Laboratorio 9 - Inyecciones SQL

En este laboratorio usted revisará una base de datos con información de transparencia y notas de cursos. Luego hackeará la base de datos a través de una aplicación no segura para cambiar su nota. Debe entregar un archivo de texto con las consultas de **P1** y las inyecciones de **P2**.

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 mensuales de todos los empleados de la Universidad para el año 2023. Escriba una consulta SQL para obtener los datos de todos los empleados con un apellido paterno elegido por usted que le da resultados no vacíos (puede devolver las tuplas enteras) ordenados por saldo (la columna `total`) descendente.

(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] Tenga en cuenta que en Postgres se pueden hacer consultas de esta forma:

- `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] En `http://cc3201.dcc.uchile.cl/v1/`, puedes encontrar la aplicación web (hipotética) que se llama *Transparencia UChile*. Puedes entrar un apellido paterno y ver algunos empleados con ese apellido, ordenado por el sueldo bruto mensual. Piense que usted es un simple usuario de la aplicación, por lo tanto, usted no tiene privilegios en la base de datos; usted tampoco tiene acceso a leer o modificar el esquema de la base de datos ni el código de la aplicación. Todo su poder se basa en la capacidad de entrar un apellido en la aplicación. Su objetivo es ingresar inyecciones SQL para cambiar su nota. *Hints*:

- La base de datos requiere que la nota esté entre 1,0 y 7,0.
- En base a los resultados de la aplicación, se pueden adivinar los atributos (y sus tipos) de la consulta usada por la aplicación (`nombres`, `apellido_p`, ...).

- 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).
- No se puede cambiar la primera parte de la consulta; hay que continuar la consulta.
- Se puede terminar la consulta y empezar otro comando SQL con '`;`'.
- Si quiere ver algún resultado, solo se puede usar una consulta (sin ingresar '`;`').
- Si la aplicación no da ningún mensaje, no *siempre* significa que su ataque fue infructuoso. ¿Acaso esperaba un mensaje de felicitaciones por hackear la base de datos?
- De ser necesario, se puede transformar un valor requerido en un valor de otro formato; lo que importa es que pueda deducir la información necesaria a partir de la respuesta.
- Si no necesita una columna particular, puede asignar un constante a esa columna con el tipo correcto; p.ej., `SELECT ..., 'X' AS apellido_m, 0 AS mes, ...`

Tendrá que ingresar inyecciones SQL para:

(a) [9 PUNTOS] devolver todas las tablas en la base de datos;

(b) [9 PUNTOS] devolver las columnas de la tabla `nota.cc3201` y sus tipos;

(c) [9 PUNTOS] devolver su nota en la tabla `nota.cc3201`;

(d) [9 PUNTOS] cambiar su nota en la tabla `nota.cc3201`;

(e) [9 PUNTOS] cambiar su comentario en la tabla `nota.cc3201`.

En `http://cc3201.dcc.uchile.cl/v2/`, existe otra versión de la aplicación que usa sentencias pre-compiladas. Intente realizar sus ataques de nuevo. ¿Funcionan?

Puede revisar el código de ambas aplicaciones en el servidor desde SSH (no dentro de `psql`) con los siguientes comandos (algo útil para los proyectos):

- `more /var/www/html/v1/index.php`
- `more /var/www/html/v2/index.php`