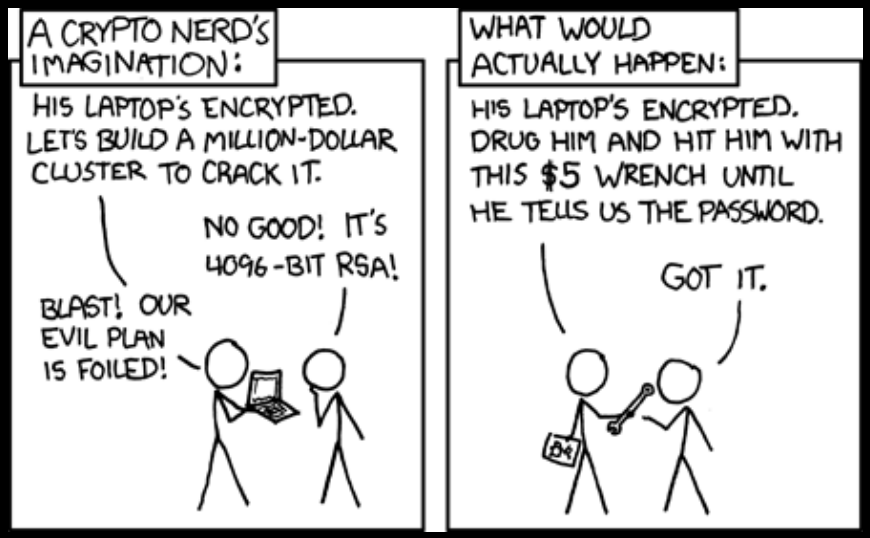
HCS Boot Camp: Week 5

Preventing Security Compromises



XSS Attacks

- “Cross-Site Scripting Attacks.”
- Non-Persistent (data provided in requests is not sanitized, hence, client-side script can be injected).
- Persistent (the server saves the data provided, and then displays it on every page).

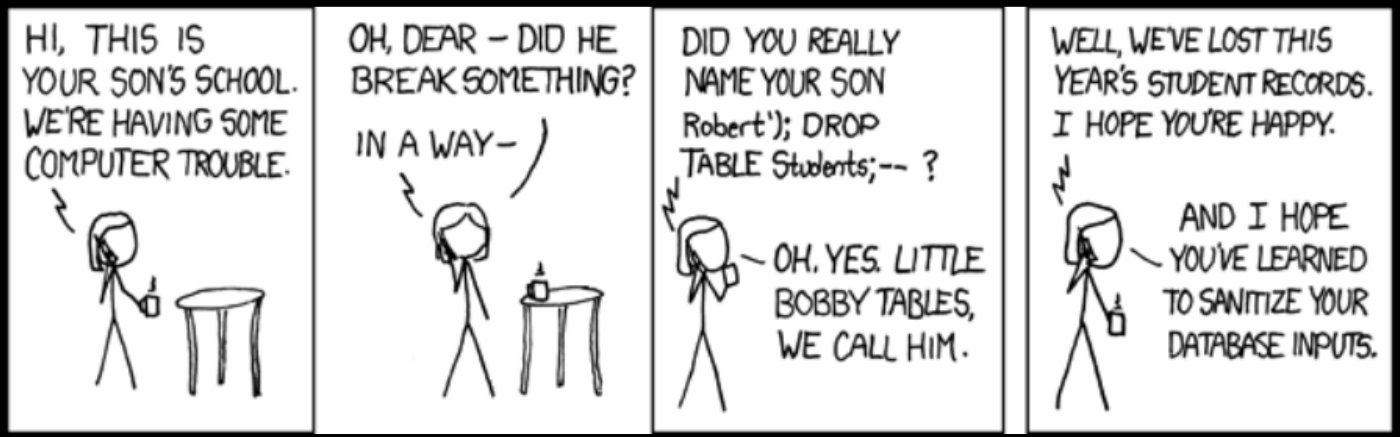
SQL Injection

- When input is not sanitized, and the input is used to execute queries directly, you can “inject” SQL into the input.
- For instance

```
“SELECT * from users where username=‘$_GET[\
“uname\”]’”
```

- What happens when:

```
$_GET [‘uname’] = “ ’; DROP TABLE users;”
```



SQL Injection

CSRF

- You are “authenticated” (via some SESSION, or a cookie) on some site.
- A Cross-Site Request Forgery is a request to the aforementioned site from a different site.
- Per the example for CSRF from Wikipedia:
  `<img src=”http://bank.exmple/withdraw?account=bob&amount=1000000&for=mallory />`

General Principles

- Update software.
- Monitor.
- *Don’t ever trust the user, or their browser/agent.*

General Principles

- Any data that is acquired from the user *should* be sanitized.
- Don’t write your own crypto.
- In general, use libraries.
- Use frameworks to avoid XSS and CSRF attacks.

Now go do this yourself!

```
git clone git://github.com/abrody/blag.git
```