Execute No Evil

> Author: howard41436 | 楊皓丞 | B06902097

Description

(1) New Message: "Hey dude. So we have this database system at work and I just found an SQL injection point. I quickly fixed it by commenting out all the user input in the query. Don't worry, I made the query so that it responds with boss's profile, since he is kind of the only person actively using this database system, and he always looks up his own name, lol. Anyway, guess we'll go with this til' the sysadmin comes and fixes the issue."

Huh, so hear no evil, see no evil, ... execute no evil?

source: source.php

Vulnerability

SQL Injection

Solution

From the source we can see that user input is concatenated into ther query string, so it is vulnerable to sql injection. However, user input is commented by `/* */`, and the character `*` is blacklisted. But, in mysql, `/*! */` syntax will make the content between uncommented. Therefore, we can perform a union-based sql injection.

The payload `name=!'Geronimo' UNION SELECT 1, table_schema, 3 FROM information_schema.schemata UNION SELECT 1, 2,` can help us find all the database names. By similar methods, we can find the target table name and column name. At last, we can use the payload `name=!'Geronimo' UNION SELECT 1, whatsthis, 3 FROM ctf.flag UNION SELECT 1, 2,` to get the flag.