EECS 388 Discussion 4

Crypto Project Review & Intro to Web Project

Length Extension

- Constructing the URL:
  - new_token = md5(state = old_token.decode(“hex”), length of padded original msg)
  - new_msg = “user=...” + padding(len(“user=...” + 8) * 8) + command3

- Original URL = http://eecs388.org/project1/api?token=b301afea7dd96db3066e631741446ca1&user=admin&command1=ListFiles&command2=NoOp

- New URL = http://eecs388.org/project1/api?token=d127a022396f60239b3d08ecc0c4e3f4&user=admin&command1=ListFiles&command2=NoOp%80%00%00%00%00%98%01%00%00%00%00%00%00&command3=DeleteAllFiles

Hash Collision

- Modify the suffix so that “I come in peace.” prints for one file and “Prepare to be destroyed!” prints for the other

- Possible solution suffix:

“””

```
if sha256(blob).hexdigest() == “...”:
    print “I come in peace.”
else:
    print “Prepare to be destroyed!”
```

Writeup

- Why does using HMAC instead of MD5 prevent length extension?

- Why are hash collisions dangerous?

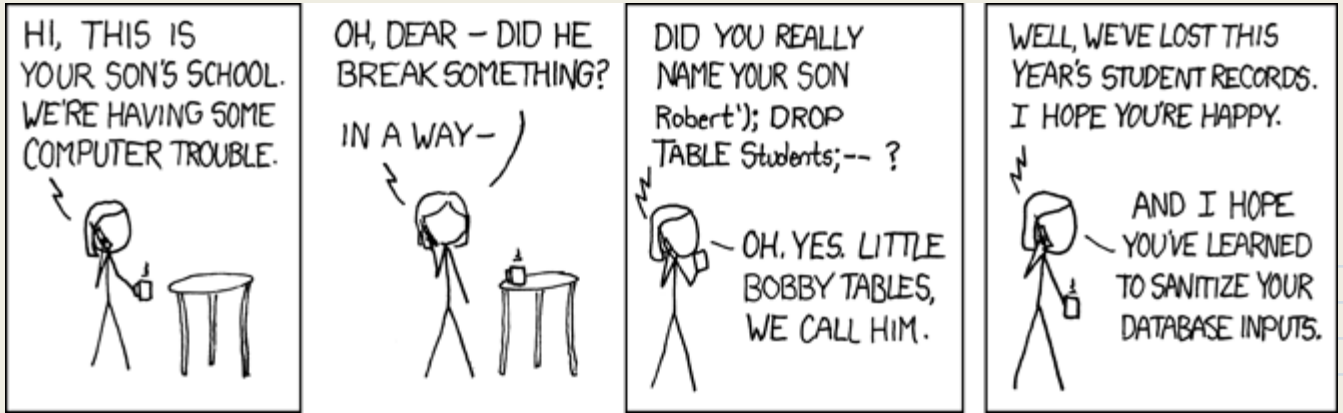
Web Project

- Introduction to various web attacks
  - SQL Injection
  - Cross-Site Request Forgery (CSRF)
  - Cross-Site Scripting (XSS)

- Learn to use HTML, Javascript

SQL

- SQL
  - Way to store and access data
  - Databases implement efficient operations on large data
- Sanitize Input!



SQL Injection Attack

- Your job:
  - Add to the SQL while getting around sanitization

  - $sql = “SELECT * FROM users WHERE username= ‘$username’ AND password=‘$password’;”

CSRF

- Send a request to a site impersonating a user

- Why is it dangerous?
  - Attacker can delete data
  - Can login another user

- How to:
  - send a post-request from same origin
  - submit a form without user recognizing a change

XSS

- Run code on a different site
- Retrieve data
  - Session Cookie
  - Record User Actions
- Change User Settings
  - Take over an account

XSS How

- Embed script into input
  - e.g. <script> … </script>

- Navigate around sanitization

HTML

- Basic syntax example:

```
<html>
    <head>
        <script type=”text/javascript” src=”//.../jquery.js”></script>
        <script type=”text/javascript”>
            //your code here
        </script>
    </head>
    <body>
        <h1> This will show up </h1>
    </body>
</html>
```

Some HTML Elements

- create a hyperlink that displays custom text and takes you to an arbitrary website:
  - your text

- html <form> is used to collect user input
  - <form action> will submit the form to the given address

```
<form action=”URL” method=“POST” target=“_self”>
<input type=“text” name=“user” >
<input type=“hidden” name=“id” value=“someval” >
<input type=“submit” value=“Submit” >
</form>
```

HTTP Requests

- Get
  - Requests data from a source
  - can be put into a URL: http://eecs388.org/project2./search?...&q=...
- Post
  - Submits data to be processed at a source
- Can be embedded into HTML forms

Javascript

- Used to make websites interactive
- Can change anything on the page
  - With some security limitations
- Recommend you use jquery to get/modify anything

Resources

- Various links covering attacks supplied in project spec
- Learn and test HTML/Javascript here: http://www.w3schools.com

Questions?