

# Hack this workshop!

---

Christopher Riley

# Introduction

---

**What is a security vulnerability?**

# Types of security vulnerabilities

---

# Unvalidated user input

---

## Injection eg SQL, XSS

---

# SQL Injection

```
1 <?php
2
3 $loggedIn = mysqli_query(
4     $db,
5     "
6     SELECT *
7     FROM `users`
8     WHERE (`username` = '$_POST[username]')
9     AND (`password` = '$_POST[password]')
10    "
11 );
```

# SQL Injection

```
1 <?php
2
3 $_POST['username'] = "admin";
4 $_POST['password'] = "' OR '1' = '1";
```



# SQL Injection

```
1 <?php
2
3 $loggedIn = mysqli_query(
4     $db,
5     "
6     SELECT *
7     FROM `users`
8     WHERE (`username` = 'admin')
9     AND (`password` = '' OR '1' = '1')
10    "
11 );
```

## Stored XSS Injection

```
1 <div class="post">
2   <h2 class="posttitle"><?php echo $postTitle?></h2>
3   <div class="postbody">
4     <?php echo $postBody?>
5   </div>
6 </div>
```

# Stored XSS Injection

```
1 <div class="post">
2   <h2 class="posttitle"><b>Any <span>HTML</span> in post title will be rendered</b></h2>
3   <div class="postbody">
4     <h3>As well as anything in the post body</h3>
5     <script>alert('Even javascript!')</script>
6   </div>
7 </div>
```

# Reflected XSS Injection

```
1 <div class="topbar">
2     <?php if (isset($_GET['name'])): ?>
3         <span><?php echo $_GET['name']?></span>
4     <?php else:?>
5         <form method="get" action="/">
6             <input name="name" type="text" placeholder="Enter your name" />
7         </form>
8     <?php endif?>
9 </div>
```

# Variable Injection

```
1 <?php
2
3 $dbhost = 'localhost';
4 $dbuser = 'root';
5 $dbpass = 'mySup3rS3(ur3P4ssW0Rd';
6
7 foreach ($_POST as $key => $value) {
8     $$key = $value;
9 }
10
11 mysqli_connect($dbhost, $dbuser, $dbpass);
```

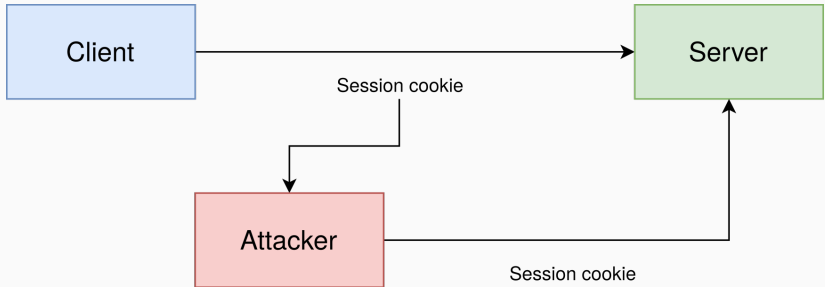
**Lack of or broken access control**

---

# The login cookie



# Session Hijacking





# Exploring vulnerabilities

---

## Trying "bad" values

## Trying "bad" values

- ', %, -

## Trying "bad" values

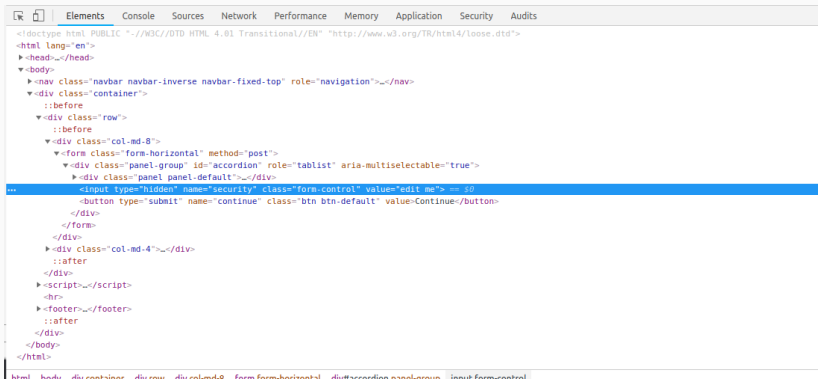
- ', %, -
- `<script>alert('xss');</script>`

## Trying "bad" values

- ', %, -
- `<script>alert('xss');</script>`
- 0.5, aaa...a

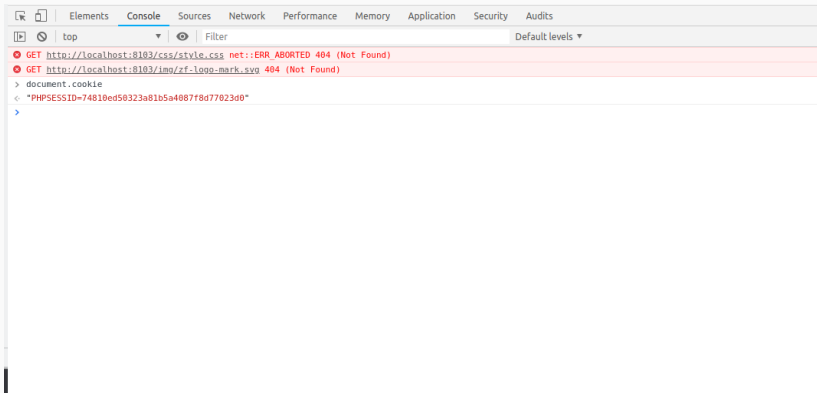
## Using the inspector

# Using the inspector



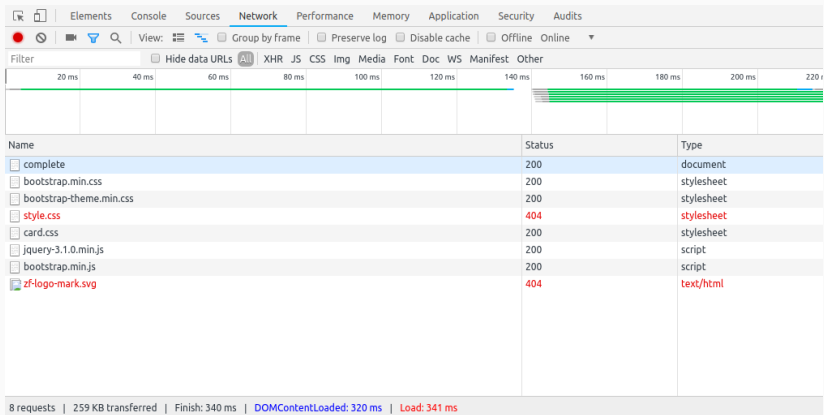
```
<!doctype html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html lang="en">
<head></head>
<body>
  <nav class="navbar navbar-inverse navbar-fixed-top" role="navigation"></nav>
  <div class="container">
    <div class="row">
      <div class="col-md-8">
        <form class="form-horizontal" method="post">
          <div class="panel-group" id="accordion" role="tablist" aria-multiselectable="true">
            <div class="panel panel-default"></div>
            <input type="hidden" name="security" class="form-control" value="edit me">
            <button type="submit" name="continue" class="btn btn-default" value="Continue"></button>
          </div>
        </form>
      </div>
      <div class="col-md-4"></div>
    </div>
  <script></script>
  <hr>
  <footer></footer>
</div>
</body>
</html>
```

# Using the inspector





# Using the inspector



# Using the inspector

The screenshot shows the Chrome DevTools interface with the **Network** panel selected. The top toolbar includes icons for Elements, Console, Sources, Network, Performance, Memory, Application, Security, and Audits. Below the toolbar, the **View:** section has checkboxes for Group by frame, Preserve log, Disable cache, and Offline. The **Filter** section has a dropdown set to **All** and tabs for XHR, JS, CSS, Img, Media, Font, Doc, WS, Manifest, and Other. A timeline at the top shows a sequence of requests, with the last one highlighted in green. The **Headers** panel is open for the selected request, showing the following details:

- Name:** complete
- General:**
  - Request URL:** http://localhost:8103/s7cFmkFFsWRJOIU\_ZclT2a3jLknjNA/complete
  - Request Method:** GET
  - Status Code:** 200 OK
  - Remote Address:** [::1]:8103
  - Referrer Policy:** no-referrer-when-downgrade
- Response Headers:**
  - Cache-Control:** no-store, no-cache, must-revalidate
  - Connection:** keep-alive
  - Content-Type:** text/html; charset=UTF-8
  - Date:** Mon, 29 Oct 2018 21:09:27 GMT
  - Expires:** Thu, 19 Nov 1981 08:52:00 GMT
  - Pragma:** no-cache
  - Server:** nginx/1.13.12

At the bottom left, a summary bar indicates **8 requests** and **259 KB transferred**.

## Digging for information

## Digging for information

- .htaccess, web.config, \*.inc

## Digging for information

- .htaccess, web.config, \*.inc
- robots.txt

## Digging for information

- .htaccess, web.config, \*.inc
- robots.txt
- /

## Digging for information

- .htaccess, web.config, \*.inc
- robots.txt
- /
- /admin, /backups

## Legal implications

---



# Exercises

---

## Conclusion

---

# Thanks

- @giveupalready
- <https://github.com/carnage>
- <https://carnage.github.io>