en el campo de texto "nombre". Su objetivo
  ```

es realizar inyecciones SQL a través del campo de texto para intentar cambiar sus puntos extras. *Hints*:

- La base de datos requiere que los puntos esten entre 0.0 y 1.0.
- Puede inyectar la consulta de P1 (d), para saber los nombre de las columnas.
- Se sabe que el sistema de base de datos es Postgres (hay formas de adivinar el sistema usado; p.ej. se puede probar con consultas que solo funcionan con un sistema particular).
- Parece que el programador dejo accidentalmente en el código fuente de la página web un link a github que podría serle útil.
- Utilice `--` para comentar todo lo que viene a continuación.
- Si la página arroja algún error, falla o no devuelve ningún resultado, no *siempre* significa que su ataque fue infructuoso. ¿Acaso esperaba un mensaje de felicitaciones por hackear la base de datos?

Tendrá que ingresar inyecciones SQL para:

(a) devolver todas las tablas en la base de datos;

(b) devolver las columnas de la tabla `puntos.iic2413` y sus tipos;

(c) devolver su puntaje extra en la tabla `puntos.iic2413`;

(d) cambiar su puntaje extra en la tabla `puntos.iic2413`;

(e) cambiar su comentario en la tabla `puntos.iic2413`.

(f) Escriba una propuesta de código php que arregla esta vulnerabilidad. Indique que linea debe cambiarse por cual. Hint: vea el codigo fuente del html para ver donde podría encontrar el código fuente de la aplicación.