CS326 – Systems Security

Lecture 22

SQL Injection and other Web attacks

Elias Athanasopoulos
athanasopoulos.elias@ucy.ac.cy

SQL Injection

- Web apps often use persistent storage (databases)
- Code and storage communicates using Structured Query Language (SQL)
- Injection: mix data with SQL code

SQL Example

```
SELECT * FROM users WHERE name = "Alice";
```

Return from a table named ‘users’ all records where name is *Alice*.

SQL queries and web apps

- Web apps construct SQL queries dynamically

```
// user_name is given by the user
sql_query = " SELECT * FROM users WHERE
            name = ' " + user_name + " '; ”
db.execute_query(sql_query);
```

- *sql_query* embeds the content of *user_name*
- Consider: `user_name = Alice'; DROP TABLE users;`

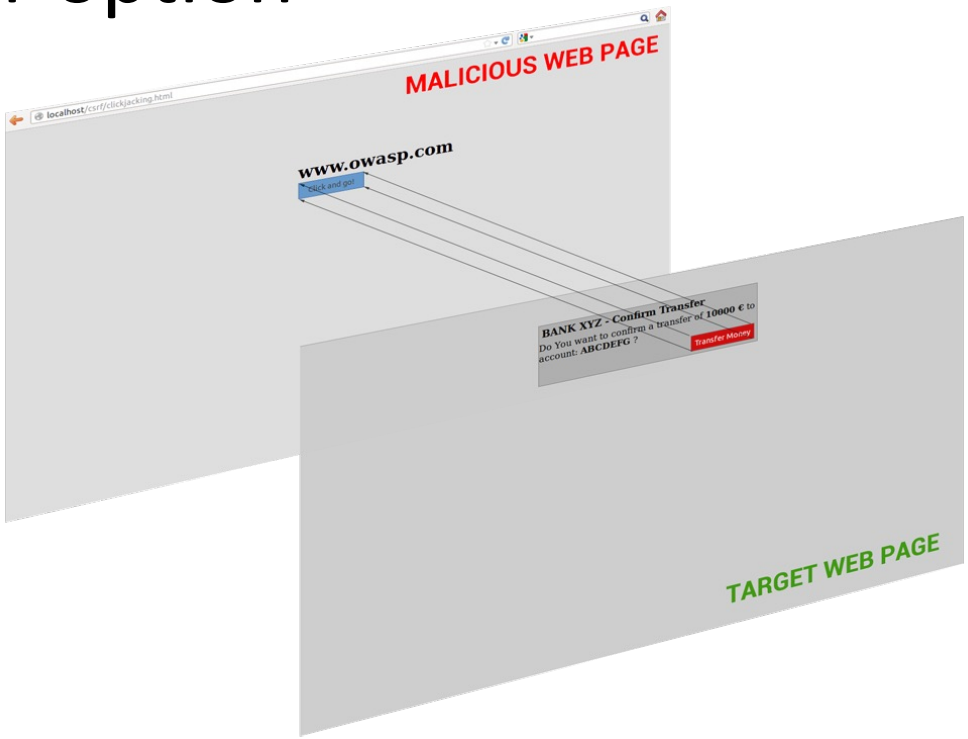
SQL Injection Solutions

- Prepared statements using SQL templates

```
stmt = db.prepare("SELECT * FROM users WHERE
user_name= ?");
stmt.set(1, user_name);
stmt.execute();
```

Clickjacking

- Visual tricks for making the user to click on a particular option



X-Frame-Options header

- The attacker needs to frame the target web site inside the malicious web site

- Prevent a web page from being framed
  - **DENY, SAMEORIGIN**, **ALLOW-FROM uri**,

Automated Data Input

- Scripts can fill in web forms
  - Create automatically many e-mails
  - Brute-force passwords
  - Click ads

CAPTCHAs

(Completely Automated Public Turing Test To Tell Computers and Humans Apart)