Laboratorio 6 - Acceso Programático e Inyecciones SQL

Profesores: Claudio Gutiérrez
Matías Toro
Auxiliares: Scarlett Plaza
Daniel Radrigán
Cristian Salazar
Fran Zautzik

En este laboratorio usted hackeará el sistema de bases de datos del curso a través de una página web. Usted debe entregar un archivo de texto con las consultas de **P1** y las inyecciones de **P2** junto a alguna evidencia de su hackeo (foto).

P1. [15 PUNTOS] Conéctese vía SSH al servidor, donde encontrará las dos tablas que usaremos. Puedes revisar los detalles de las dos tablas usando `\d+ TABLA`.

(a) [5 PUNTOS] En `uchile.transparencia` encontrará las remuneraciones brutas de todos los empleados de la Universidad desde enero de 2015. Escriba una consulta SQL para obtener los datos de todos los empleados con un apellido paterno elegido por usted (puede devolver las tuplas enteras) ordenados por saldo (la columna `total`).

(b) [5 PUNTOS] En `nota.cc3201` encontrará las notas finales de CC3201 de usted y sus compañeros (obviamente no tienen nada que ver con la realidad... ¿o sí?). Escriba una consulta SQL que obtenga solo **su** nota final del ramo (puede devolver la tupla entera; puede usar una condición sobre `id` o `nombre` o lo que le parece conveniente).

(c) [5 PUNTOS] Si le parece injusta la nota, puede intentar cambiarla. Escriba una instrucción SQL que modifique **solamente su nota**. Como es de esperar, la base de datos está preparada para este tipo de "ataques". Escriba el resultado de su instrucción.

(d) [5 PUNTOS] 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 `nota.cc3201` y sus tipos. (Serán útiles estas consultas.)

P2. [45 PUNTOS] Adicionalmente usted se entera de que existe una página web (`http://cc3201.dcc.uchile.cl`) que se conecta a la misma base de datos, pero que extrae la información de transparencia de funcionarios de la universidad. En dicha página, usted puede ingresar

algún apellido paterno, y se entregarán los 250 sueldos mensuales más altos de empleados con **exactamente** ese apellido.

¿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 "apellido". Su objetivo es realizar inyecciones SQL a través del campo de texto para intentar cambiar su nota. *Hints*:

- La base de datos requiere que la nota esté entre 1.0 y 7.0.
- 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.
- Si la página arroja algún error, 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) [7 PUNTOS] devolver todas las tablas en la base de datos;

(b) [7 PUNTOS] devolver las columnas de la tabla `nota.cc3201` y sus tipos;

(c) [7 PUNTOS] devolver su nota en la tabla `nota.cc3201`;

(d) [7 PUNTOS] cambiar su nota en la tabla `nota.cc3201`;

(e) [7 PUNTOS] cambiar su comentario en la tabla `nota.cc3201`.

(f) [10 PUNTOS] Escriba una propuesta de código python 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.