CS 142 Review Session

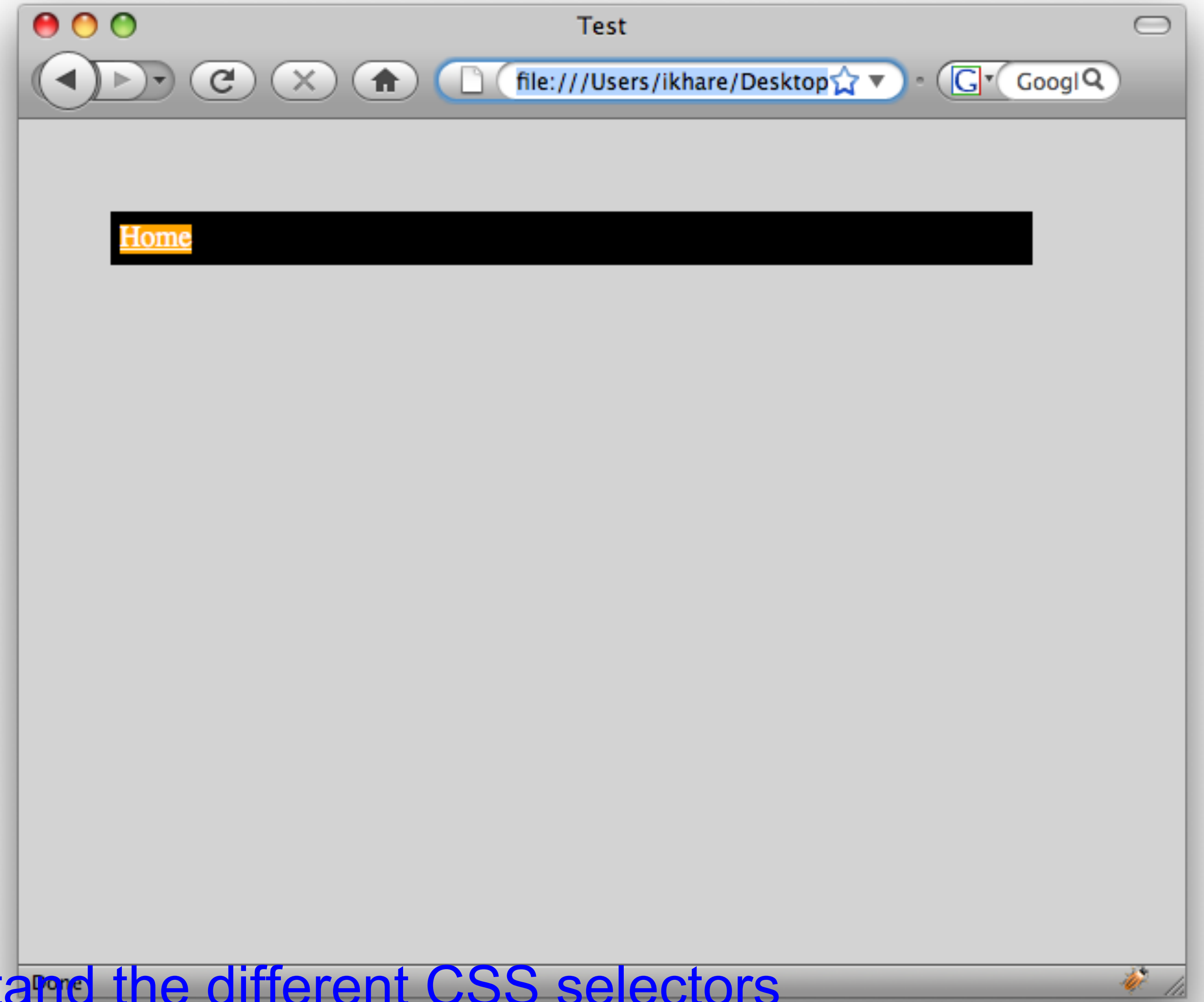
Gary Luu, Troy Brant, Tom Wang, Indrajit Khare

HTML & CSS - What does this look like?

```
<body>
    <div id="content">
        <span class="menuItem"><a
href="home.html">Home</a></span>
    </div>
</body>
```

```
body
{
  background: lightgray;
  margin: 50px;
  padding: 0px;
}

#content
{
  width: 500px;
  background: black;
}

span
{
  display: block;
}

.menuItem
{
  padding: 5px;
}

.menuItem a
{
  color: white;
  background: orange;
}
```

HTML & CSS - What does this look like?



Understand the different CSS selectors

setTimeout - When is the alert fired?

```
function foo()
{
    alert("foo called!");
}

setTimeout(foo(), 1000);
setTimeout("foo()", 1000);
setTimeout(foo, 1000);
```

setTimeout - When is the alert fired?

```
function foo()
{
    alert("foo called!");
}

setTimeout(foo(), 1000); //fired immediately
setTimeout("foo()", 1000); //the string is evaluated as
                         //javascript 1 sec later, which calls the function
setTimeout(foo, 1000); //the function is called 1 sec later
```

setTimeout & setting event handlers

When are are the alerts fired?

```
function foo()
{
    var img = new Image();

    img.onerror = function() {
        alert("foo load error");
    };

    setTimeout(function() {
        alert("foo setTimeout");
    }, 1000);

    img.src = "bar.html";
}
```

setTimeout & setting event handlers

When are are the alerts fired?

```
function foo()
{
    var img = new Image();

    img.onerror = function() {//remember this is non-blocking!
        alert("foo load error"); //when the image is loaded and
                //the browser realizes it's not an image
    };

    setTimeout(function() {//remember this is non-blocking!
        alert("foo setTimeout");//after 1 second
    }, 1000);

    img.src = "bar.html";
}
```

Same-Origin Policy

- http://code.google.com/p/browsersec/wiki/Main
- High-level way to think about SOP
  - Rule of Thumb: If it allows something insecure, SOP blocks it.
- For scripting frames: Scheme, domain, port
- For cookies: Scheme, domain, path

Same-Origin Policy Questions

- Why is path useless for cookie security?

- What happens when scheme is not part of SOP?
  - Setup: http://vulnerable.com contains iframe to https://vulnerable.com.

Same-Origin Policy Questions

- Why is path useless for cookie security?
  - Same-origin policy for scripting frames means if a.com/pathA contains an iframe to a.com/pathB, a.com/pathA can access a.com/pathB's cookie through the iframe, even if cookie is supposed to be read only by a.com/pathB.
- What happens when scheme is not part of SOP?
  - Setup: http://vulnerable.com contains iframe to https://vulnerable.com.
  - If scheme is not part of SOP, the page http://vulnerable.com can script https://vulnerable.com. Attacker can change content to http://vulnerable.com to script the iframe for https://vulnerable.com.

Sample Question

In class we discussed the following PHP script for a login page:

```
$username = $_GET[user];
$password = $_GET[pwd];
$sql = "SELECT *
FROM usertable
WHERE username = ’$username’
AND password = ’$password’ ";
$result = $db->query($sql);
if ($result->num_rows > 0) { /* Success */ }
else { /* Failure */ }
```

(a) (2 points) Explain why a URL where user is set to " ’ or 1 = 1 -- " will result in a successful login.

Sample Question

In class we discussed the following PHP script for a login page:

```
$username = $_GET[user];
$password = $_GET[pwd];
$sql = "SELECT *
FROM usertable
WHERE username = ’$username’
AND password = ’$password’ ";
$result = $db->query($sql);
if ($result->num_rows > 0) { /* Success */ }
else { /* Failure */ }
```

(a) (2 points) Explain why a URL where user is set to " ’ or 1 = 1 -- " will
result in a successful login.

Resulting query "SELECT * FROM usertable WHERE username = '' or 1 = 1 -- ..." will always return the entire usertable, since 1 = 1 is always true.

Sample Question

(b) (2 points) Suppose we change lines 1 and 2 to
$username = addslashes($ GET[user])
$password = addslashes($ GET[pwd])
Recall that the addslashes function adds a slash before every quote. That is addslashes("a'b") will output the string "a\'b". Explain why this prevents the attack from part (a).

Sample Question

(b) (2 points) Suppose we change lines 1 and 2 to
$username = addslashes($ GET[user])
$password = addslashes($ GET[pwd])
Recall that the addslashes function adds a slash before every quote. That is
addslashes("a'b") will output the string "a\'b". Explain why this prevents
the attack from part (a).
Resulting query becomes "SELECT * FROM usertable WHERE username = '\' or 1 = 1 -- ' AND password = ...". All the input from part (a) is now the value of username. None of it affects the structure of the SQL statement.

Sample Question

(c) (9 points) Does addslashes completely solve the problem? Consider the GBK Chinese unicode character set. Some characters in GBK are single bytes while others are double bytes. In particular, the following table shows a few GBK characters:

0x 5c = \
0x 27 = ’
0x bf 27 = ¿’
0x bf 5c = 縗

That is, the database interprets 0xbf27 as two characters, but interprets 0xbf5c as a single chinese character.

Show that using addslashes as in part (b) leads to a SQL injection attack. What value of user will result in a successful login?

Sample Question

(c) (9 points) Does addslashes completely solve the problem? Consider the GBK Chinese unicode character set. Some characters in GBK are single bytes while others are double bytes. In particular, the following table shows a few GBK characters:

0x 5c	= \
0x 27	= ’
0x bf 27	= ¿’
0x bf 5c	= 縗

That is, the database interprets 0xbf27 as two characters, but interprets 0xbf5c as a single chinese character.
Show that using addslashes as in part (b) leads to a SQL injection attack. What value of user will result in a successful login?

If attacker inputs ¿' or 1=1, when the slash is added, the codes for the first three characters (¿\') becomes 0xbf5c27. The database interprets 0xbf5c as the Chinese character 縗, so the resulting string is interpreted as 縗’ or 1=1. The quote is no longer escaped.

Sample Question

(d) (2 points) How should addslashes be implemented to defend against your attack from part (c)?

Sample Question

(d) (2 points) How should addslashes be implemented to defend against your attack
from part (c)?
One way to fix addslashes is to call addslashes recursively after addslashes made any changes to the string. In the previous example, after the call to the first addslashes results in 縗' or 1=1, addslashes would be called again, this time modifying the string into 縗¥' or 1=1. So we no longer have problem with this attack. Unfortunately, this solution modifies anything with ¿', which should not be a big problem since ¿ is rare.

Disclaimer: I do not know if the solutions here would give you full credit for the problem, as I don't have the answer key to any of the previous CS 155 finals.

Other Security Problems

- XSS
  - JavaScript is executed in the context of the vulnerable page.
  - Consequences: Take over account with session ID, read or modify vulnerable page, ...
  - Prevention: Proper sanitization of user input

Other Security Problems

- CSRF
  - Form is submitted from across domain. Takes advantage of victim's session cookie to submit form as the victim.
  - Consequences: Depends on form
  - Prevention #1: Submit session ID as part of hidden input
    - Why not? This works against CSRF. However, we should avoid possibly leaking the session ID when user e-mails page or if it's in browser history. Essentially defense-in-depth.
  - Prevention #2: secret token (as part of hidden input)

Other Security Problems

- Session fixation
  - Attacker gets a session ID and sets the victim's session ID to his session ID
  - Consequences: Take over account
  - Prevention: Reset session ID after login
- Know how these attacks work!

Bob wants to display a list of employees for his internal HR app, but his list of employees is blank. What's wrong with his code? (assume @employees is populated correctly and both the model and controller are fine)

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.
org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
    <title>Khare Korp - Employee Directory</title>
</head>
<body>
    <h2>Employees:</h2>
    <ul>
    <% @employees.each do |employee| %>
        <li><% link_to employee.last_name + ", " + employee.first_name,
:action => "employee", :id => employee.id %></li>
    <% end %>
    </ul>
</body>
</html>
```

Bob wants to display a list of employees for his internal HR app, but his list of employees is blank. What's wrong with his code? (assume @employees is populated correctly and both the model and controller are fine)

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.
org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
    <title>Khare Korp - Employee Directory</title>
</head>
<body>
    <h2>Employees:</h2>
    <ul>
    <% @employees.each do |employee| %>
        <li><% link_to employee.last_name + ", " + employee.first_name,
:action => "employee", :id => employee.id %></li>
    <% end %>
    </ul>
</body>
</html>
```

A: <%... should be <%=...

Same code, find the security vulnerability:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.
org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
    <title>Khare Korp - Employee Directory</title>
</head>
<body>
    <h2>Employees:</h2>
    <ul>
    <% @employees.each do |employee| %>
        <li><%= link_to employee.last_name + ", " + employee.first_name,
:action => "employee", :id => employee.id %></li>
    <% end %>
    </ul>
</body>
</html>
```

Same code, find the security vulnerability:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.
org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
    <title>Khare Korp - Employee Directory</title>
</head>
<body>
    <h2>Employees:</h2>
    <ul>
    <% @employees.each do |employee| %>
        <li><%= link_to employee.last_name + ", " + employee.first_name,
:action => "employee", :id => employee.id %></li>
    <% end %>
    </ul>
</body>
</html>
```

A: h() is missing

Ok, so we needed to add an h() call to make it more secure. So Bob changes some controller code from this:

```
@employee = Employee.find(params[:id])
@employee.address = params[:address]
@employee.save()
```

to this:

```
@employee = Employee.find(params[:id])
@employee.address = h(params[:address])
@employee.save()
```

Is the second version more secure than the first? Why or why not?

Ok, so we needed to add an h() call to make it more secure. So Bob changes some controller code from this:

```
@employee = Employee.find(params[:id])
@employee.address = params[:address]
@employee.save()
```

to this:

```
@employee = Employee.find(params[:id])
@employee.address = h(params[:address])
@employee.save()
```

Is the second version more secure than the first? Why or why not?

A: in general, no, not more secure. h() just sanitizes user-generated text when you want to redisplay that text. It keeps arbitrary javascript from being executed by URL encoding the text so it isn't executed as code. There's no real advantage to saving the h()'ed version of the text to the database when you should ALWAYS be using h() to display the text in the view anyway.

Rails MVC

Do these belong in a model, view, or a controller?

1) User.find(1)

2) belongs_to

3) form_for

4) A function that loops through every pixel in an image and inverts the color value

5) validates_presence_of

Rails MVC

Do these belong in a model, view, or a controller?

1) User.find(1)

2) belongs_to

3) form_for

4) A function that loops through every pixel in an image and inverts the color value

5) validates_presence_of

A: 1) controller
2) model
3) view
4) controller (or better yet, the helper function file)
5) model

Migrations & Models

Currently Database:

User:
    id
    firstName
    lastName

Photo:
    id
    user_id
    fileName

Want to add comments:

- Must be associated with photo
- Has a "comment" text field

Current Model files

```
class User < ActiveRecord::Base
  has_many :photos

end

class Photo < ActiveRecord::Base
  belongs_to :user
end
```

Migrations & Models

Generate a migration and model through the script:

```
script/generate model Comment photo_id:integer comment:text
```

This script generates the migration file below as well as comment.rb in app/models/

Remember the model name must be singular!
You have to add: "belongs_to :photo" in comment.rb
You have to add: "has_many :comments" to photo.rb

```
class CreateComments < ActiveRecord::Migration
  def self.up
    create_table :comments do |t|
      t.integer :id
      t.integer :photo_id
      t.text :comment
    end
  end

  def self.down
    drop_table :comments
  end
end
```