

Web Security

Thierry Sans

|99|

Sir Tim Berners-Lee



← → C ⌂ info.cern.ch/hypertext/WWW/TheProject.html ⭐

World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#) , [Policy](#) , November's [W3 news](#) , [Frequently Asked Questions](#) .

[What's out there?](#) Pointers to the world's online information, [subjects](#) , [W3 servers](#), etc.

[Help](#) on the browser you are using

[Software Products](#) A list of W3 project components and their current state. (e.g. [Line Mode](#) ,[X11 Viola](#) ,[NeXTStep](#) ,[Servers](#) ,[Tools](#) ,[Mail robot](#) ,[Library](#))

[Technical](#) Details of protocols, formats, program internals etc

[Bibliography](#) Paper documentation on W3 and references.

[People](#) A list of some people involved in the project.

[History](#) A summary of the history of the project.

[How can I help ?](#) If you would like to support the web..

[Getting code](#) Getting the code by [anonymous FTP](#) , etc.

2014

Web Portals



Accounting and Billing



E-Learning



Publishing



Content Management

Instagram icon



E-Health



Collaboration



Social Networks



Web security is a major concern

Facebook closes hole that let spammers auto-post to walls, friends

Social-networking site plugs a second hole that allowed spammers to automatically post to people's pages.

by **Elinor Mills**  @elinormills / September 7, 2010 12:37 PM PDT / Updated: September 7, 2010 4:00 PM PDT

CNET › Security › GhostShell claims breach of 1.6M accounts at FBI, NASA, and more

GhostShell claims breach of 1.6M accounts at FBI, NASA, and more

The hacktivist group says it obtained the records via SQL injection at government sites.

by **Casey Newton**  @CaseyNewton / December 10, 2012 3:13 PM PST / Updated: December 10, 2012 3:19 PM PST

Yahoo Mail hijacking exploit selling for \$700

XSS vulnerability allows attacks to steal and replace tracking cookies, as well as read and send e-mail from a victim's account.

by Steven Musil  @stevenmusil / November 26, 2012 6:02 PM PST / Updated: November 27, 2012 3:32 PM PST

Researchers point out holes in McAfee's Web site

McAfee says it is working to fix three holes researchers found in its Web site.

by Elinor Mills  @elinormills / March 28, 2011 7:28 PM PDT

Cross Site Scripting in download.mcafee.com. "In a worst case scenario this vulnerability could allow attacks that spoof the McAfee brand by presenting a URL that looks like it directs to a McAfee Web site but in fact directs elsewhere."

Researchers find security holes in NYT, YouTube, ING, MetaFilter sites

Attackers could have used vulnerabilities on several Web sites to compromise people's accounts, allowing them to steal money, harvest e-mail addresses, or pose as others online.

by Elinor Mills  @elinormills / October 2, 2008 1:02 PM PDT / Updated: October 2, 2008 2:31 PM PDT

The vulnerability arises from a coding flaw that could allow someone to do a cross-site request forgery (CSRF) attack in which a "malicious Web site causes a user's Web browser to perform an unwanted action on a trusted site," according to the report.

Researcher finds serious Android Market bug

Google applies technical fix to bug, but Jon Oberheide says Android Market should be alerting phone owners when an app is being remotely downloaded via the Web site.

Oberheide described the XSS vulnerability as "low-hanging fruit" and said he was surprised no one had discovered it before. Such bugs are very common in Web sites.

Twitter hit by multiple variants of XSS worm

Summary: During the weekend and early Monday, at least four separate variants of the original StalkDaily.com XSS worm hit the popular micro-blogging site Twitter, automatically hijacking accounts and advertising the author's web site by posting tweets on behalf of the account holders, by exploiting cross site scripting flaws at the site.



By Dancho Danchev for Zero Day | April 14, 2009 -- 02:19 GMT (03:19 BST)

 Follow @danchodanchev

New security holes found in D-Link router

Security researcher reveals multiple Web-based security vulnerabilities in the D-Link 2760N.

by Seth Rosenblatt  @sethr / November 11, 2013 12:54 PM PST / Updated: November 12, 2013 4:54 PM PST



A new spate of vulnerabilities have been found in a D-Link router, a security researcher said Monday.

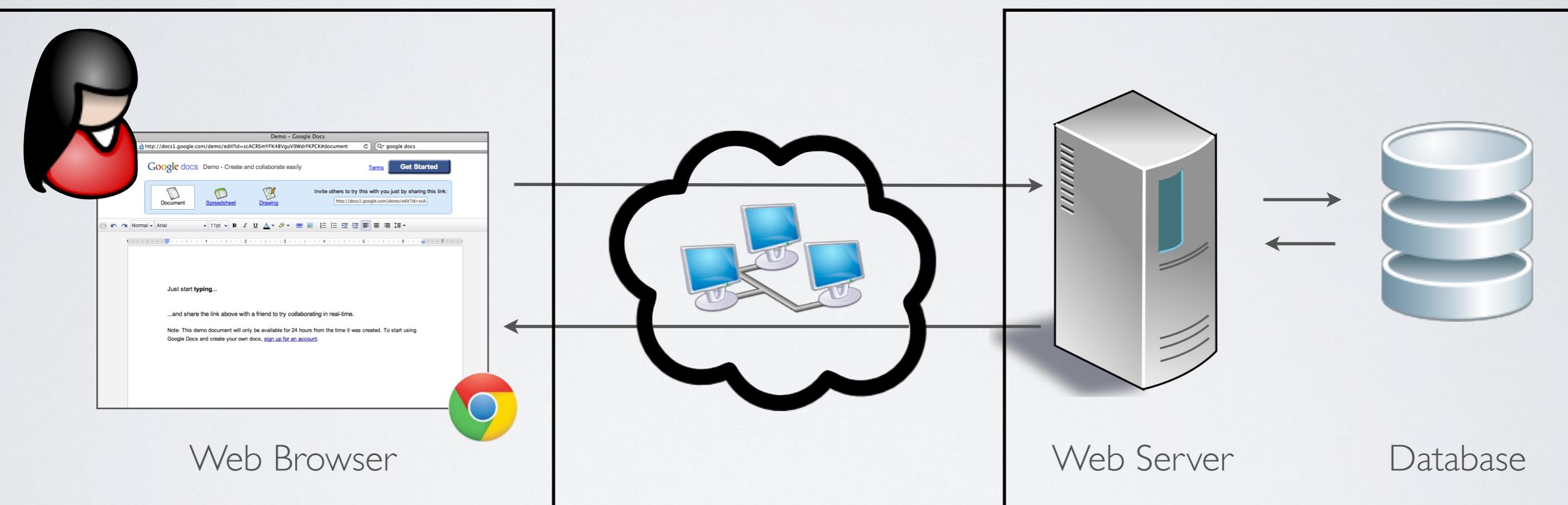
The D-Link 2760N, also known as the D-Link DSL-2760U-BN, is susceptible to **several cross-site scripting (XSS) bugs** through its Web interface, **reported ThreatPost**.

The Big Picture

The web architecture

Client Side

Server Side



Securing the web architecture means securing ...

- The network
- The DNS (Domain Name System)
- The web server operating system
- The web server application (*Apache* for instance)
- The database application (*Oracle* for instance)
- The web application

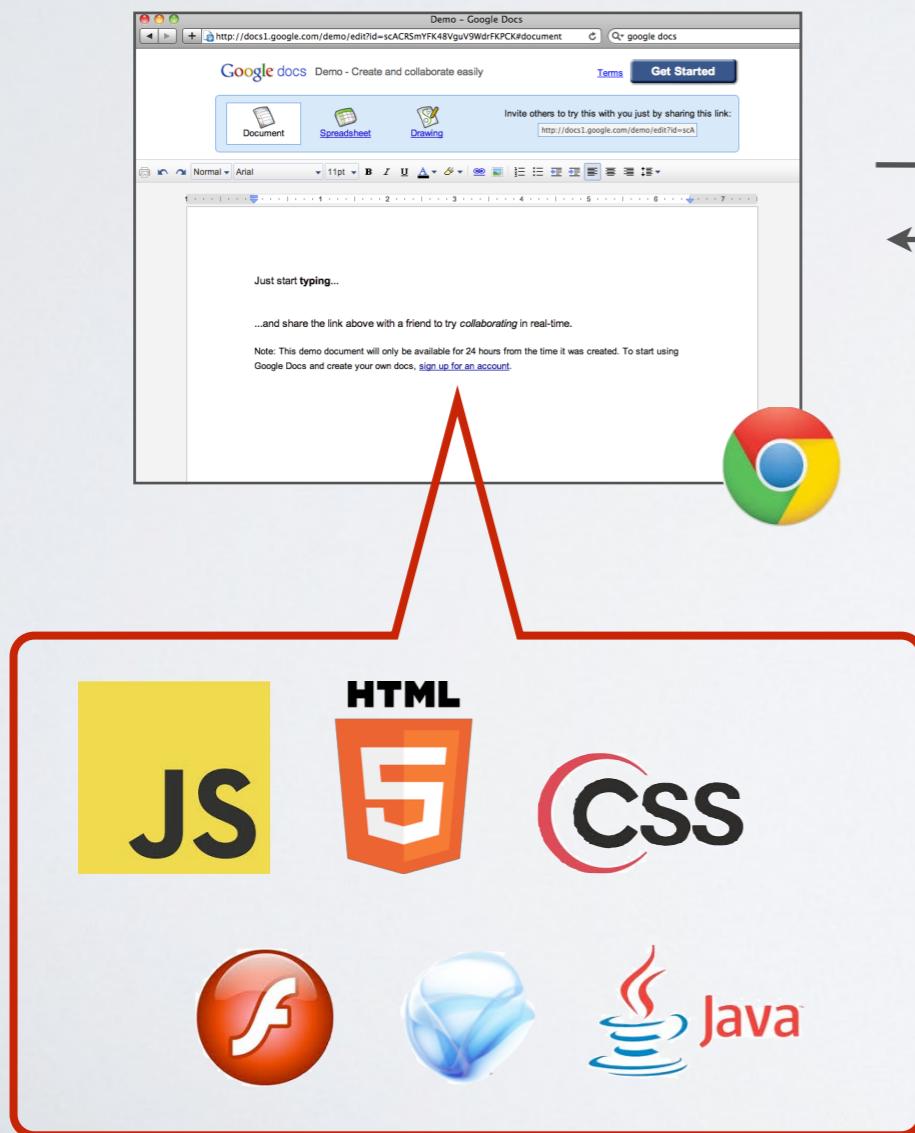
Our focus here!

What is a web application?

program running
on the browser

+

program running
on the server



Anatomy of a web application

The HTTP protocol

Network protocol for requesting/receiving data on the Web

- Standard TCP protocol on **port 80** (by default)
- **URI/URL** specifies what resource is being accessed
- Different **request methods**

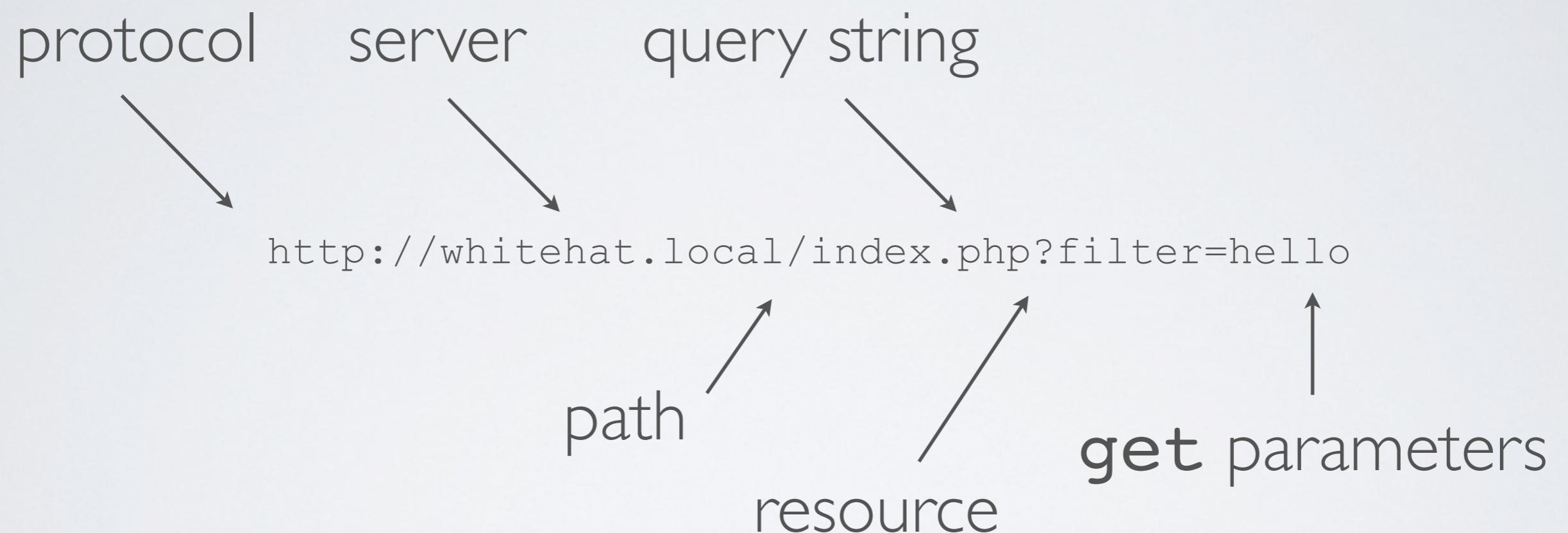
Let's look at what a web server does

telnet to a web server

```
> telnet whitehat.local 80  
GET /
```

enter HTTP requests

Anatomy of a URL



Authentication and Authorization

✓ Authentication

- Who are the authorized users?

✓ Authorization

- Who can access what and how?

The simple recipe for user authentication

1. **Ask the user for a login and password** and send it to the server (HTTP/POST request)
2. **Verify the login/password** based on information stored on the server (usually in the database)
3. **Start a session** once the user has been authenticated
4. **Grant access to resources** according to the session

The concept of session

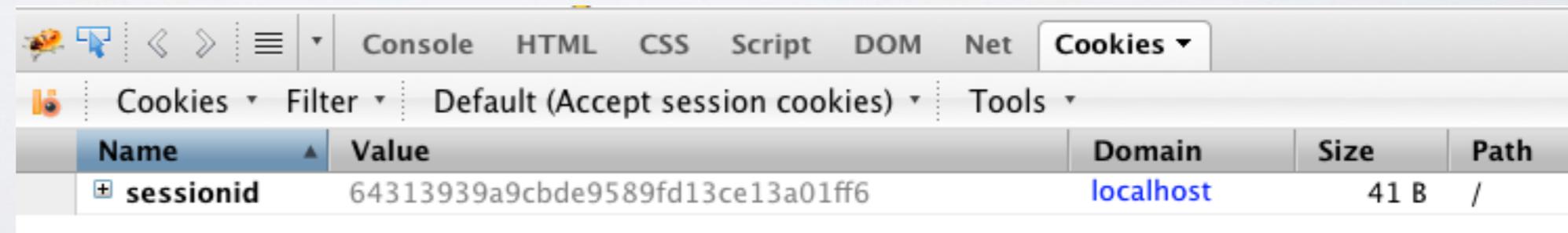
There is a **session id** (aka token)
between the browser and the web application

This session id should be **unique** and **unforgeable**
(usually a long random number or a hash)

- Stored in the cookie

The session id is bind to **key/value pairs data**

- Stored on the server

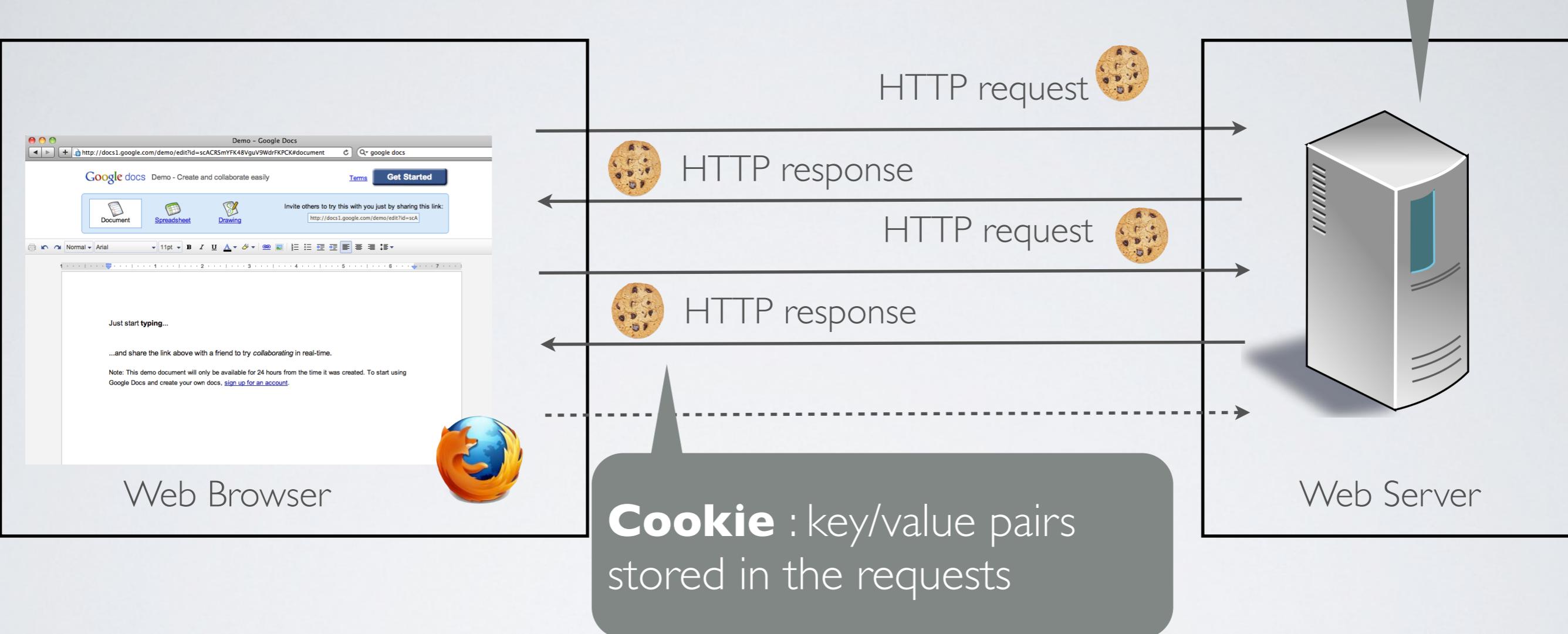


A screenshot of the Firebug developer toolbar, specifically the Cookies panel. The panel has a header with tabs: Console, HTML, CSS, Script, DOM, Net, and Cookies (which is currently selected). Below the header is a toolbar with icons for Cookies, Filter, Default (Accept session cookies), and Tools. The main area is a table with columns: Name, Value, Domain, Size, and Path. There is one row visible in the table:

Name	Value	Domain	Size	Path
sessionid	64313939a9cbde9589fd13ce13a01ff6	localhost	41 B	/

The big picture

Session : key/value pairs stored on the server



The user can **create, modify, delete** the session ID in the cookie

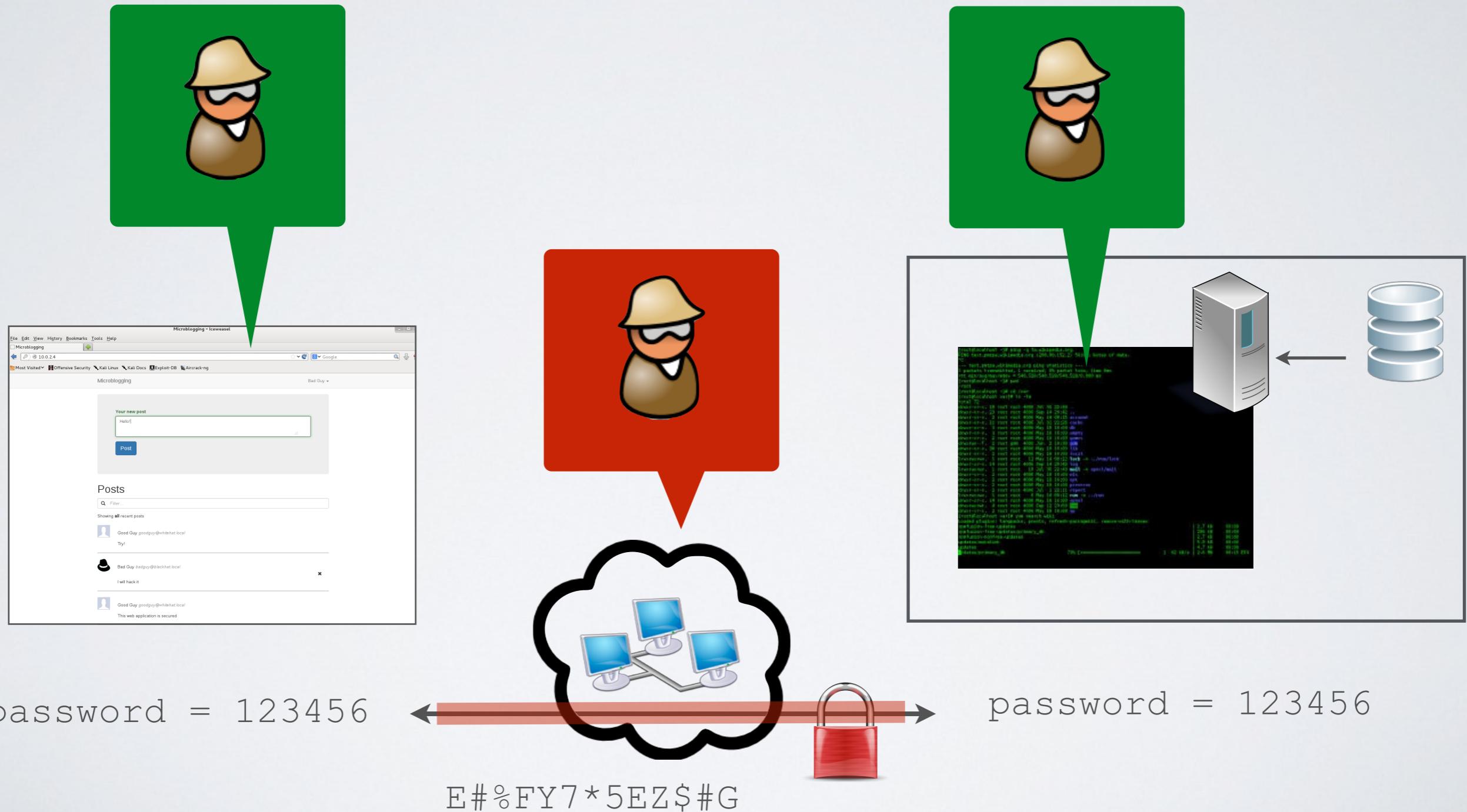
But **cannot access** the key/value pairs stored on the server

Hacking Authentication

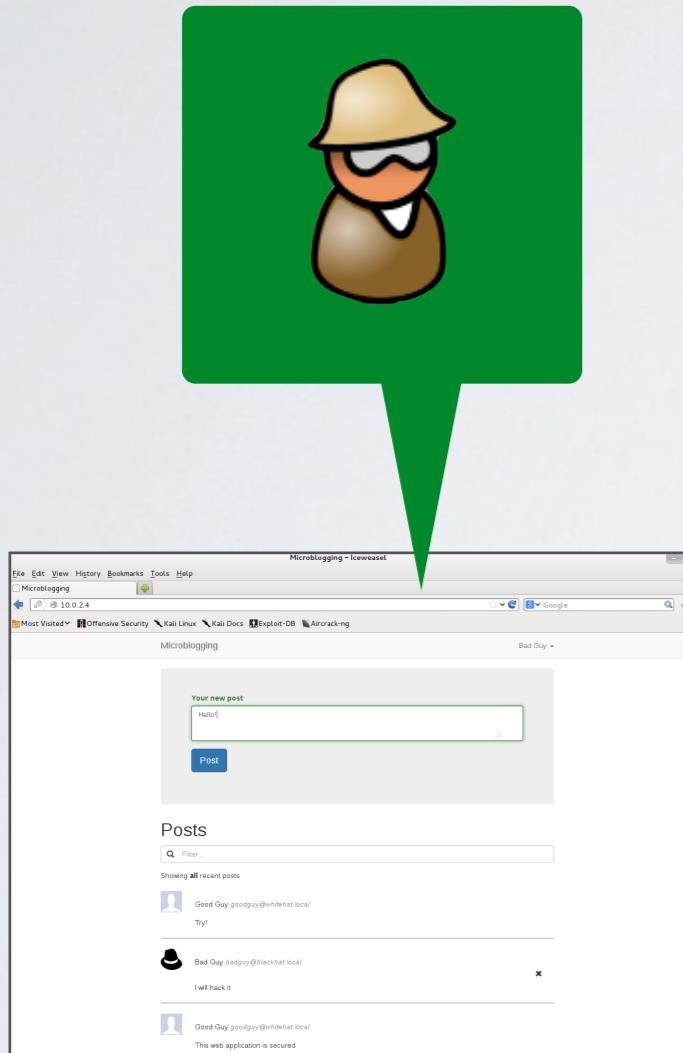
How to steal user's credentials

- ~~Brute force the password~~
- ~~Brute force the session ID~~
- Steal the user's password
- Steal the user's session ID

Limitation of HTTPS



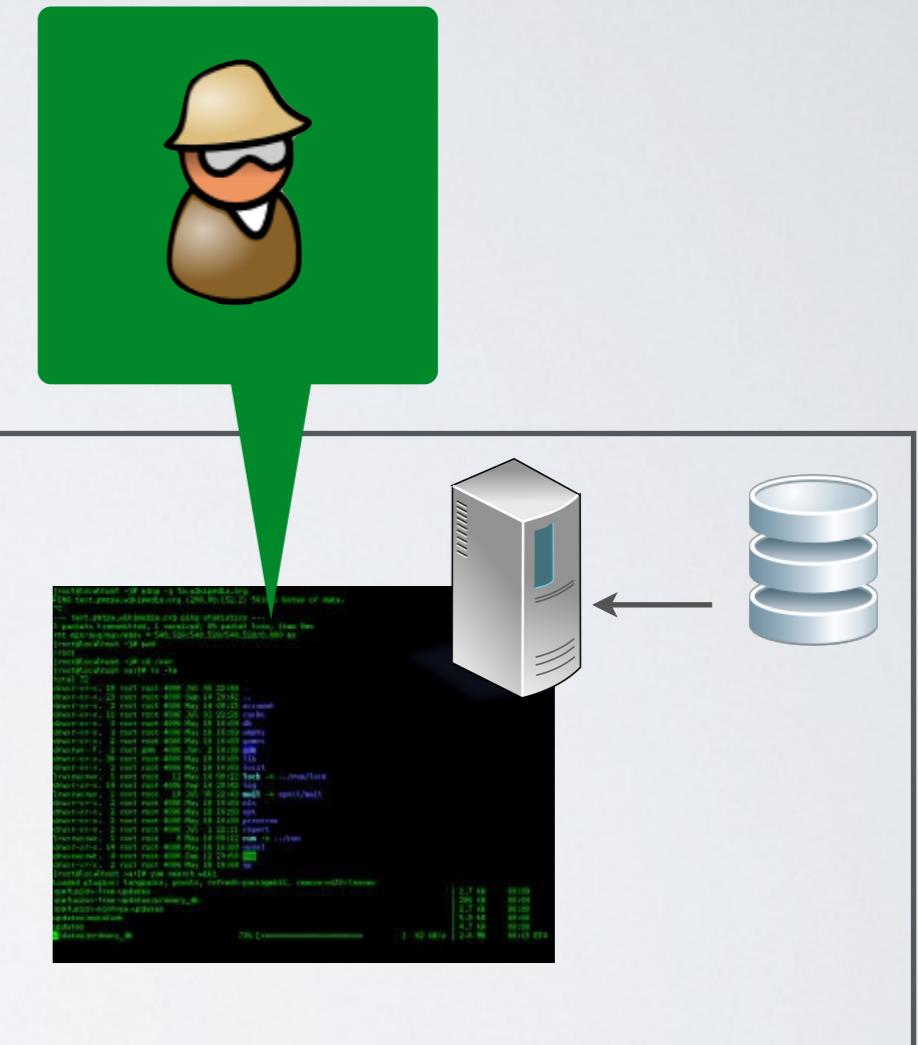
Stealing passwords from the client



- Social engineering - Phishing
- Keyloggers (keystroke logging)
- Data mining (emails, logs)
- Hack the client's code

Stealing passwords from the server

- Hack the server
- Hack the server's side code



Hacking the Client's Side Code

Client side's attacks

Content Spoofing

- inject arbitrary HTML content into a webpage

CSRF

- inject arbitrary urls into a webpage

xss

- inject arbitrary Javascript code into a webpage

Content Spoofing

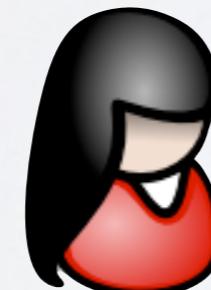
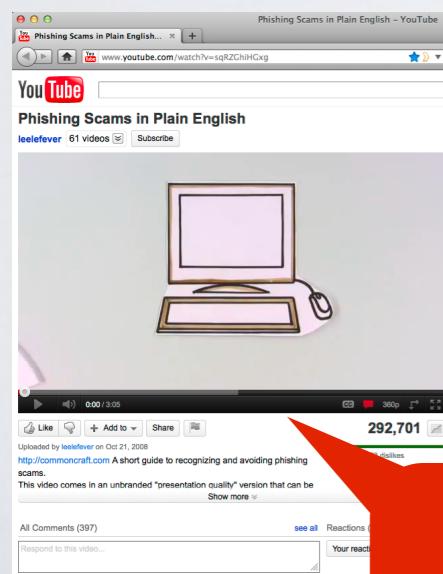
injecting arbitrary HTML content into a webpage



GET /?videoid=527

<html ...

comment = "Fun stuff ...



GET /?videoid=527

<html ...

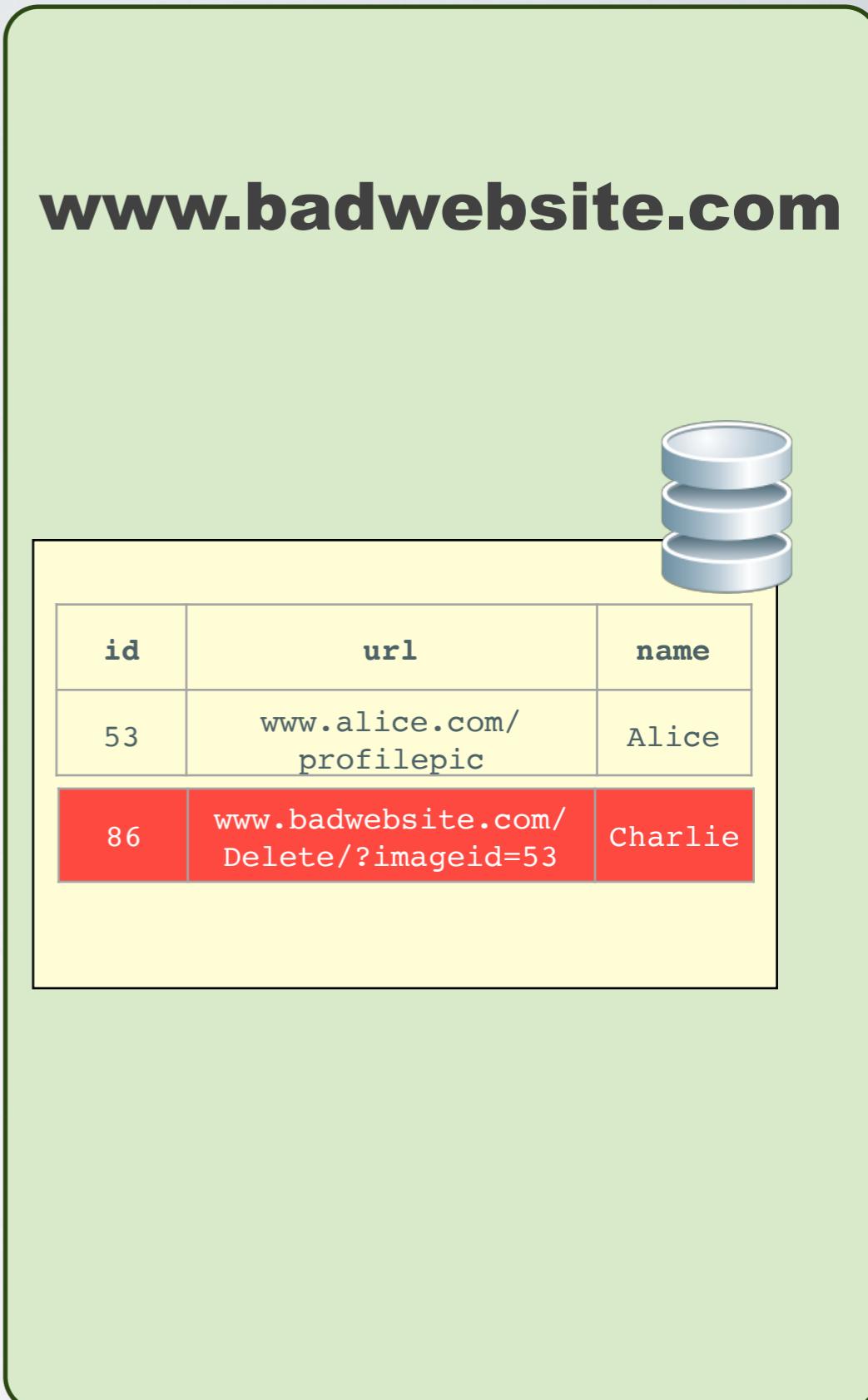
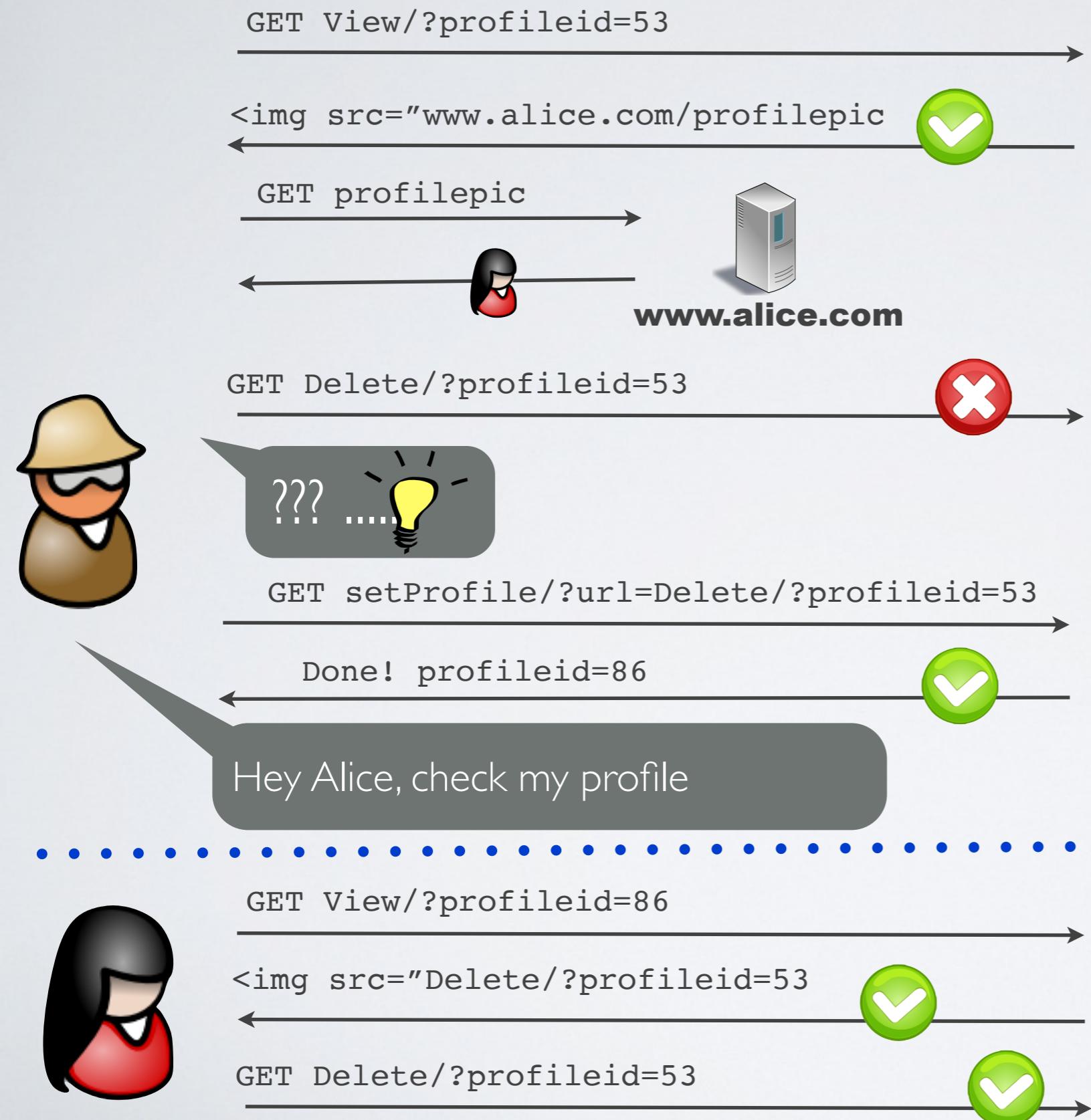


The page contains the attacker's code.

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

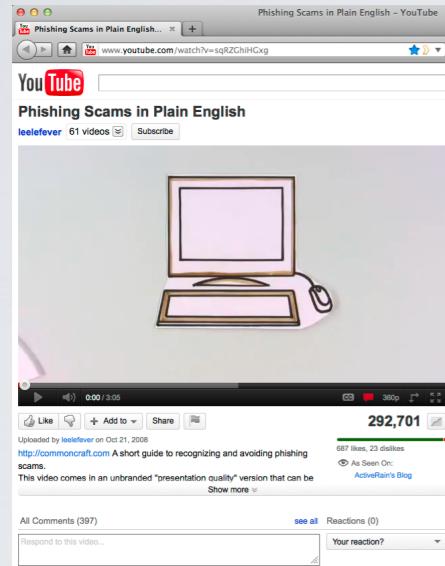
CSRF attack

injecting arbitrary urls into a webpage



XSS attack

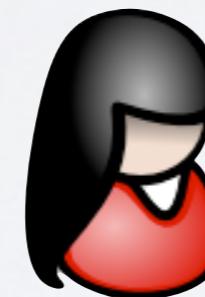
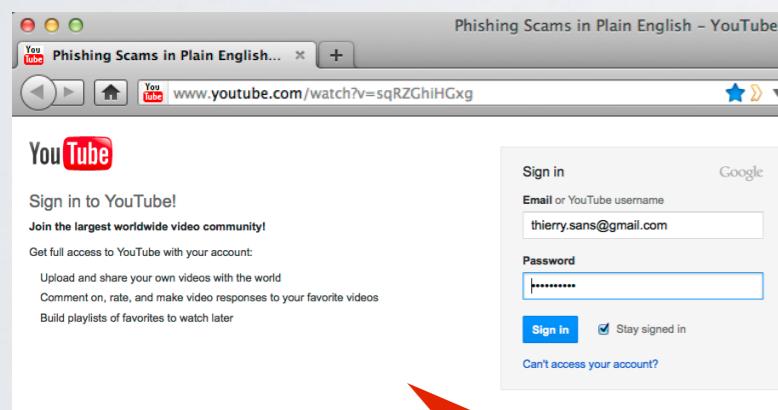
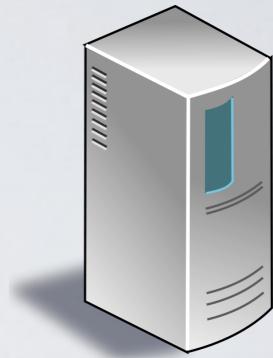
injecting arbitrary javascript into a webpage



GET /?videoid=527

<html ...

comment = "<script> ...



GET /?videoid=527

<html ...

login=Alice&password=123456

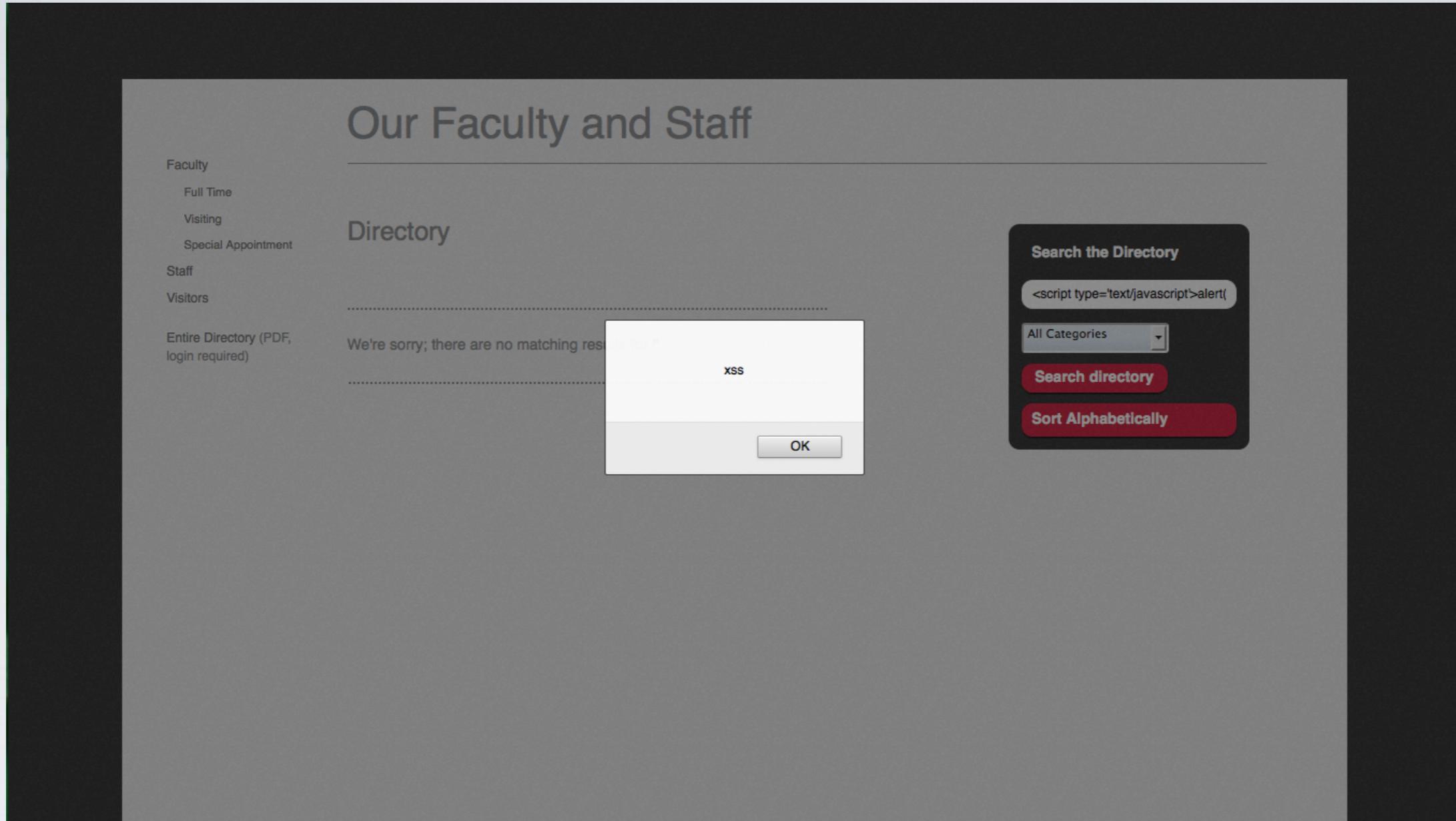


The script contained in the comments
modifies the page to look like the login page!

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

Scope of XSS attacks

- **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



XSS attacks are widespread

Hacking the Server's Side Code

Server's side attacks

SQL injection

- inject arbitrary SQL code executed on the server's database

File inclusion

- inject arbitrary code executed on the server

SQL Injection Attack

inject arbitrary SQL code executed on the server's database

loginPage.html



123456' OR '1'='1

name=Alice&pwd=123456



Access Deny!

Access Granted!

checkPassword.php

```
<?php  
  
$uid = SQLQuery("SELECT uid FROM LoginTable  
                  WHERE login=" . $_POST['name'] .  
                  "AND password =" . $POST['pwd']);  
  
if ($uid)  
    echo "Access Granted";  
else  
    echo "Access Denied";  
?>
```



Scope of SQL injection attacks

- retrieves, adds, modifies, deletes arbitrary information
- bypasses authentication
- installs a reverse shell

File Inclusion Attack

inject arbitrary code executed on the server



Web Penetration Testing

Web application security tools

- Proxy mapper
- Vulnerability scanner
- Replay HTTP requests
- (Exploit tool)

Commercial

AppScan



Acunetix

Burp Suite



Nikto



W3af

Vega



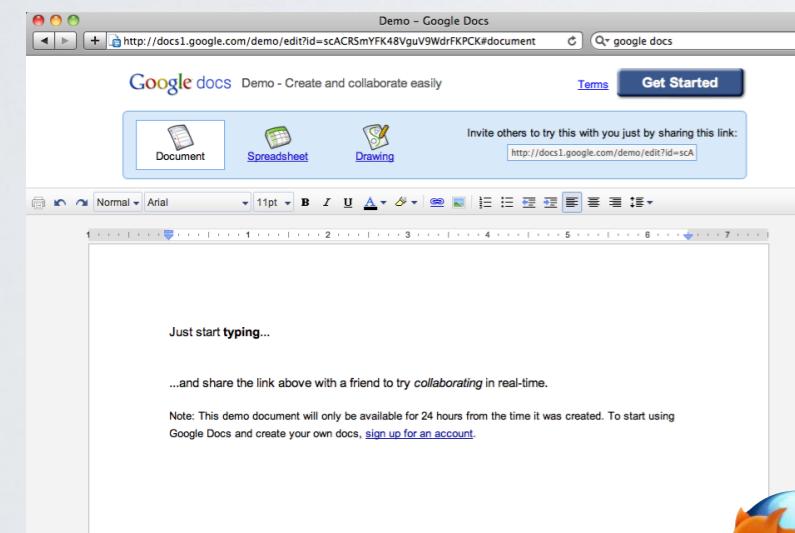
Open Source

... among others

Conclusion

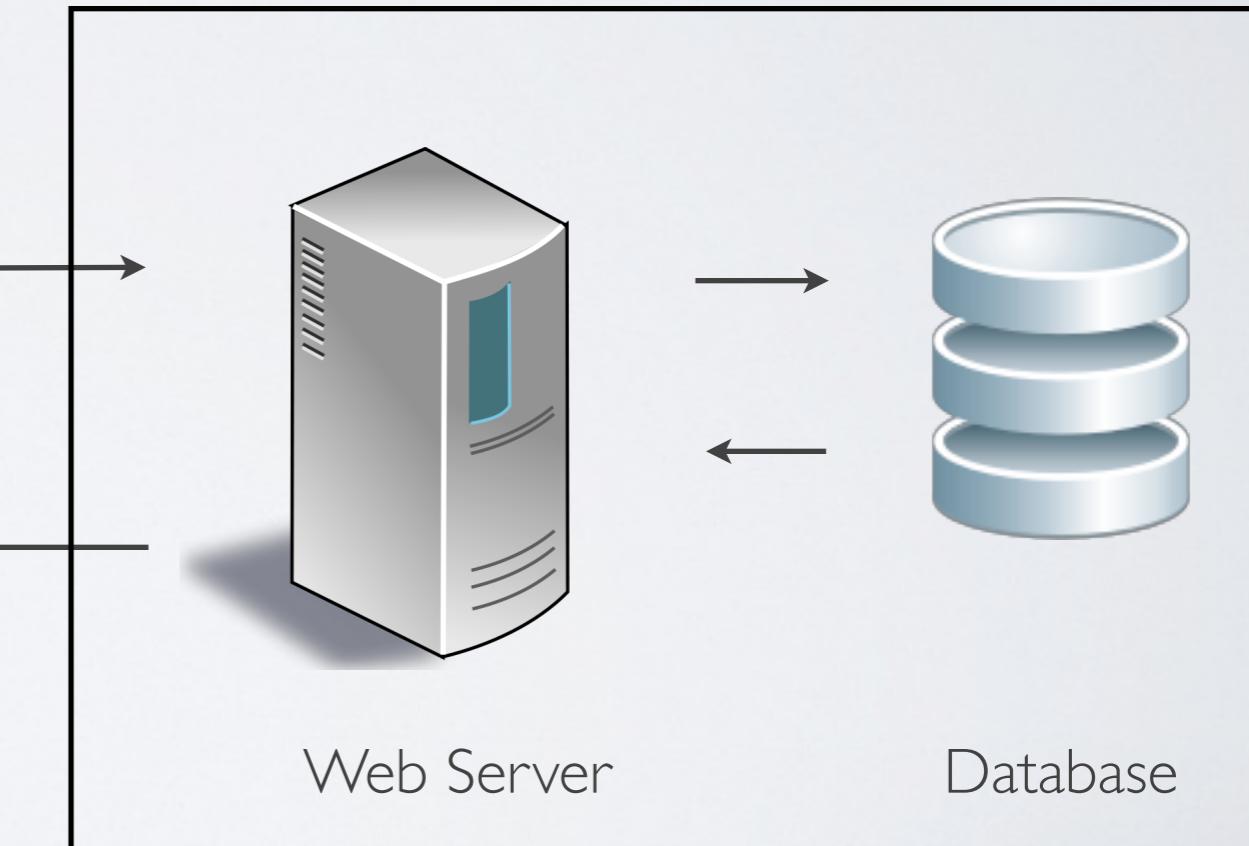
You have **absolutely no control** on the client

Client Side



Web Browser

Server Side



References

- Mozilla Secure Coding Guideline
https://wiki.mozilla.org/WebAppSec/Secure_Coding_Guidelines
- Ruby on Rails Security Page
<http://guides.rubyonrails.org/security.html>
- Django Security Page
<https://docs.djangoproject.com/en/dev/topics/security/>
- PHP Security Pages
<http://php.net/manual/en/security.php>
<http://phpsec.org/projects/guide/>