

Application vulnerabilities and defences

In this lecture

We examine the following :

- **SQL injection**
- **XSS**
- **CSRF**

SQL injection

- **SQL injection** is a basic attack used to either gain unauthorized access to a database or to retrieve information directly from the database.
- The principles behind a SQL injection are simple and these types of attacks are easy to execute and master.

- **SQL injection is a code injection technique that exploits a security vulnerability occurring in the database layer of an application.**
- **The vulnerability is present when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed.**

Basic examples of injection attacks

- Assume the application is vulnerable to SQL injection, as it uses unvalidated user input to form SQL strings.
- For instance, the following application has an email form where users enter their email address:

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

where <user_input> is what you enter in the form.

If we enter 'someone@somewhere.com' in the form, the resulting SQL is:

```
select email from users where email = 'someone@somewhere.com'
```

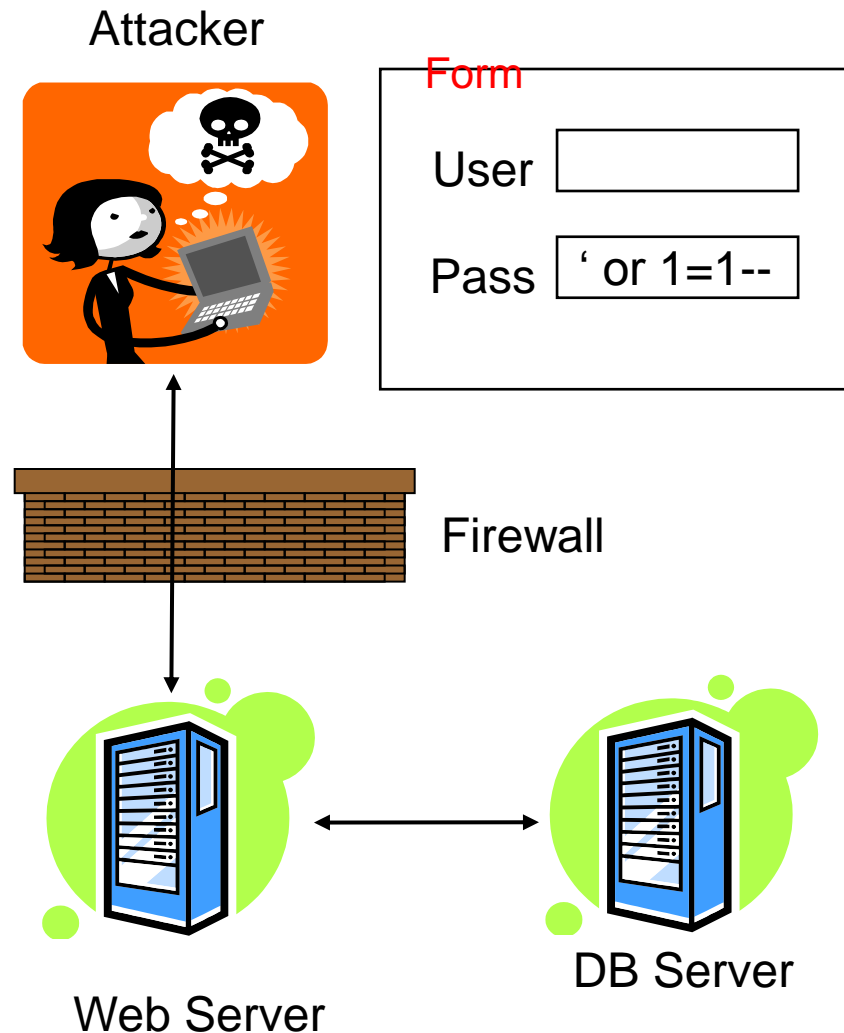
- That is likely to produce a syntax error in the application when the database parses the SQL and objects to the last '.
- Having tried this, the SQL injector already knows you concatenate strings without validating them and that you are vulnerable.
- Now assume the form input is changed to:

someone@somewhere.com' or 'x' = 'x

- The resulting SQL is:
- **select** email **from** users **where** email = **'someone@somewhere.com' or 'x' = 'x'**
- The typical logic that such an application expects is that so long as a row is returned, the email address must be valid, and hence you gain access.

SQL Injection

1. App sends form to user.
2. Attacker submits form with SQL exploit data.
3. Application builds string with exploit data.
4. Application sends SQL query to DB.
5. DB executes query, including exploit, sends data back to application.
6. Application returns data to user.



From SQLi to Bank Accounts



hackers used a SQL injection attack against the California ISP Sebastian to access a database of customers' e-mail addresses, user names and clear text passwords -- and then using that data to steal money from those customers.

- **The hackers claim to have stolen \$100,000 by leveraging user names and passwords taken from a California ISP to access victims' bank accounts.**

Cross-Site Scripting (XSS)

- **Cross-Site Scripting (XSS)** attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted web sites.
- XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user.
- Flaws that allow these attacks to succeed are quite widespread and occur anywhere a web application uses input from a user within the output it generates without validating or encoding it

- An attacker can use XSS to send a malicious script to an unsuspecting user.
- The end user's browser has no way to know that the script should not be trusted, and will execute the script.
- Because it thinks the script came from a trusted source, the malicious script can access any cookies, session tokens, or other sensitive information retained by the browser and used with that site.
- These scripts can even rewrite the content of the HTML page

Cross-Site Request Forgery (XSRF)

- ❑ **CSRF** is an attack which forces an end user to execute unwanted actions on a web application in which he/she is currently authenticated.
- ❑ With a little help of social engineering (like sending a link via email/chat), an attacker may trick the users of a web application into executing actions of the attacker's choosing.
- ❑ A successful CSRF exploit can compromise end user data and operation in case of normal user.
- ❑ If the targeted end user is the administrator account, this can compromise the entire web application.

- **Cross-Site Request Forgery (CSRF)** is an attack that tricks the victim into loading a page that contains a malicious request.
- It is malicious in the sense that it inherits the identity and privileges of the victim to perform an undesired function on the victim's behalf, like change the victim's e-mail address, home address, or password, or purchase something.
- CSRF attacks generally target functions that cause a state change on the server but can also be used to access sensitive data.