Web Security Part 2: XSS & CSRF

Summer University 2014
Prof. Tom Austin
San José State University

Code injection attacks

Last session we reviewed SQL injection attacks, where an attacker gains access to the database to steal or modify information.

Now we focus on HTML and JavaScript attacks:

- Cross-site request forgery (CSRF)
- Cross-site scripting (XSS)
- Eval injection

Crash Course on HTML

Welcome to my webpage

This is my webpage. There are many like it, but this one is mine.

```
<h1>Welcome to my webpage</h1>
<p>This is my webpage.
There are many like it,
but this one is mine.</p>
```

Dynamically Generated Pages

- Most webpages do not exist until you visit them.
- Think of a form letter or template: some general content with holes for specific details.

MAD LIBS JUNIOR

OUR CAFETERIA

Our school cafeteria has really ___sleepy___ food. Just thinking about it makes my stomach ___squash___. The spaghetti is ___red___ and tastes like ___shovels___. One day, I swear one of my meatballs started to ___rub___! The turkey tacos are totally ___squishy___ and look kind of like old ___mosquitos___. My friend Dana actually likes the meatloaf, even though it's ___funky___ and ___scary___. I call it "mystery meatloaf" and think it's really made out of ___meatballs___. My dad said he'd make my lunches, but the first day, he made me a sandwich out of ___eyelashes___ and peanut butter! I think I'd rather take my chances with the cafeteria!

Generated Webpage Example

```
<p>Welcome back,<?php echo $name;?></p>
```

Name: Tom
Password:
Submit Query

```
<p>Welcome back,Tom</p>
```

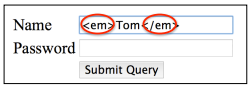
Welcome back, Tom

Code injection

- Web browsers cannot distinguish between the template, and data typed in by a user.
- If the site designer is not careful, it is possible to sneak in code.

Injecting Formatting Data

```
<p>Welcome back,<?php echo $name;?></p>
```




```
<p>Welcome back,<em>Tom</em></p>
```

Welcome back, *Tom*

Some important HTML tags

- A paragraph of text
- Specifies a block of text
- Similar to div, but not a block
- A hyperlink
- <h1>A big heading</h1>
- <h2>A smaller heading</h2>
-
- There are many more. See http://www.w3schools.com/html/html_intro.asp.

Problem 2.1

Go to http://cs31.cs.sjsu.edu/⟨**group**⟩/thanks.php.

a. In the "Hero's name" field (outlined in red), enter:
   Batman
   Hit "Submit". What happens?
b. Inject another formatting tag into the same field. What tag did you use, and what was the result?
c. Embed the image http://cs31.cs.sjsu.edu/images/35274977.jpg into the page. Describe how you did it.

Introduction to JavaScript

JavaScript

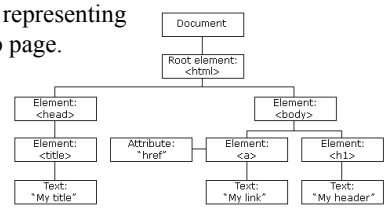
- JavaScript looks superficially similar to languages like C++ and Java
- Primarily client-side programming (i.e. code running in your browser & your machine), but some server-side variants
  - JVM: Rhino & Nashorn
  - Node.js
- http://w3schools.com/js/default.asp

JavaScript example

```
var x = 42; //vars in JavaScript
function addOne(y) {
  return y + 1;
}
var z = addOne(x);//Call function
alert("z is " + z); //pop-up box
```

The Document Object Model (DOM)

The DOM is a tree structure representing your web page.



JavaScript and the DOM

Some methods/attributes that may be helpful:

- `document.getElementById('elid')` finds the element with an attribute of "elid" and returns it.
- `alert("Some message")` opens a pop-up window displaying "Some message".
- `innerHTML` attribute of any element will change its contents.

JavaScript DOM example

```
<h1 id='title1'>Test Page</h1>
<button onclick='
  document.
    getElementsByTagName("h1")[0].
    style.color="red"'
>Make Red</button>
<button onclick='
  document.
    getElementById("title1").
    style.color="black"'
>Reset</button>
```

Javascript urls

- Some browsers permit `javascript:` urls.
  `–javascript:alert(4+3);`
- This feature has been disabled in other browsers. Why?

Problem 2.2 & 2.3

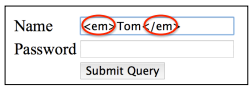
Practice JavaScript.

Do problems 2 & 3 from http://cs31.cs.sjsu.edu/labs/exercise2.pdf.

Cross-Site Scripting (XSS)

Injecting Formatting Data

```
<p>Welcome back,<?php echo $name;?></p>
```

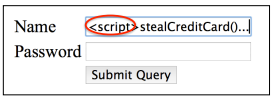



```
<p>Welcome back,<em>Tom</em></p>
```

Welcome back, *Tom*

Cross-Site Scripting (XSS)

```
<p>Welcome back,<?php echo $name;?></p>
```



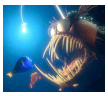

```
<p>Welcome
  back,<script>stealCreditCard()…</p>
```

Cross-Site Scripting (XSS)

Improperly sanitized input allows an attacker to inject JavaScript code as a first step to

perform a CSRF attack,

redirect the user to a phishing site,

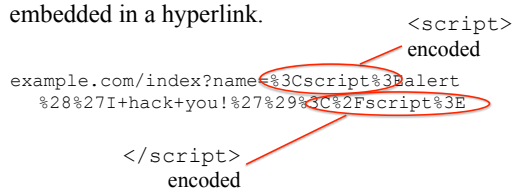


send the user to a drive-by download site,

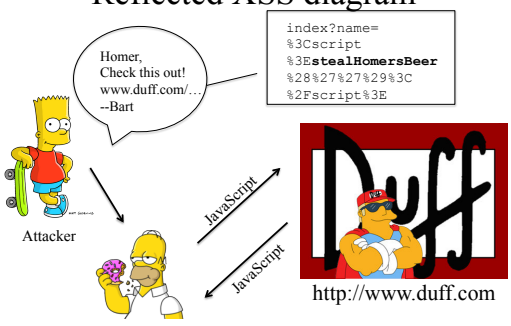
or install a keylogger.

Reflected XSS attacks

- Request carefully encodes JavaScript, which is executed in the browser on the response
- Requires interaction from the user, but may be embedded in a hyperlink.



Reflected XSS diagram



Homer Gets Pwned

To stop reflected XSS attacks, some browsers compare HTTP requests and responses.

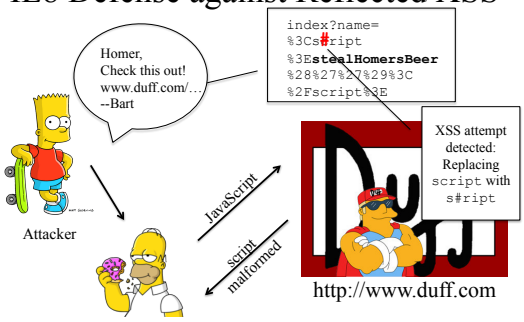
Internet Explorer uses regular expressions and maims any script tags it identifies

Chrome identifies script DOM nodes coming from requests and suppresses them.

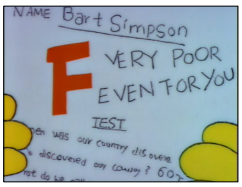
Firefox relies on addons, most notably NoScript.

For a detailed discussion, see Bates et al.'s *"Regular Expressions Considered Harmful in Client-Side XSS filters"* [2010].

IE8 Defense against Reflected XSS



Attack Fails



Problem 2.4

Go to http://cs31.cs.sjsu.edu/⟨group⟩/thanks.php. In the "Hero's name" field, enter:

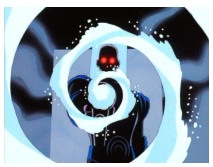
```
<script>
 alert("I am Gotham's reckoning")
</script>
```

a) Observe the url bar. Was this attack a reflected or a stored XSS attack?

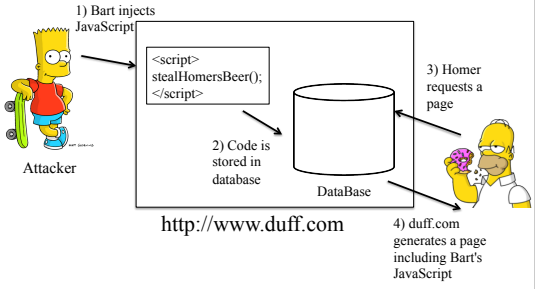
b) What is the difference in behavior between different browsers?

Stored XSS attacks

- Also known as persistent XSS attacks.
- Store JavaScipt in the website's database.
- While less common than reflected XSS, stored XSS is more powerful:
  - Evades browser-based XSS filters
  - Does not rely on user's clicking malicious links.



Stored XSS diagram



Homer Gets Pwned

Problem 2.5

Using a stored XSS attack, inject code into `thanks.php` that changes the first HTML element on the page to read "Kneel Before Zod!".

(JavaScripts `document.getElementById` and `innerHTML` features may be useful).

Note that this attack should work in all browsers, but it might require you to reload the page.

Cross-Site Request Forgery (CSRF)

What is a CSRF attack?

Pronounced "sea surf", it stands for

Cross
Site
Request
Forgery

What is a CSRF attack?

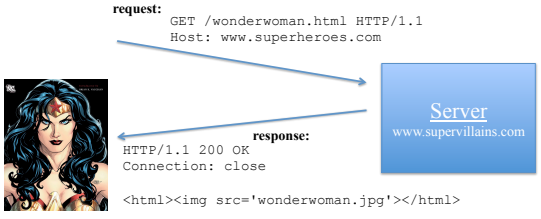
Pronounced "sea surf", it stands for

Cross Site — Generally this attack involves 2 sites

Request Forgery — A request is made, but it is not a request intended by the user

Hypertext Tranfer Protocol (HTTP)

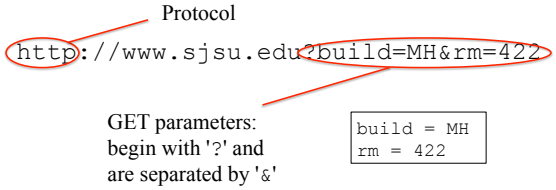
- Foundation of communication on the web
- It is a request-response protocol



GET and POST

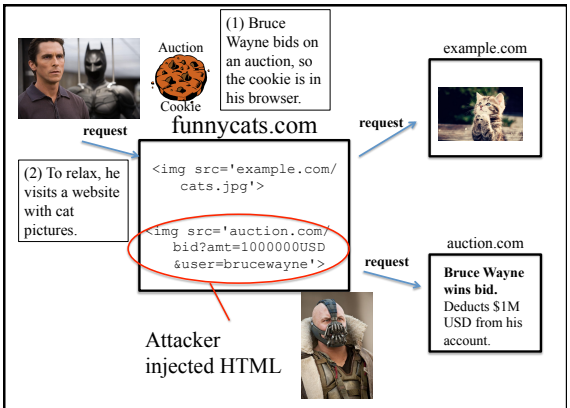
- GET and POST are the two most common request methods in HTTP
  - GET requests retrieve data
  - POST requests may modify the server state
- The distinction between GET and POST is a convention that is *usually* followed

Anatomy of a URL



HTTP Cookies

- Small pieces of data stored in a user's browser to **preserve state.**
- Sites often use cookies to keep track of their users.
- *Authentication cookies* are the most common method of determining if a user is logged in to a website. **An attacker can abuse this mechanism.**

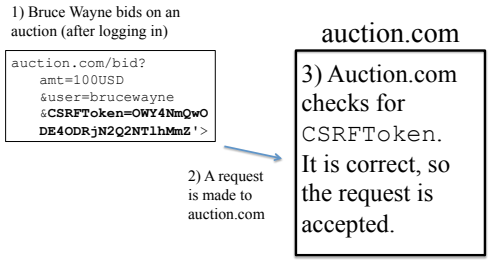


CSRF defense: Synchronizer Token Pattern

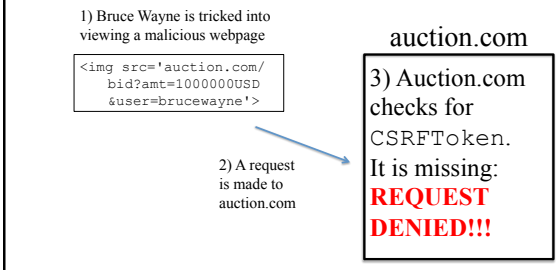
- Random challenge tokens are inserted in the page (usually as hidden form elements)
- Server verifies the tokens are correct
- https://www.owasp.org/index.php/Cross-Site_Request_Forgery_%28CSRF%29_Prevention_Cheat_Sheet

Synchronizer Token Example

Synchronizer Token Example



Synchronizer Token Example



Problem 2.6

See problem 6 of http://cs31.cs.sjsu.edu/labs/exercise2.pdf.

A banking site is available at http://cs31.cs.sjsu.edu/bank/index.php.

Eval injection

Eval is evil. Avoid it.
Eval has aliases. Don't use them.
--Douglas Crockford

`eval`: playing with fire

- Many languages have an `eval` construct to execute code stored in a string.
- `eval` is a powerful construct, allowing developers to do many things they cannot otherwise do.
- `eval` is also dangerous and horribly, horribly abused.
- Richards et al., "The Eval that Men Do"

Eval example

```
var x = 7; // x is initially 7
var s = "x = x - 4"; // a string
eval(s); // evaluates s as
         // if it were code
alert(x); // displays 3
```

JSON: A necessary eval?

JavaScript Object Notation is a common, concise way of transferring data.

```
{ firstName: "Tyrion",
  lastName: "Lannister",
  age: 32,
  friends: [
    { name:"Bronn", role:"sellsword" },
    { name:"Podrick", role:"page" }]
}
```

Parsing JSON

Today there is `JSON.parse`, but before that was standard, `eval` served as a quick and dirty way to parse JSON.

```
var jsonData = "{acc:435921," +
    "withdraw:20, denom: 'USD'}";
eval("var transaction = " +
    jsonData);
alert("Amount of withdrawal: " +
    transaction.withdraw);
```

JSON transactions

```
[ {acc:53491, withdraw:40},
  {acc:42917, withdraw:120},
  {acc:(function(){
     alert('Mwa, ha, ha!!!');
     return 30211;
    })()}
]
```

Problems 2.7 and 2.8

These problems are a little more challenging.

For details, see problems 6 and 7 of http://cs31.cs.sjsu.edu/labs/exercise2.pdf.

Do these only after you have finished the rest of the exercises.