Web Security

Thierry Sans

Same origin policy

➡ **Ressources must come from the same domain (protocol, host, port)**

Elements under control of the same-origin policy

- Ajax requests
- Form actions

Elements **not** under control of the same-origin policy

- Javascript scripts
- CSS
- Images, video, sound
- Plugins

Examples

	client	server
same protocol, port and host	http://example.com	http://example.com
	http://user:pass@example.com	http://example.com
top-level domain	http://example.com	http://example.org
host	http://example.com	http://other.com
sub-host	http://www.example.com	http://example.com
sub-host	http://example.com	http://www.example.com
port	http://example.com:3000	http://example.com
protocol	http://example.com	https://example.com

Relaxing the same-origin policy

- Switch to the superdomain with javascript
  `www.example.com` can be relaxed to **`example.com`**
- iframe
- Cross-document sharing
- JSONP

Attacks

- SQL injection
- Content Spoofing
- Cross-Site Scripting
- Cross-site Request forgery

SQL Injection

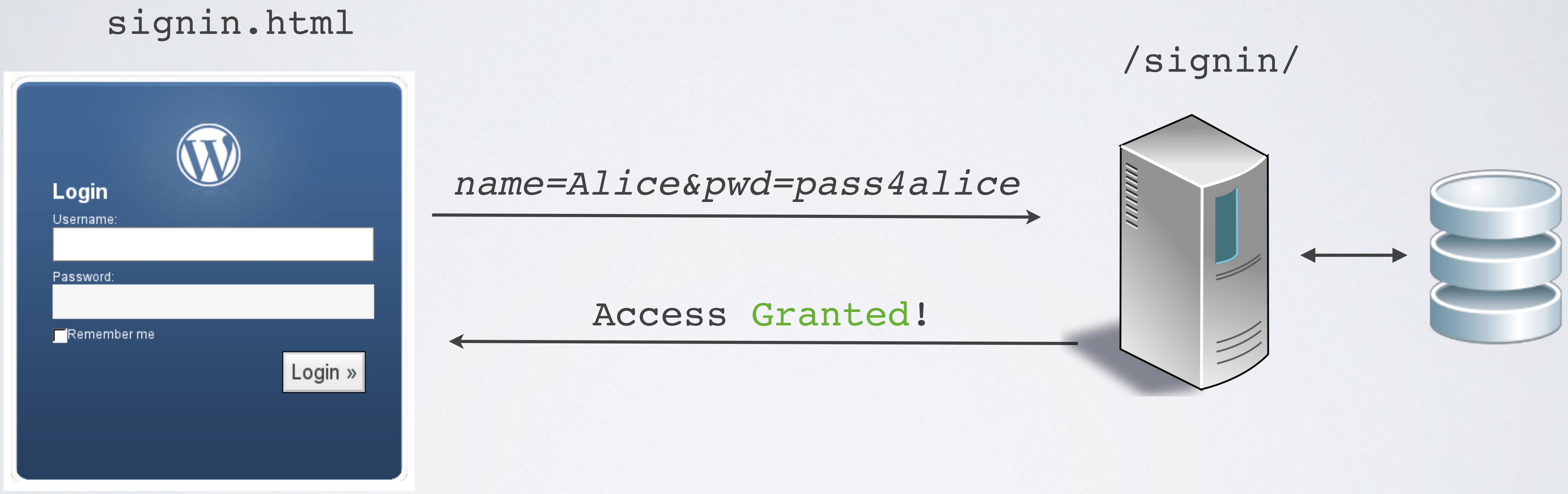
Problem

➡ An attacker can inject SQL/NoSQL code

◉ Retrieve, add, modify, delete information

◉ Bypass authentication

Checking password



SQL Injection

```
db.run("SELECT * FROM users
WHERE USERNAME = '" + username + "'
  AND PASSWORD = '" + password + "'"
```

```
username: alice
password: pas[illegible]lice
```

```
blah' OR '1'='1
```

NoSQL Injection

```
db.find({  username: username,
           password: password   });
```

```
username: alice
password: pas    lice
```

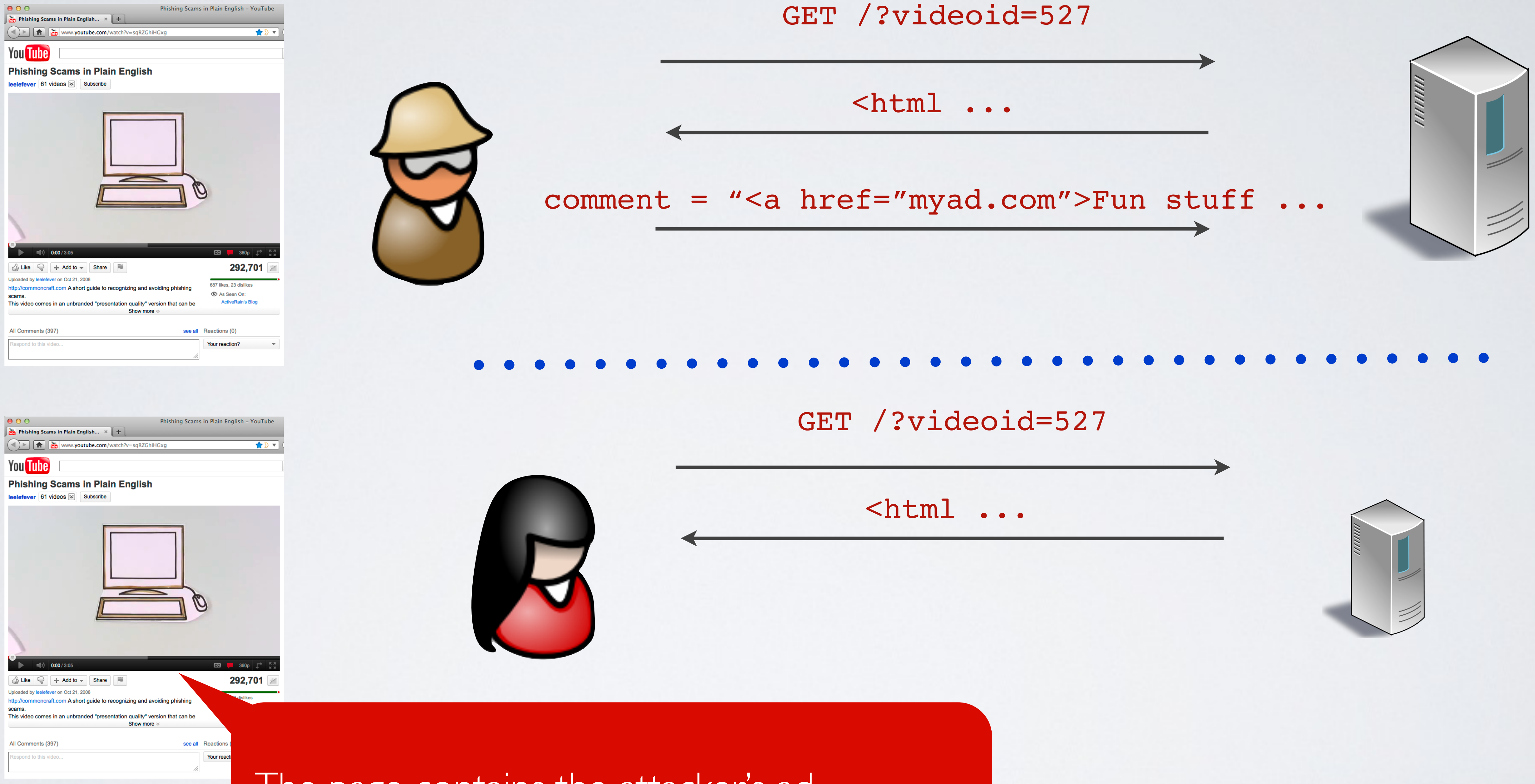
```
{gt: ""}
```

Generic Solution

- ✓ SQL - use a query API
- ✓ SQL/NoSQL - validate inputs

Content Spoofing

Content Spoofing


Phishing Scams in Plain English
GET /?videoid=527
<html ...
comment = “<a href=”myad.com”>Fun stuff ...
GET /?videoid=527
<html ...

The page contains the attacker’s ad.

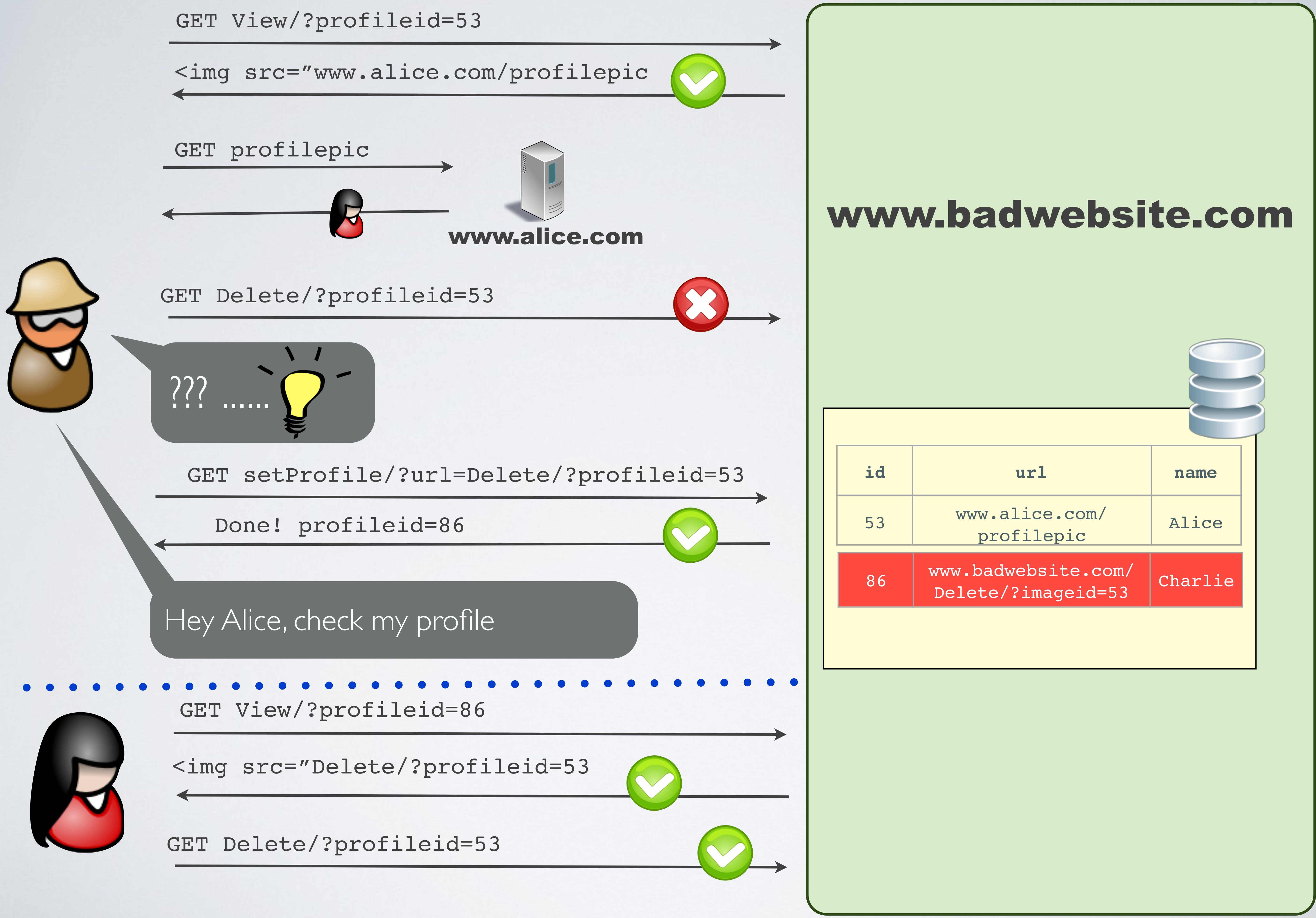
* *Notice that Youtube is **not** vulnerable to this attack*

Problem

➡ An attacker can inject HTML tags in the page

◉ Add illegitimate content to the webpage (ads most of the time)

Cross-Site Request Forgery



id	url	name
53	www.alice.com/ profilepic	Alice
86	www.badwebsite.com/ Delete/?imageid=53	Charlie

Problem

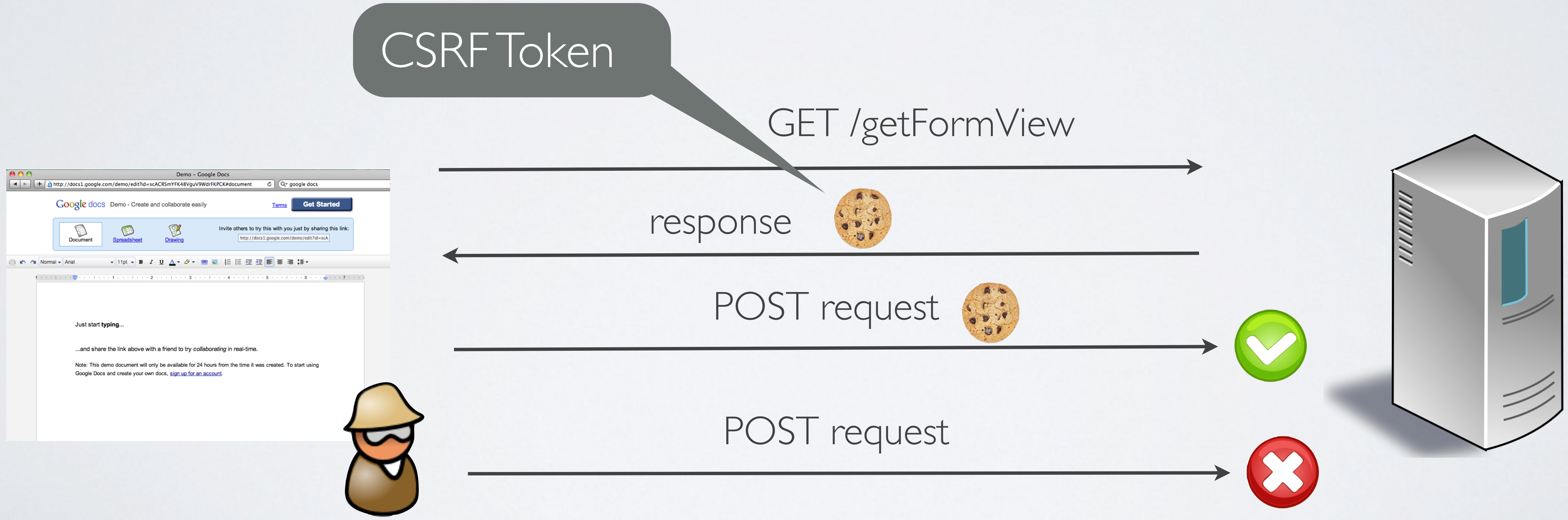
- ➡ An attacker can call do HTTP request by injecting url-based HTML tags in the page that the browser will retrieve automatically

- ◉ Inject an image content

- ◉ Insert any HTML content for which the CSS image background can be defined

Generic Solution

✓ Protect legitimate requests with a CSRF token


CSRF Token
GET /getFormView
response
POST request
POST request
Demo – Google Docs
http://docs1.google.com/demo/edit?id=scACRSmYFK48VguV9WdrFKPCK#document
google docs
Google docs Demo - Create and collaborate easily
Terms
Get Started
Document
Spreadsheet
Drawing
Invite others to try this with you just by sharing this link:
http://docs1.google.com/demo/edit?id=scA
Normal
Arial
11pt

Just start **typing**...

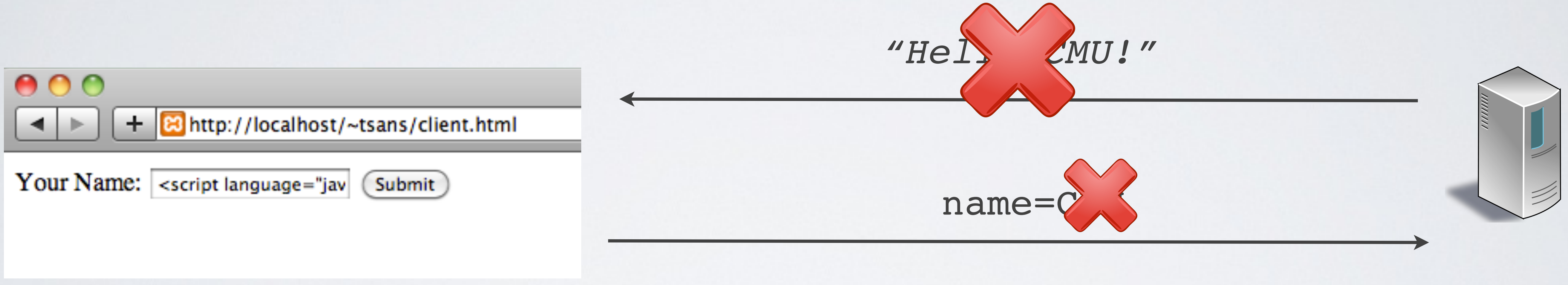
...and share the link above with a friend to try *collaborating* in real-time.

Note: This demo document will only be available for 24 hours from the time it was created. To start using Google Docs and create your own docs, sign up for an account.

Cross-Site Scripting (XSS)

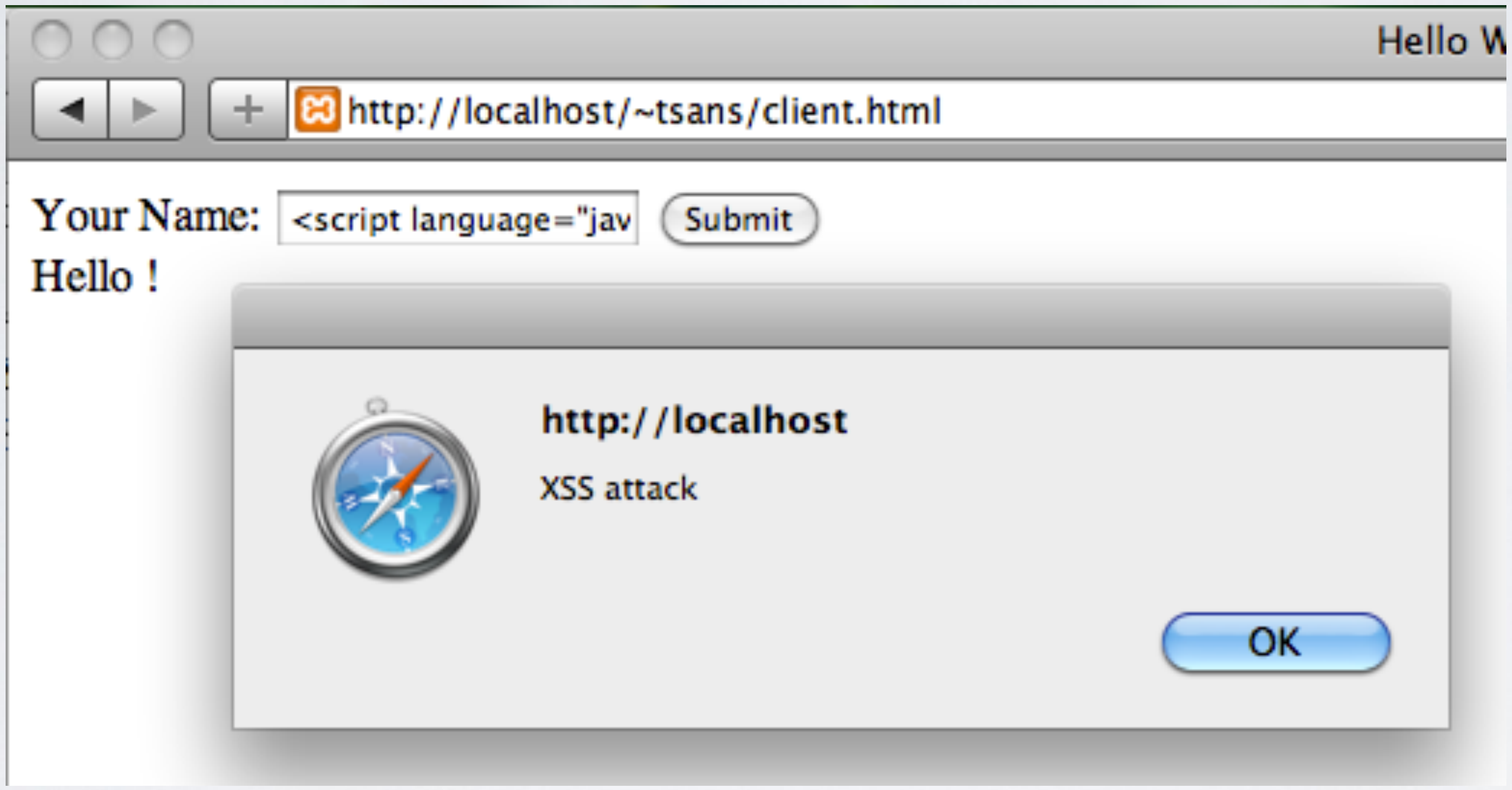
Cross-Site Scripting Attack (XSS attack)

`“Hello <script language="javascript">`**`alert(“XSS attack”);`**`</script>!”`



`name=<script language="javascript">`**`alert(“XSS attack”);`**`</script>`

XSS Attack = Javascript Code Injection



Problem

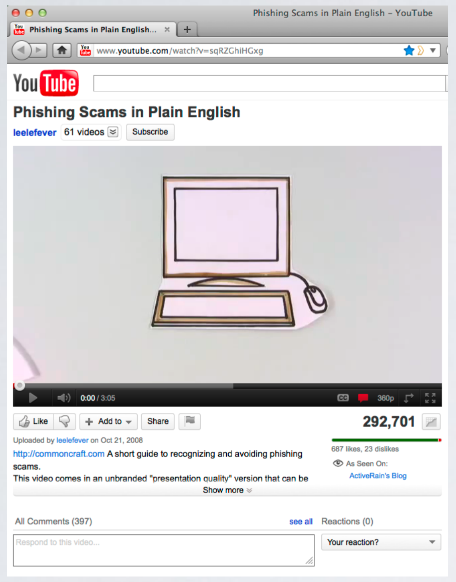
- An attacker can inject **arbitrary javascript code** in the page that will be executed by the browser

- **Inject illegitimate content** in the page (same as content spoofing)
- **Perform illegitimate HTTP requests** through Ajax (same as a CSRF attack)
- **Steal Session ID** from the cookie
- **Steal user's login/password** by modifying the page to forge a perfect scam

Solution

✓ Sanitize "tainted" output data
i.e data made from input data

Forging a perfect scam



Phishing Scams in Plain English – YouTube

Phishing Scams in Plain English...

www.youtube.com/watch?v=sqRZGhiHGxg

Phishing Scams in Plain English

leelefever 61 videos Subscribe

0:00 / 3:05 360p

Like Add to Share 292,701

Uploaded by leelefever on Oct 21, 2008

http://commoncraft.com A short guide to recognizing and avoiding phishing scams.

This video comes in an unbranded "presentation quality" version that can be

Show more

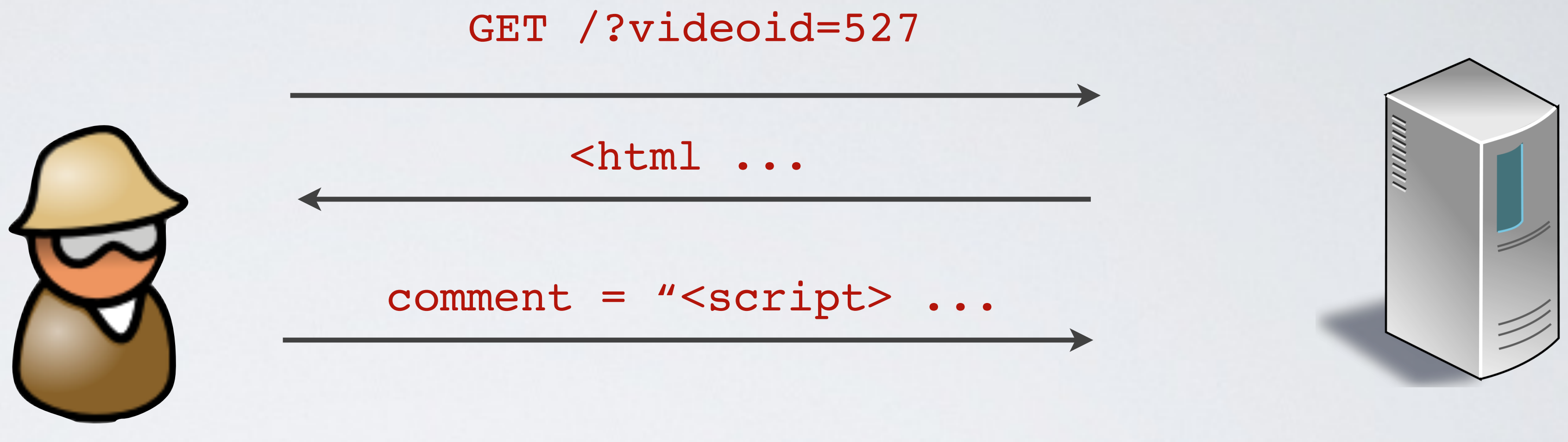
687 likes, 23 dislikes

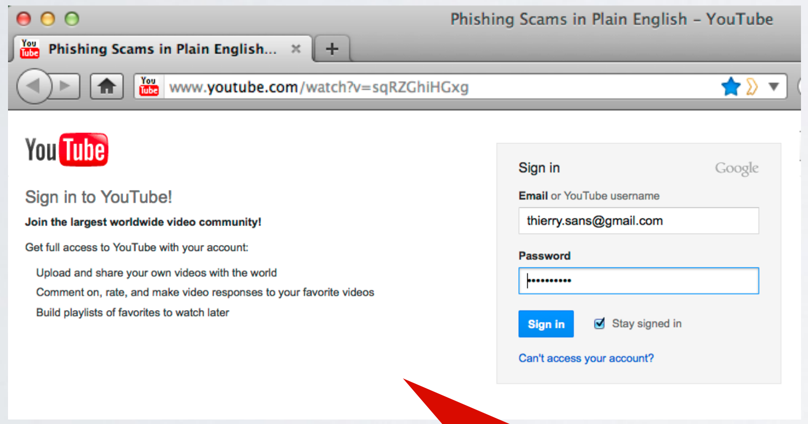
As Seen On: ActiveRain's Blog

All Comments (397) see all Reactions (0)

Respond to this video...

Your reaction?





Phishing Scams in Plain English – YouTube

Phishing Scams in Plain English...

www.youtube.com/watch?v=sqRZGhiHGxg

Sign in to YouTube!

Join the largest worldwide video community!

Get full access to YouTube with your account:

Upload and share your own videos with the world
Comment on, rate, and make video responses to your favorite videos
Build playlists of favorites to watch later

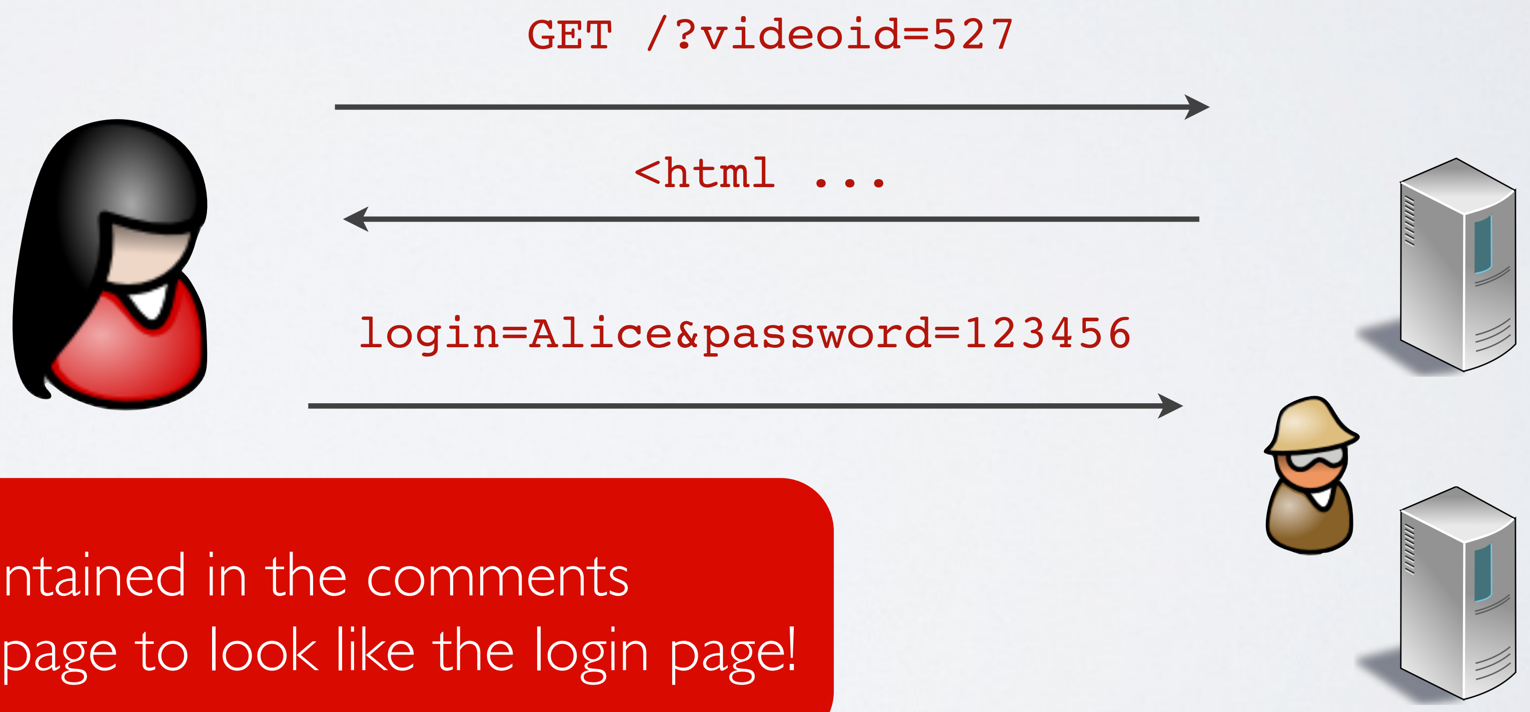
Sign in Google

Email or YouTube username: thierry.sans@gmail.com

Password: ••••••••••

Sign in [✓ Stay signed in]

Can't access your account?



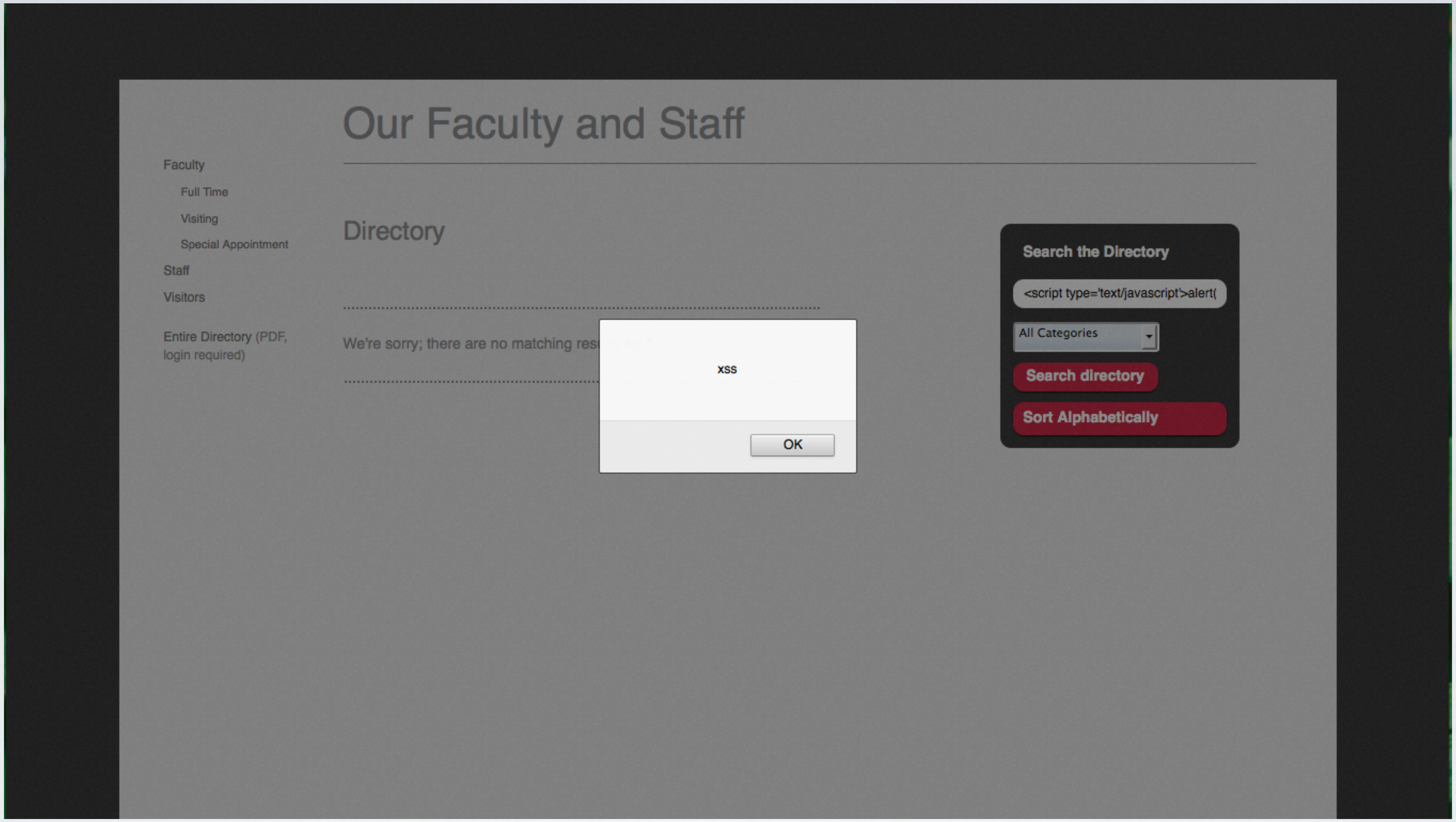
The script contained in the comments modifies the page to look like the login page!

* *Notice that Youtube is **not** vulnerable to this attack*

It gets worst - XSS Worms

Spread on social networks

- Samy targeting MySpace (2005)
- JTV.worm targeting Justin.tv (2008)
- Twitter worm targeting Twitter (2010)


Our Faculty and Staff

Faculty
Full Time
Visiting
Special Appointment
Staff
Visitors

Entire Directory (PDF, login required)

Directory

We're sorry; there are no matching res

xss

OK

XSS attacks are widespread

Generic solution for injection-based vulnerabilities

- ✓ Always escape tainted data i.e. data that comes from (or derived from) user inputs