SECURITY

SQL INJECTION

XSS - CROSS SITE SCRIPTING

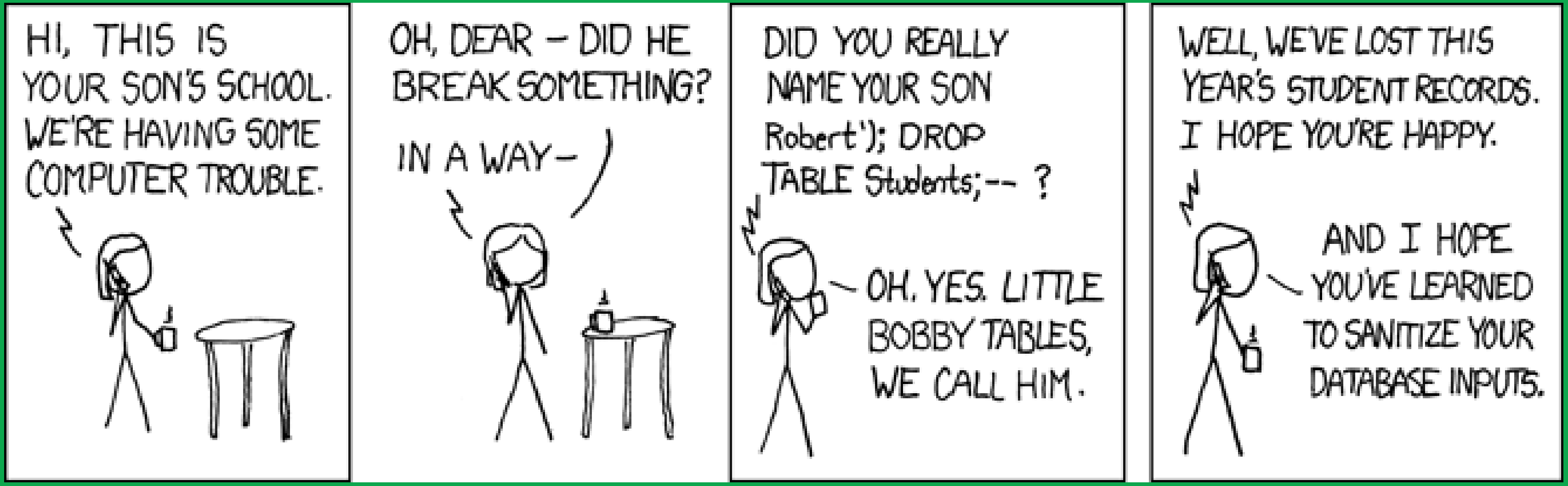
PHISHING
INTRUDER
FRAUD
SCAM
SPYWARE
PASSWORD
IDENTITYTHEFT
VIRUS
CRACKER
HACKER

SQLi - SQL Injection

- What is SQLi?
- It's 2016! Is this still a thing??
- How do we guard against it?

What is SQLi?

> Allowing execution of arbitrary SQL through unfiltered user input


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

Is this still a thing?

- Indian Embassy Hacks - November
- Arizona Voter Registration Database - September
- Ubuntu Forums of 2 million usernames - July
- **YES!**

```
"; select * from posts; -- lulz
```

Show some lulz

Bad example is bad

App

How do we guard against it?

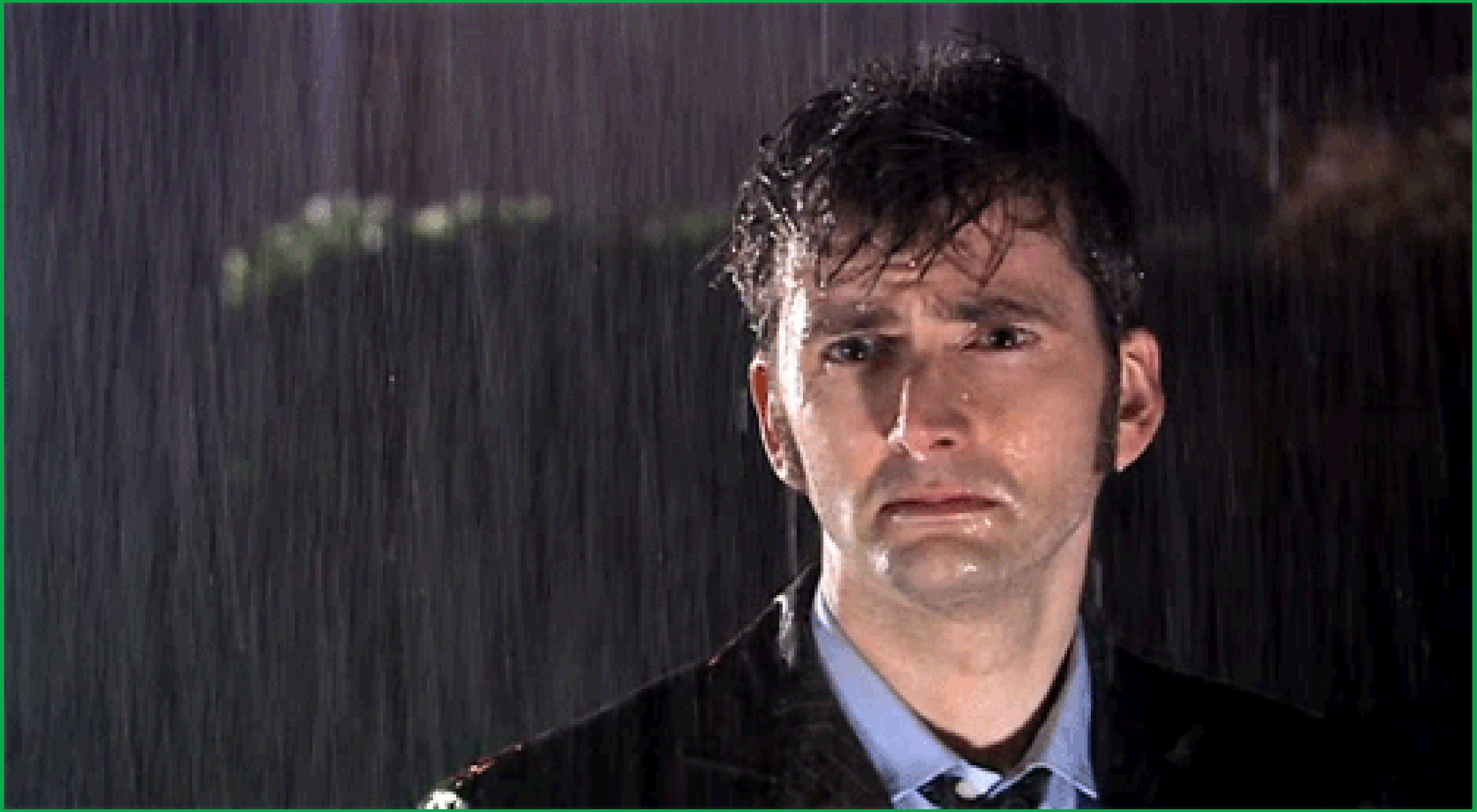
- ORM
- Query parameters
- Keep systems up to date
- Keep gems/modules up to date
- Escape

Parameterized

```
connection.query('SELECT * FROM `users` WHERE `username` = ?', ['superman'])
```

Escape Example

```
def some_sql
  sql = "select u.id, u.email, p.id from users u
        join pages p on p.owner_id = u.id
        and p.created_at between #{ActiveRecord::Base.sanitize(min_page_created_date.strftime('%F'))}
          and #{ActiveRecord::Base.sanitize(max_page_created_date.strftime('%F'))}
        and p.owner_type = 'User'
        and u.id not in (select user_id from posts
          where created_at > #{ActiveRecord::Base.sanitize(last_post_date.strftime('%F'))})"
end
```



XSS - Cross Site Scripting

What is XSS?

> Cross-site scripting (XSS) is a type of computer security vulnerability typically found in web applications. XSS enables attackers to inject client-side scripts into web pages viewed by other users

Types of XSS

- Reflected
- Persisted

Reflected

```
http://some_url.com/q=?<script>alert('boom')<script>
```

Persisted

```
<script> document.write('<body><h1>Haxored!</h1></body>') </script>

<script> window.location = 'http://download-my-game-plz.com' </script>

<script> alert('lulz BOOM!') </script>
```

I TRIED TO WARN YOU

How do we stop it?

- Escape all the things!

In react, don't use __dangerouslySetInnerHtml unless you have escaped required tags.

In ERB don't html_safe and raw() unless you have escaped all the things.

Escaping all the things still fails

MySpace XSS Worm

2005? I feel so old

But how is that possible? Aren't attackers cyber ninjas with zero days you need super firewalls and an office of security people to stop?

No, most "attacks" are incredibly stupid, send an email to 50,000 companies with a .wsf file attached that downloads an EXE from Moldova

— SwiftOnSecurity (@SwiftOnSecurity) November 19, 2016

Questions? Answers?

http://edh.getitdn.com/users - Shitty App is Shitty