Security

Christian Rudder

August 2024

Contents

1 Introduction **3**

2 SQL **3**

2.1 SQL Basics 3

This page is left intentionally blank.

1 Introduction

Hackers, Security issues, and exploits all exists, because the software we use isn't perfect. It's written by another human, and even the best of us don't account for every possible scenario. This doesn't mean just software, but languages, protocols, and even hardware.

The languages we use are solutions, we thought would work, but they are only as good as the people who wrote them. Binary is a solution, it's not the intrinsic solution, it's just the one we use. So we built assembly, and then we built C, and so on abstracting the difficult parts. But the more we abstract, the more we lose control, and the more cracks appear.

2 SQL

2.1 SQL Basics

SQL stands for "***Structured Query Language***," used to query against databases with tables containing columns of data, which most often relate to each other.

Using select words like `SELECT`, `FROM`, `WHERE`, ignoring case. It's good practice to use **all caps for SQL keywords**, and **lowercase for table and column names**. Here's a simple example:

```sql
SELECT * FROM my_table
```

Selects all (*) columns from the table `my_table`.

Now, we are a record company with bands, albums, and songs:

```sql
CREATE DATABASE test; -- creating a test database
DROP DATABASE test; -- deleting the test database

CREATE DATABASE concise_records; -- creating our database
USE concise_records; -- selecting our database to run commands on it

CREATE TABLE bandds (); -- creating a table for our bands
DROP TABLE bandds; -- deleting it because of our typo

-- Create bands table: artist names (at most 255 characters), CANNOT be NULL/EMPTY
CREATE TABLE bands (
    name VARCHAR(255) NOT NULL
);

-- Add id column to bands, auto increment, not NULL, make this column important
ALTER TABLE bands
ADD COLUMN id INT NOT NULL AUTO_INCREMENT PRIMARY KEY;
```

We created the database, `concise_records`, and a table `bands` with two columns: `name` and `id`.

The `PRIMARY KEY` acts as an ID for each row, useful for drawing a thread of relationships between tables where the ID is present.

Definition 2.1: Primary Key

A column which identifies each row in a table. It must be unique, and it cannot be NULL.

To create our albums table:

```
CREATE TABLE albums (
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    release_date DATE NOT NULL,
);
```