Week 02
Web Hacking II

Nathan

Announcements

- CSAW!!!

- Fall Recruitment Event

sigpwny{do_you_pronounce_it_as_sql_or_sql}

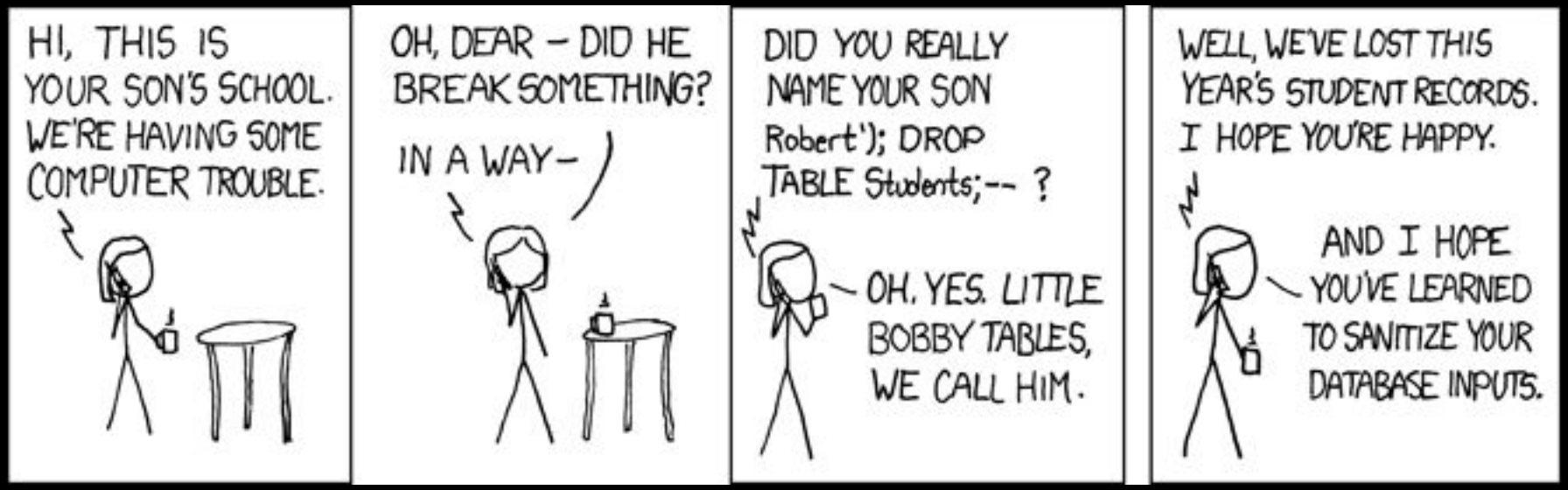


Table of Contents

SQL Injection

Supply input that maliciously changes a SQL statement

SQL Overview

- Language used to store/retrieve things in a database
- Server code will construct a “query” from user input
  - User logins
  - Search results
  - Payment info

SQL Query Example

```
SELECT * FROM | users | WHERE | username = 'admin' AND password = 'password'
```

Get all "rows" (entries) from...

the table called "users"

such that the following condition is true...

the username column (field) is "admin" and the password column is "password"

SQL Query In Context

```php
<?php
    $username = $_POST['username'];
    $password = $_POST['password'];

    //Actual SQL query is here V
    $query = "SELECT * FROM users WHERE username = '$username' AND password = '$password'";

    $results = $db->query($query);
    $row = $results->fetchArray();

    echo 'Welcome', $row['username'];
?>
```

It puts our username input directly into the query!

What can we insert for **$username** and **$password** to make this misbehave?

```
SELECT * FROM users WHERE username = '$username' AND password = '$password'
```

↓

```
$username = admin'--          <-- is a line comment in SQL!
$password = sigpwny               (like // in C++)
```

↓

```
SELECT * FROM users WHERE username = 'admin'--' AND password = 'sigpwny'
```

↓

```
SELECT * FROM users WHERE username = 'admin'-- ' AND password = 'sigpwny'
```

↓

```
SELECT * FROM users WHERE username = 'admin'
```

This SQL expression will always log us in as the user with username “admin” without needing any password!

SQL Injection Techniques

- Basic **(username for SQL 1 is mrprez420)**
  - Login as other users by changing WHERE clause
  - Use for SQL 1 and 2 challenges
- Union
  - Exfiltrate data from SQL database (users and passwords, etc)
  - Use for SQL 3 challenge
- Blind
  - Sometimes you can't see result of query
  - Use sleep and timing to still leak information
- Sqlmap!

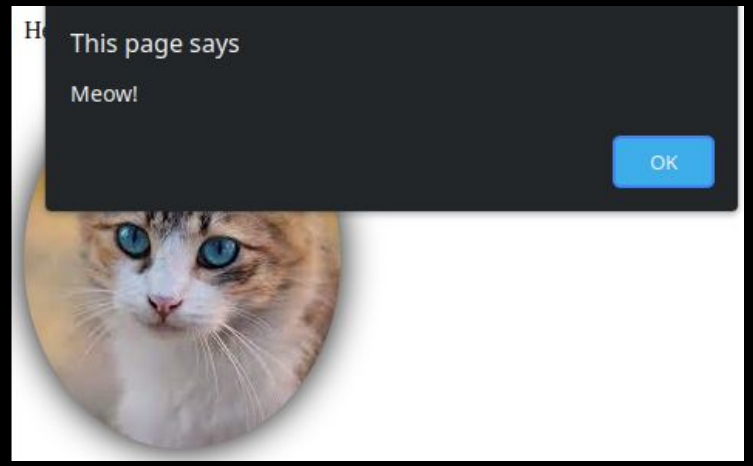
XSS

Put your own javascript on web pages for other clients to execute!

Javascript Recap

- Programming language that interacts with webpage
- Runs in **browser** (client side!)

```
<script>
    document.getElementById("cat").onclick = () => alert("Meow!");
</script>
```



XSS Vulnerability

app.js

```
app.get('/view', function(req, res) {
    let message = req.query.message || "";
    res.render('view', {message:
message});
});
```

view.ejs

```
<body>
  <div class="container">
    <p><%- message %></p>
  </div>
</body
```

This endpoint puts a user's message directly in the HTML!
What can we put as a message that will make the website misbehave?

XSS Vulnerability

view.ejs

```
<body>
  <div class="container">
    <p><%- message %></p>
  </div>
</body
```

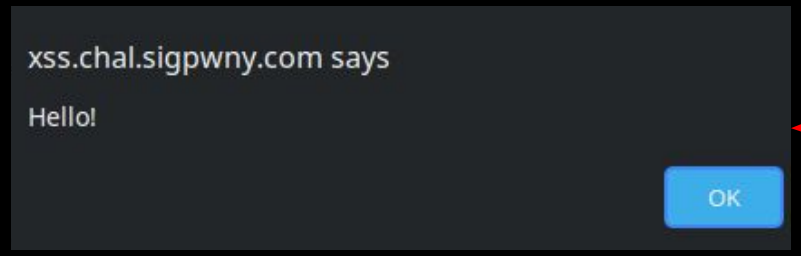
message

“<script>alert(‘Hello!’)</script>”

Result!

```
<body>
  <div class="container">

<p><script>alert('Hello!')</script> </p>
  </div>
</body
```



XSS Techniques

- <script>alert(1)</script>
-
  - also onerror=...
- SVG XSS!

XSS Post-Exploitation

- Send link with your XSS to an admin
- Steal their cookies, read the page contents
- Do any operation as the logged in user
- Why is XSS necessary? Why can't we send them to our own site?
  - Browsers will only let javascript read site info on same domain

Go try for yourself!

https://ctf.sigpwny.com

- 3 SQL chals
- 3 XSS chals
- Fun MSA (Midwest Security Agency) lore

Next Meetings

Weekend Seminar: CSAW Recap!

- How we did
- Solutions to some challenges

Next Thursday: Crypto I!

- Crypto fundamentals (keys, communication, encoding)
- Introduction to basic crypto schemes (symmetric and asymmetric)