Mitigating SQL Injection Vulnerability with Prepared Statements

Security

Otago Polytechnic
Dunedin, New Zealand

Do you want sql injection?

```
$id = $_GET['id'];
$query = "SELECT * FROM users WHERE id=$id";
$db->query($query);
```

Because that's how you get SQL injection!

Step 1

```
$id = $_GET['id'];
```

Take untrusted input,...

Step 2

```
$id = $_GET['id'];
$query = "SELECT * FROM users WHERE id=$id";
```

...construct an SQL query string with the input,...

Step 3

```
$id = $_GET['id'];
$query = "SELECT * FROM users WHERE id=$id";
$db->query($query);
```

…send the query string to the database. Now you're on your way to executing somebody else's code on your server

Mitigating the risk

First, we should **never** just accept untrustworthy input.

```
$id = $_GET['id'];
```

becomes

```
if(is_numeric($_GET['id'])) {
    $id =(int) $_GET['id'];
}
```

since id's are meant to be integers in our application.

Mitigating the risk

Next, instead of constructing a query string, we create a *prepared statement*.

```
$query = "SELECT * FROM users WHERE id=$id";
```

becomes

```
$stmt = $db->prepare("SELECT * FROM users WHERE id=?")
```

Mitigating the risk

Finally, we bind the variables in our prepared statement and execute it.

```
$db->query($query);
```

becomes

```
$stmt->bind_param("i", $id);
$stmt->execute();
```

One more thing ...

We can create database user accounts with varying levels of privileges. Set up and use accounts with the least amount of privileges needed.

For example, if your application only needs to read from the database, connect as a user that only has read privileges.

Today's Lab

1. Download code for this lab from `http://sec-student.foo.org.nz/~tclark/lab7.1.tgz`
2. The file `login.php` has already be modified to use prepared SQL statements. Use it as an example.
3. Modify `reset.php` to use prepared SQL statements.
4. Additional information is available at `http://php.net/manual/en/mysqli.quickstart.prepared-statements.php`

As always, we use PHP here as an example, but similar techniques apply to other languages.