

Application vulnerabilities and defences (part 2)

- The problem with the examples in lecture (6-2) is that the application allows unfiltered input into the SQL.
- The application needs to filter out all characters with special meaning in SQL, like single or double quotes, semi-colons, the comment introducer -- etc.

**Avoiding
SQL
injection**

- Never concatenate user input with application SQL to form the SQL sent to the database.
- The easy way to do this is to use parameterized statements.
- Parameterized statements are where the variable parts of the SQL are replaced with markers (usually ?).
- Instead of concatenating the user input for the email address like this:

```
select email from users where email =  
'<user_input>'
```

```
select email from users where email = ?
```

- The SQL is prepared when the SQL Engine parses it, validates it and notes that there is one parameter for the email address. When you execute it, you pass the parameter separately from the SQL. How you do this depends on the language you are using.

```
char *user_input; /* points to user input string */  
SQLPrepare ('select email from users where email = ?');  
SQLBindParameter (1, user_input);  
SQLExecute;
```

Now it does not matter if a user enters any special SQL characters, because they are never parsed by the SQL engine.

- ◆ **Do not follow links from sites that navigate to security-sensitive pages referencing personal or business information.**
- ◆ **Always practice obtaining a list of attacks that have occurred on particular sites or messages boards.**
- ◆ **Do not trust links given on other sites such as e-mail or message boards.**

Avoiding XSS

- To prevent CSRF attack need to generate **Synchronizer Token Pattern**.
- These challenge **tokens** are inserted within the HTML forms and links associated with sensitive server-side operations
- By including a **challenge token** with each request, the developer has a strong control to verify that the user actually intended to submit the desired requests.

Avoiding CSRF