Web Security

CS5435:
Security and Privacy (in the wild?)

Paul Grubbs

http://www.cs.cornell.edu/~paulgrubbs/

pag225 at cornell dot edu

Liberal borrowing from Ristenpart, Wisc CS642 and Mitchell, Boneh, Stanford CS 155

Who am I?

- Fifth-year (!) PhD student in CS
- Undergrad at Indiana University
- Security, applied crypto, systems research
- Spent 2.5 years in industry before PhD
  - (crypto software engineer)

Web security part 1

Basic web security models

Browser security

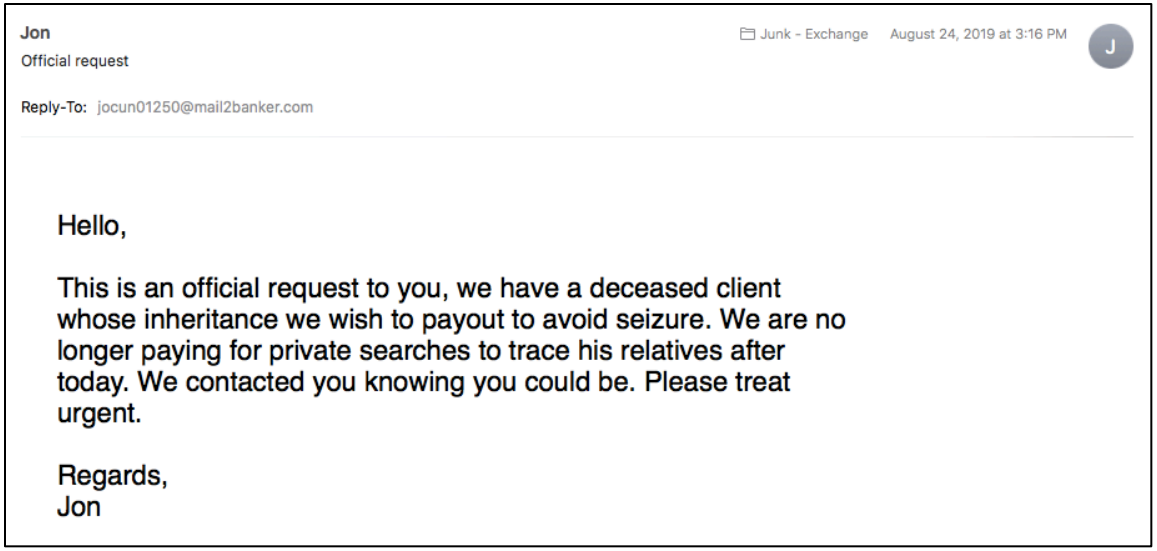
Same-origin policy / Navigation policy

Cookies / Session handling

New threat model

Recall email spam and fraud (discussed last week)

Spam and fraud attacks *abuse correctly-functioning* email…



Jon
Official request

Junk - Exchange August 24, 2019 at 3:16 PM

J

Reply-To: jocun01250@mail2banker.com

Hello,

This is an official request to you, we have a deceased client whose inheritance we wish to payout to avoid seizure. We are no longer paying for private searches to trace his relatives after today. We contacted you knowing you could be. Please treat urgent.

Regards,
Jon

New threat model

Recall email spam and fraud (discussed last week)

Spam and fraud attacks *abuse correctly-functioning* email…

Web

Today we'll talk about exploiting *mistakes* in applications.
Different kinds of problems!

Equifax breach
(flaw in Struts)

On Aug. 22, the **Apache Software Foundation** released software updates to fix a critical vulnerability in Apache Struts, a Web application platform used by an estimated 65 percent of Fortune 100 companies. Unfortunately, computer code that can be used to exploit the bug has since been posted online, meaning bad guys now have precise instructions on how to break into vulnerable, unpatched servers.

Attackers can exploit a Web site running the vulnerable Apache Struts installation using nothing more than a Web browser. The bad guy simply needs to send the right request to the site and the Web server will run any command of the attacker's choosing. At that point, the intruder could take any number of actions, such as adding or deleting files, or copying internal databases.

WWW

Tim Berners-Lee and Robert Cailliau 1990
HTTP, CERN httpd, gopher

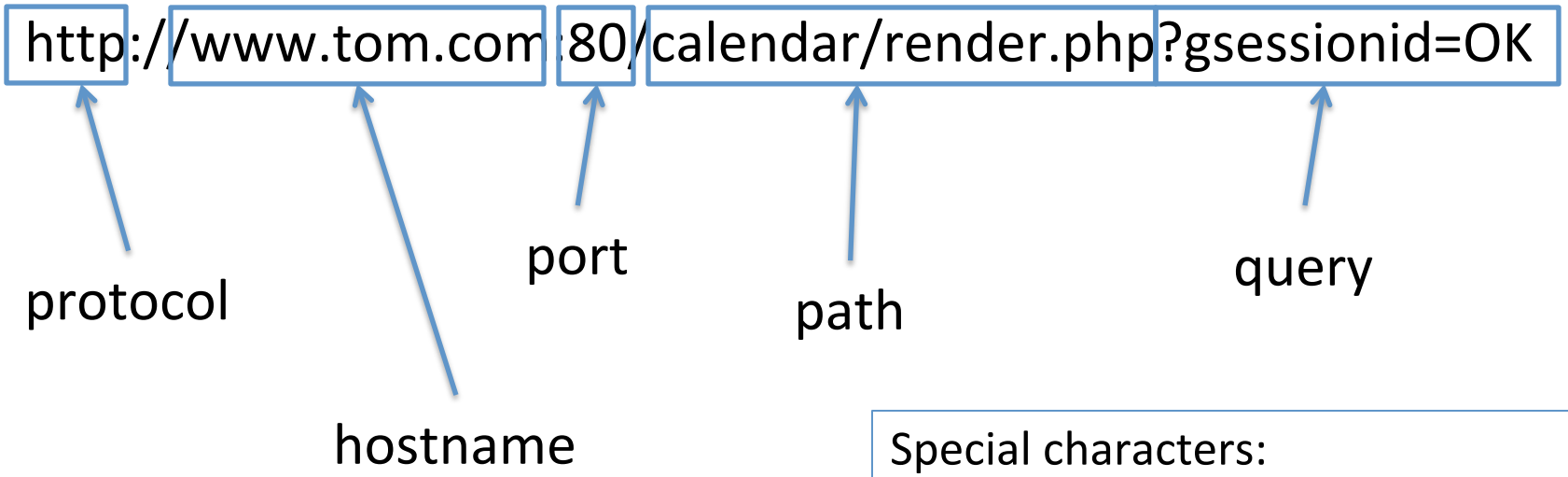
1993 Mosiac web browser (UIUC, Marc Andreesen)

1994 W3C WWW Consortium --- generate standards
Gopher started charging licensing fees
(Univ of Minnesota)

Nowadays: ecosystem of technologies

- HTTP / HTTPS (hypertext transport protocol)
- AJAX (asynchronous javascript and XML)
- PHP (hypertext preprocessor)
- Javascript
- SQL (structured query language)
- Apache
- Ruby
- http://w3schools.com/

Uniform resource locators (URLs)



URL's only allow ASCII-US characters.
Encode other characters:

%0A = newline
%20 = space

Special characters:
+ = space
? = separates URL from parameters
% = special characters
/ = divides directories, subdirectories
= bookmark
& = separator between parameters

HTTP Request

Method File HTTP version Headers

```
GET /index.html HTTP/1.1
Accept: image/gif, image/x-bitmap, image/jpeg, */*
Accept-Language: en
Connection: Keep-Alive
User-Agent: Mozilla/1.22 (compatible; MSIE 2.0; Windows 95)
Host: www.example.com
Referer: http://www.google.com?q=dingbats

```

Blank line

Data – none for GET

GET : no side effect POST : possible side effect

HTTP Response

HTTP version

Status code

Reason phrase

Headers

```
HTTP/1.0 200 OK
Date: Sun, 21 Apr 1996 02:20:42 GMT
Server: Microsoft-Internet-Information-Server/5.0
Connection: keep-alive
Content-Type: text/html
Last-Modified: Thu, 18 Apr 1996 17:39:05 GMT
Set-Cookie: …
Content-Length: 2543

<HTML> Some data... blah, blah, blah </HTML>
```

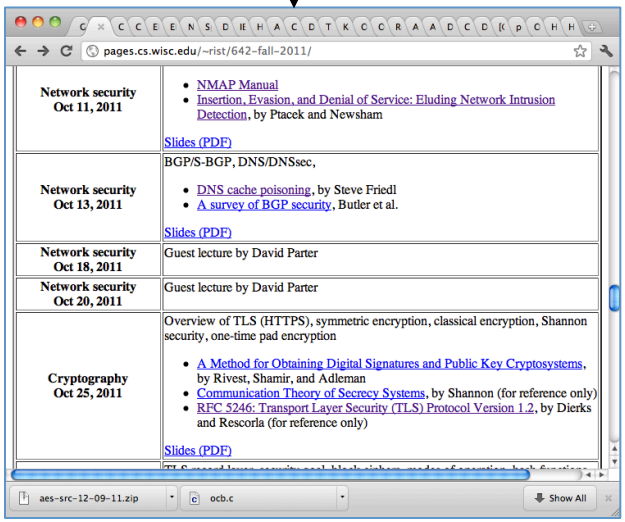
Data

Cookies

Browser execution

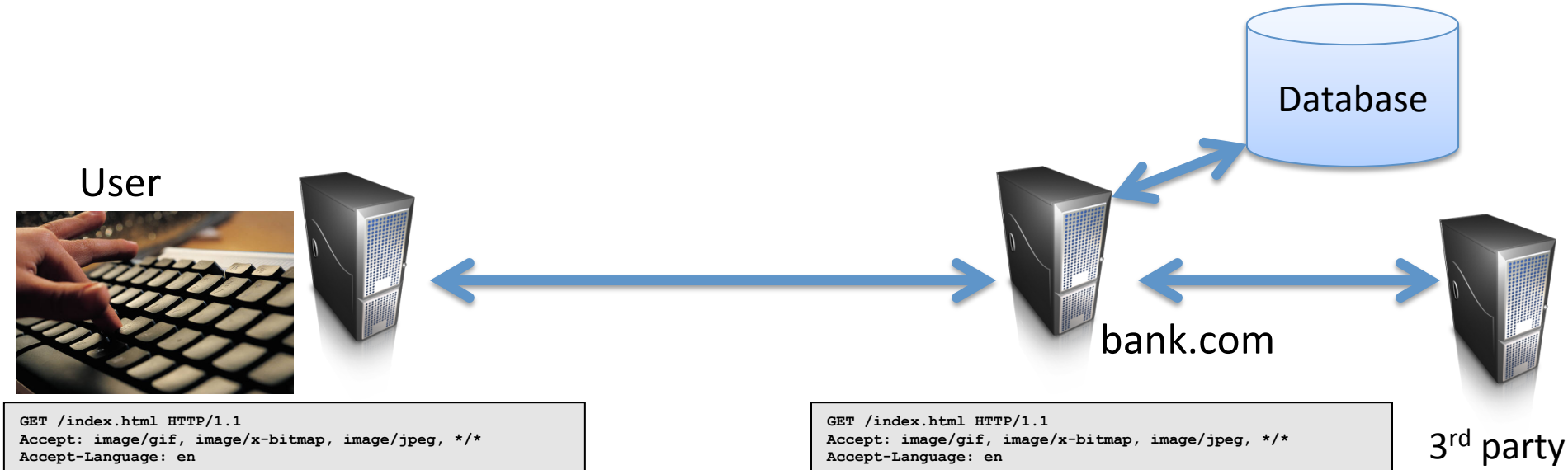
```
HTTP/1.0 200 OK
Date: Sun, 21 Apr 1996 02:20:42 GMT
Server: Microsoft-Internet-Information-
Server/5.0
Connection: keep-alive
Content-Type: text/html
Last-Modified: Thu, 18 Apr 1996 17:39:05 GMT
Set-Cookie: …
Content-Length: 2543

<HTML> Some data... blah, blah, blah </HTML>
```



- Each window (or tab):
  - Retrieve/load content
  - Render it
    - Process the HTML
    - Might run scripts, fetch more content, etc.
  - Respond to events
    - User actions: OnClick, OnMouseover
    - Rendering: OnLoad, OnBeforeUnload
    - Timing: setTimeout(), clearTimeout()

Putting it all together



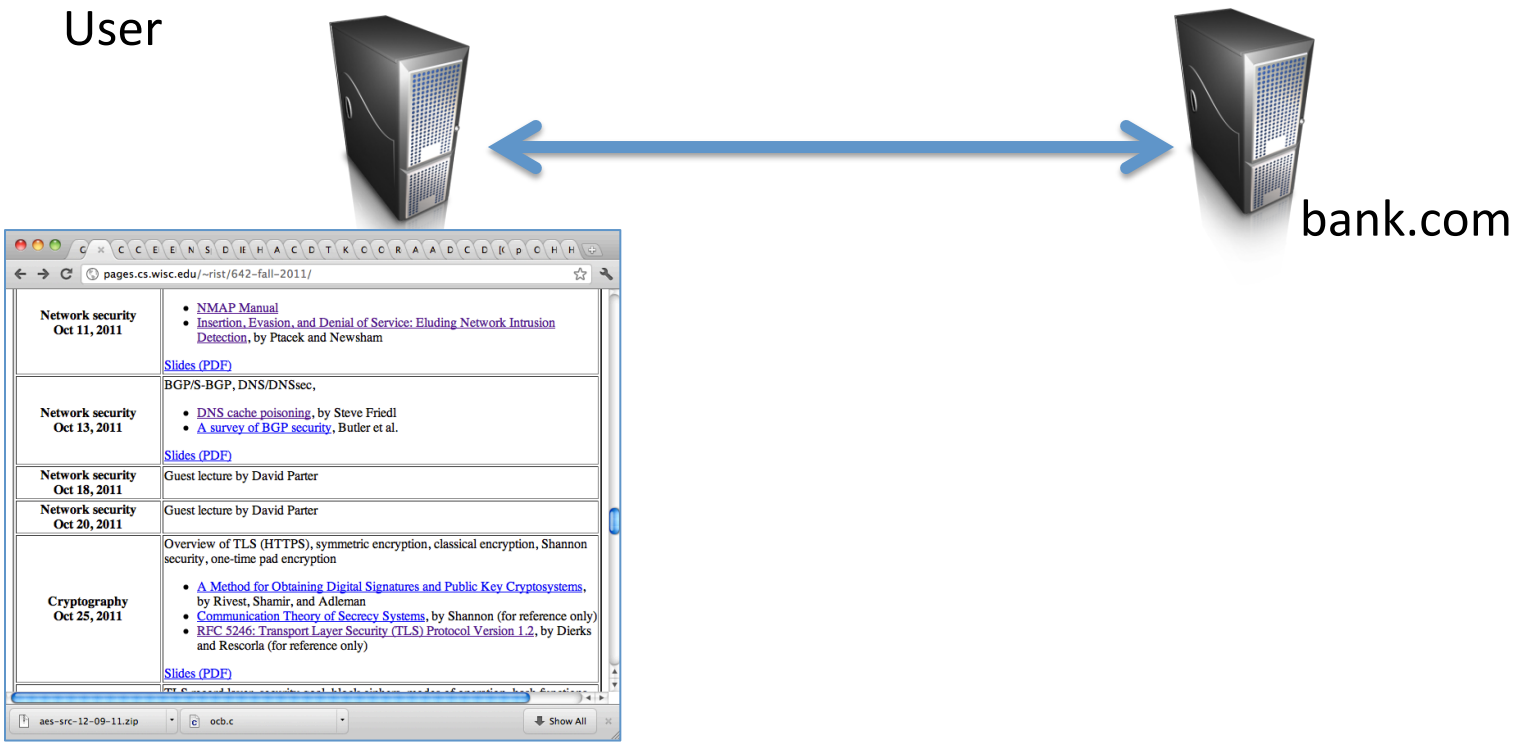

```
GET /index.html HTTP/1.1
Accept: image/gif, image/x-bitmap, image/jpeg, */*
Accept-Language: en
Connection: Keep-Alive
User-Agent: Mozilla/1.22 (compatible; MSIE 2.0; Windows 95)
Host: www.example.com
Referer: http://www.google.com?q=dingbats
```

```
GET /index.html HTTP/1.1
Accept: image/gif, image/x-bitmap, image/jpeg, */*
Accept-Language: en
Connection: Keep-Alive
User-Agent: Mozilla/1.22 (compatible; MSIE 2.0; Windows 95)
Host: www.example.com
Referer: http://www.google.com?q=dingbats
```

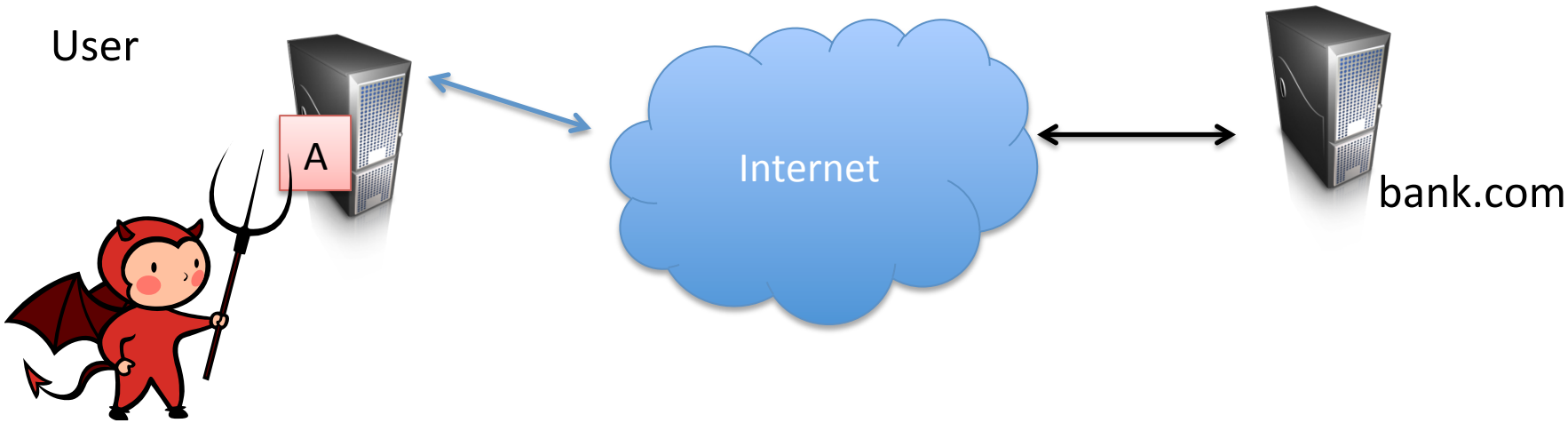
```
HTTP/1.0 200 OK
Date: Sun, 21 Apr 1996 02:20:42 GMT
Server: Microsoft-Internet-Information-
Server/5.0
Connection: keep-alive
Content-Type: text/html
Last-Modified: Thu, 18 Apr 1996 17:39:05 GMT
Set-Cookie: …
Content-Length: 2543

<HTML> Some data... blah, blah, blah </HTML>
```

Putting it all together

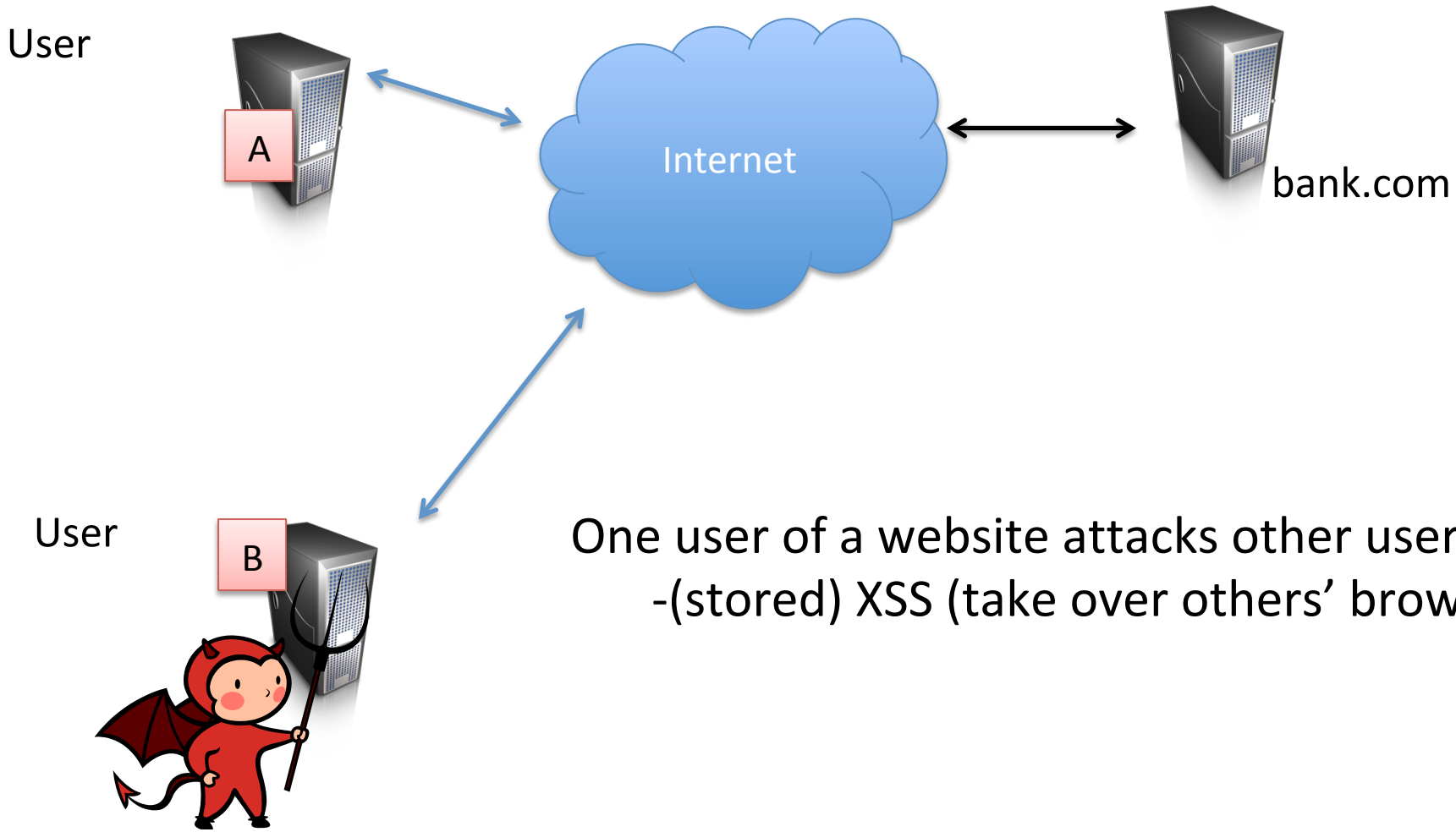


Threat model



Malicious user attacks the website
-SQL injection (steal data)

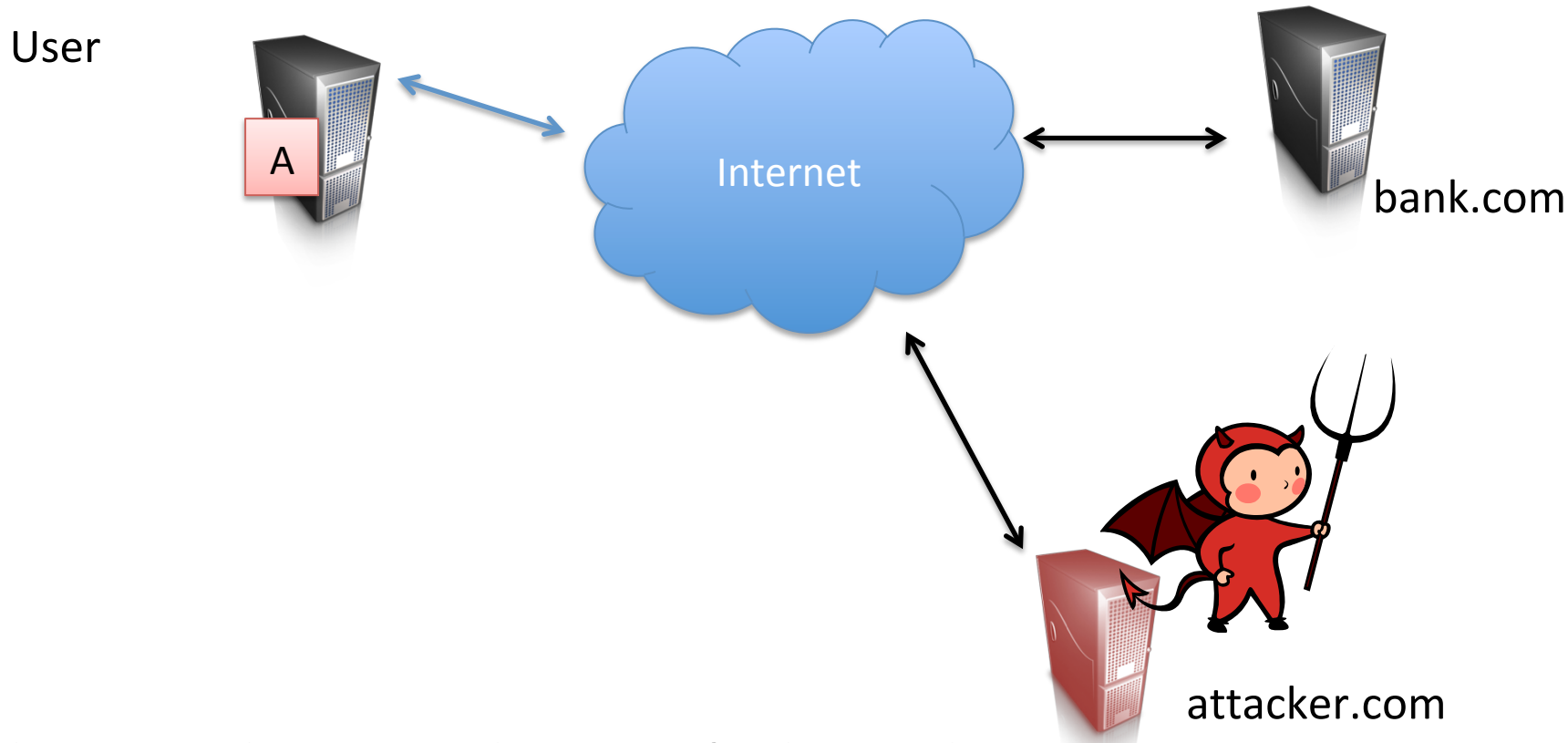
Threat model



One user of a website attacks other users

-(stored) XSS (take over others' browser)

Threat model



Malicious website attacks user of other website

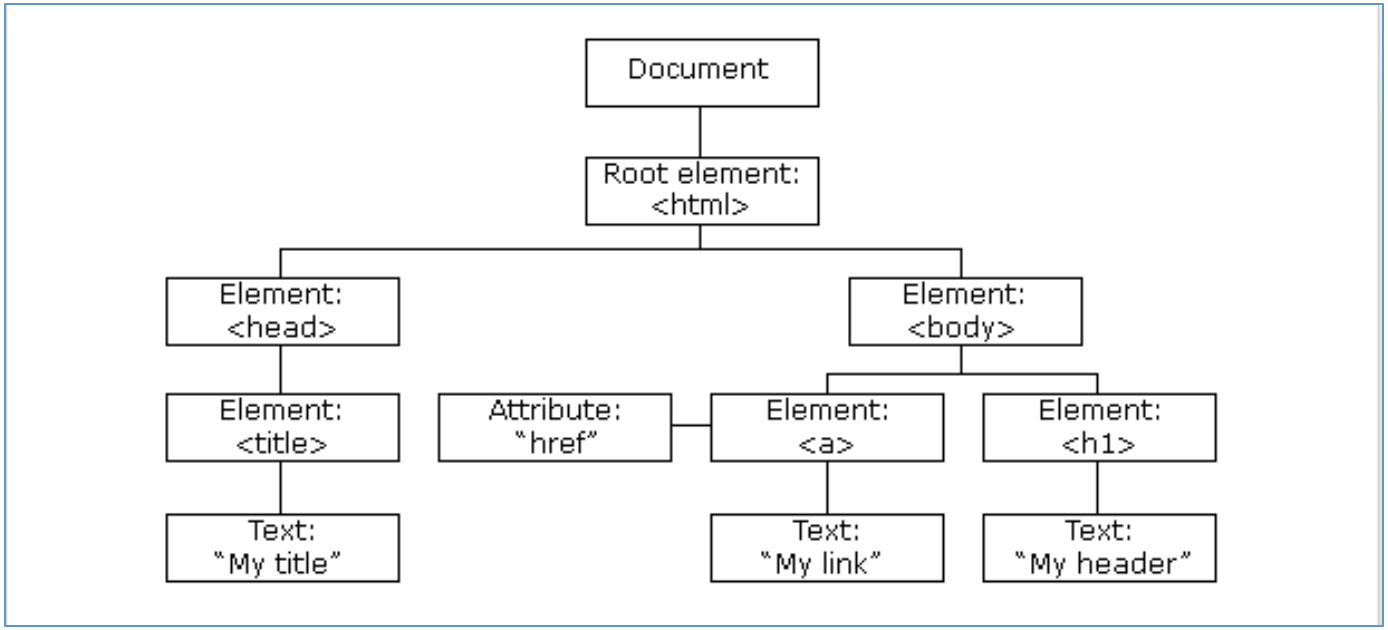
-CSRF (steal money from bank account)

Document Object Model (DOM)

Object-oriented way to refer to objects in a web page

Properties: document.alinkColor, document.URL, document.forms[ ], document.links[ ], document.anchors[ ]

Methods: document.write(document.referrer)



From http://w3schools.com/htmldom/default.asp

Document Object Model (DOM)

Object-oriented way to refer to objects in a web page

Properties: document.alinkColor, document.URL, document.forms[ ], document.links[ ], document.anchors[ ]

Methods: document.write(document.referrer)

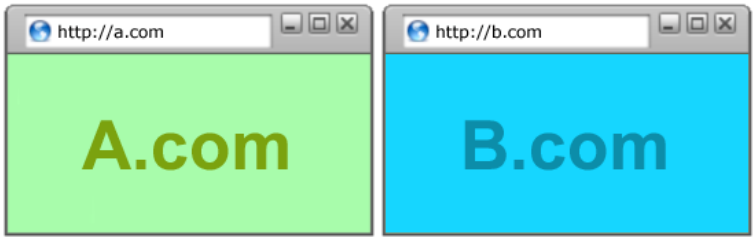
Browser Object Model (BOM)

window, document, frames[], history, location, navigator (type and version of browser)

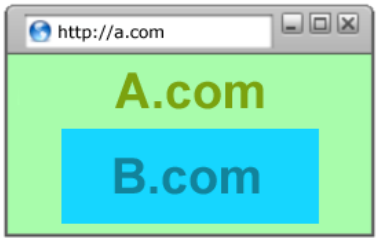
Browser security model

Should be safe to visit an attacker website

Should be safe to visit sites simultaneously



Should be safe to delegate content



Browser isolation

Browser is running untrusted inputs (attacker webpage)

Like all big, complex software, browser has security vulnerabilities

Browsers include “Rich Internet Applications” (RIAs) that increase attack surface:

e.g., Adobe Flash

Malicious website exploits browser, from there system

In-class exercise

<u>Five-minute exercise:</u> with your neighbor, use your laptop to visit a website you use every day and view its source using “inspect” or developer tools feature of your browser. Find some iframes, scripts, or DOM commands.

(1) is there content from just a single site, or multiple sites?
(2) Are scripts or images being loaded from some other source?
(3) what kinds of cookies does the site use?

Web pages are not single-origin

IFrames: **<iframe src**="//site.com/frame.html" **>** </iframe>

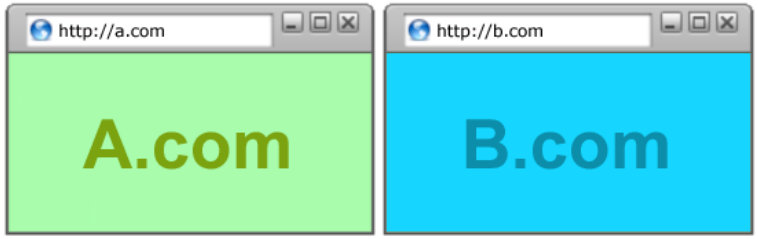
Scripts: **<script src**="//site.com/script.js" **>** </script>

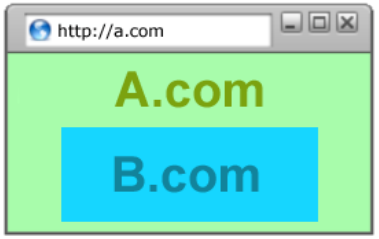
CSS:

<**link** rel="stylesheet" type="text /css" href="//site.com/theme.css" />

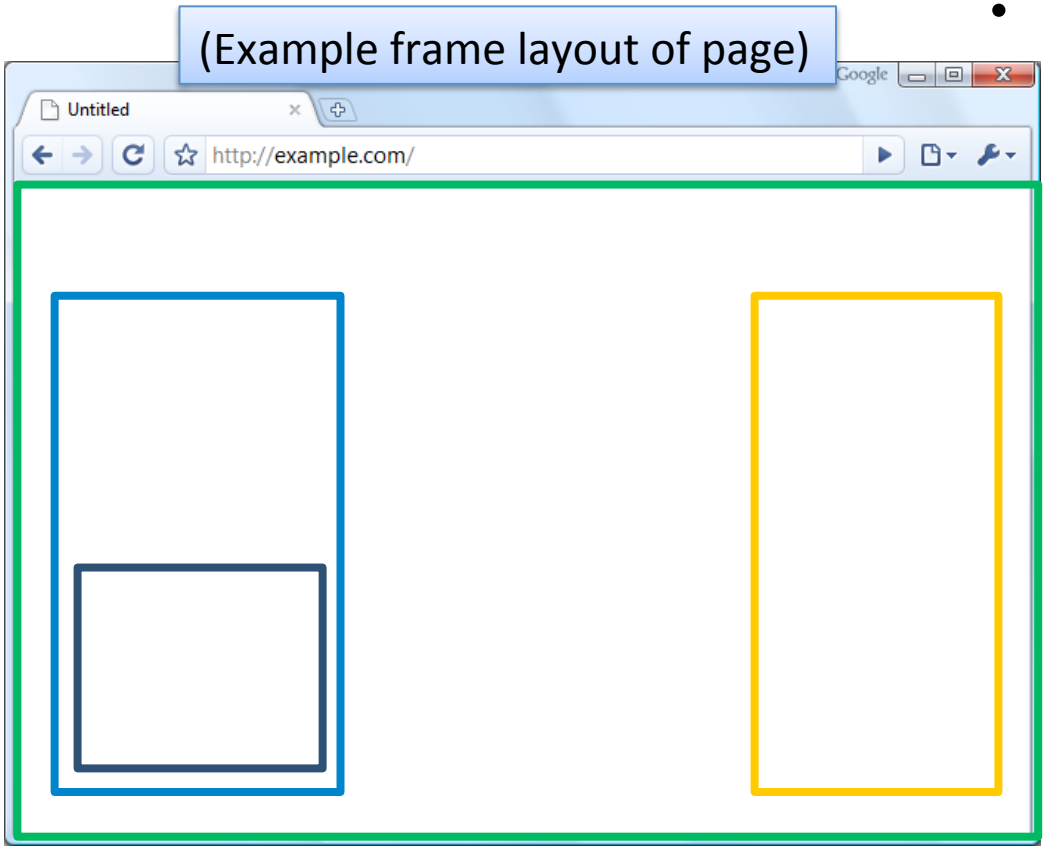
Objects (flash): [using swfobject.js script]

```
<script>
        var so = new SWFObject('//site.com/flash.swf', …);
        so.addParam('allowscriptaccess', 'always');
        so.write('flashdiv');
</script>
```





Browser handles multiple sites, must maintain separate security contexts for each



Browsers

- Primitives
  - Document object model
  - Frames
  - Cookies / local storage

Principals: Origins

- Mandatory access controls

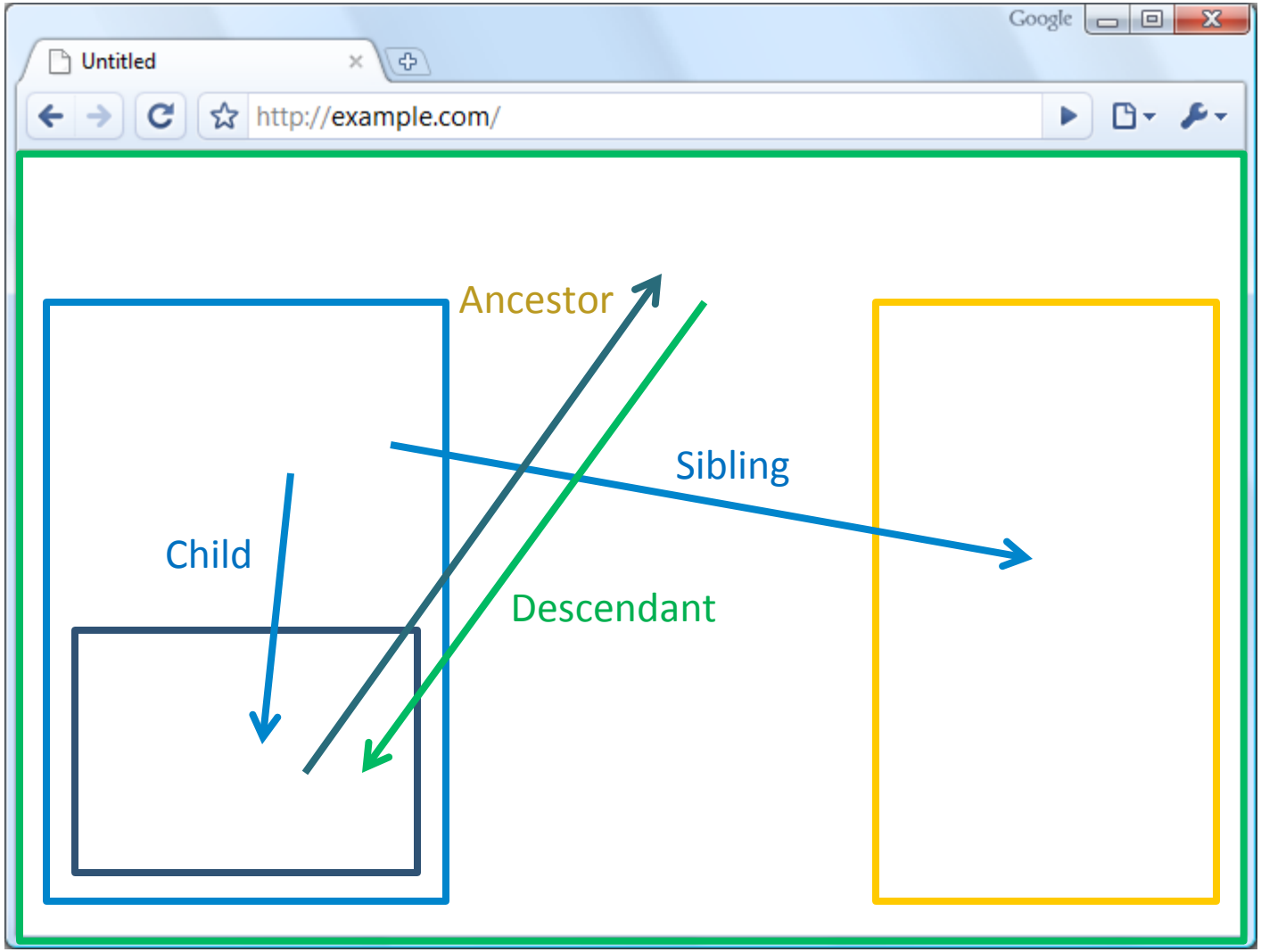
Vulnerabilities

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

Same-origin policy

- Each frame of page(s) has an origin
  - protocol://host:port
  - Origin is (protocol,host,port)
- Frame can access its own origin
  - Network access, Read/write DOM, storage (cookies)
- Frame cannot access data associated with another origin

Frame relationships



Frame policies

canScript(A,B) and canNavigate(A, B)

- Permissive
  - any frame can navigate any other frame
- Child
  - only can navigate if you are parent
- Descendent
  - only can navigate if you are ancestor

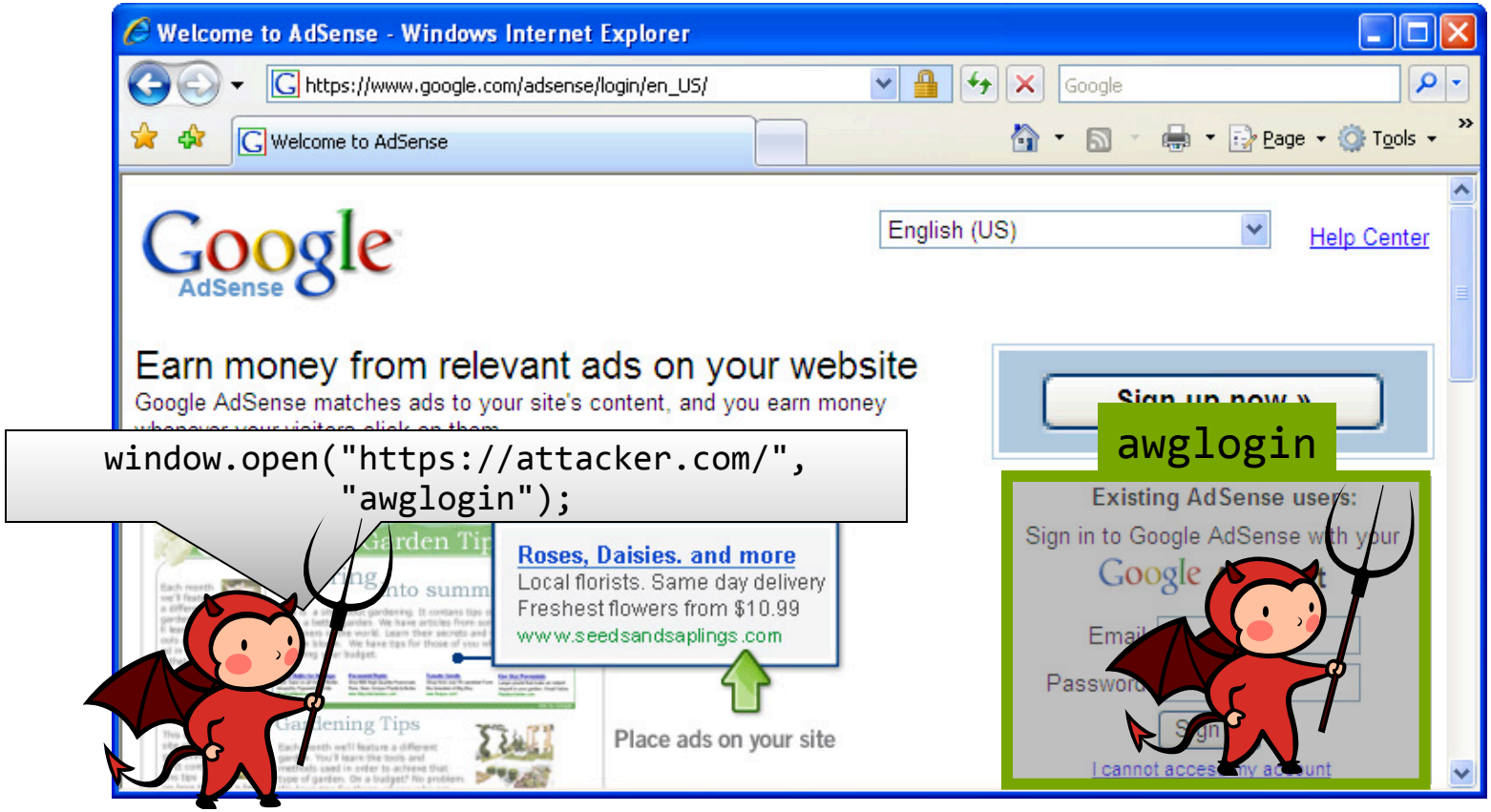
Which do you think should be used?

Legacy Browser Behavior

Browser	Policy
IE 6 (default)	Permissive
IE 6 (option)	Child
IE7 (no Flash)	Descendant
IE7 (with Flash)	Permissive
Firefox 2	Window
Safari 3	Permissive
Opera 9	Window
HTML 5	Child

Problems with permissive

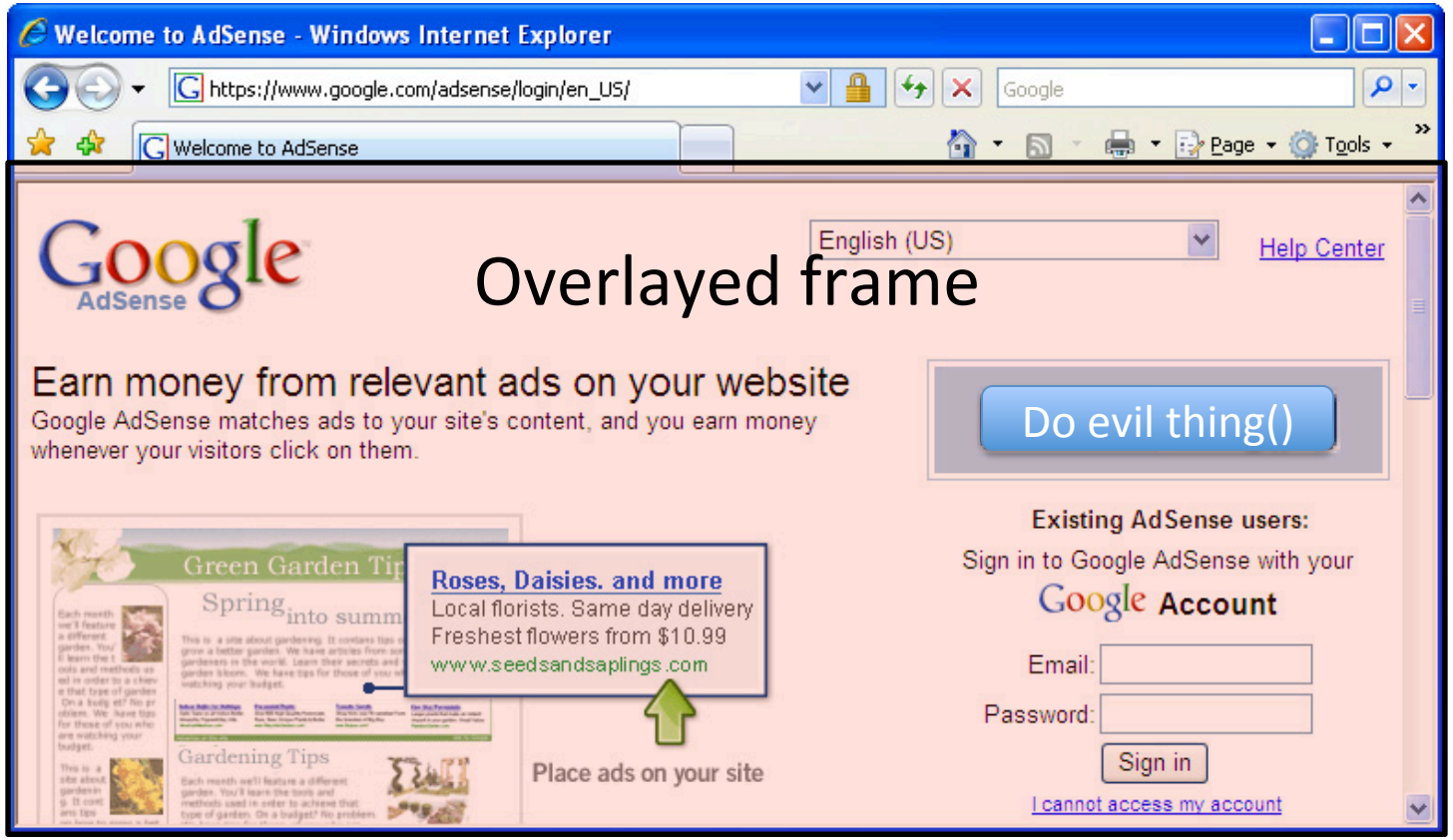
frames['right'].window.location=“evil.com/login.html”;



Adoption of Descendant Policy

Browser	Policy
IE7 (no Flash)	Descendant
IE7 (with Flash)	Descendant
Firefox 3	Descendant
Safari 3	Descendant
Opera 9	(many policies)
HTML 5	Descendant

UI Redressing (Clickjacking)



Defense: NoScript plugin attempts to prevent this for Firefox

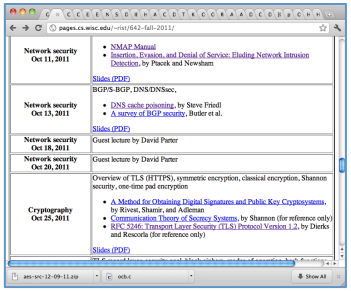
Popping back up a level

- We've just seen how browsers prevent malicious code from stealing data.
- Q: What kinds of data are useful for attackers to steal?
  - A: Cookies

What is a cookie, really?

- Stateful clients are “weird” for HTTP. Cookies remedy this.
- Types of cookies:
  - Session cookie (authentication)
  - Persistent cookie
  - tracking cookie
- three parts:
  - name
  - value
  - attribute/value pairs (e.g. expiry, security rules)

Cookies: Setting/Deleting



GET …

HTTP Header:

Set-cookie: NAME=VALUE ;

domain = (when to send) ;
path = (when to send)

scope

secure = (only send over SSL);
expires = (when expires) ;
HttpOnly

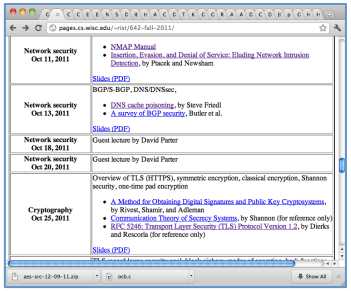
if expires=NULL:
this session only

- Delete cookie by setting “expires” to date in past
- Default scope is domain and path of setting URL
- Client can also set cookies (Javascript)

Cookie scope rules (domain and path)

- Say we are at www.wisc.edu
  - Any non-TLD suffix can be scope:
    - allowed: www.wisc.edu or wisc.edu
    - disallowed: www2.wisc.edu or ucsd.edu
- Path can be set to anything

Cookies: reading by server



GET /url-domain/url-path

Cookie: name=value

- Browser sends all cookies such that
  - domain scope is suffix of url-domain
  - path is prefix of url-path
  - protocol is HTTPS if cookie marked “secure”

Cookie security issues?

- Cookies have no integrity
  - HTTPS cookies can be overwritten by HTTP cookie (network injection)
  - Malicious clients can modify cookies
    - Shopping cart vulnerabilities
- Scoping rules can be abused
  - blog.example.com can read/set cookies for example.com
- Privacy
  - Cookies can be used to track you around the Internet
- HTTP cookies sent in clear
  - Session hijacking

STORIES OF NOTE

Batman: Arkham City This nth
D: MW3 Ships
ne Gold - Skyrim
inbow 6 Patriots
A5 Announcement
ttlefield 3 Ships
AV Trailer Next Week
3 MP Confirmed
rtal 2 DLC Oct 4

Tuesday, Nov 08, 2011

PC Batman: Arkham City This Month [10:40 am ET] - Share - 2 Comments

WBIE announces the release dates for the delayed Windows PC edition of **Batman: Arkham City**, the stealth/action sequel:

> Warner Bros. Interactive Entertainment and DC Entertainment today confirmed that the Games for Windows PC version of Batman: Arkham City will be available in North America beginning November 22, in Australia beginning November 23, in France and Benelux beginning November 24, and in other European territories beginning November 25.

Answers.com Now Only With Facebook and Own Login

Posted by **timothy** on Tuesday November 08, @12:30PM
from the you-haff-been-assimilated dept.

facebook

CptnHarlock writes

> "Today the registered users of Answers.com received an email informing them that the site has ended support for Yahoo, Twitter, Google, or LinkedIn as a way to sign into their site. Facebook is the sole external way left to log in. A local login and password were generated and sent by email and the old (non-Facebook) logins deactivated. Score another one for Facebook.com in the login consolidation wars."

Read the **14** comments

facebook google privacy

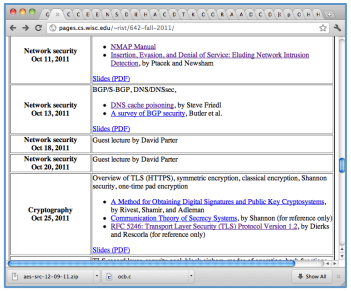
```
<script type="text/javascript">
	//<![CDATA[
		var hint   = 'mainpage';
	document.write('<script type="text/javascript" src="http://ad.doubleclick.net/adj/
ostg.slashdot/
pg_index_p1_leader;pg=index2;logged_in=0;tile='+dfp_tile+';sz=728x90;u=;ord='+dfp_or
d+'?"><\/script>');
		dfp_tile++;
	//]]>
</script>
```

In addition to ads based on interest categories, Google allows advertisers (including Google) to show you ads based on your previous interactions online, such as visits to advertisers' websites. For example, someone who visited the website of an online sporting goods store can receive ads about special offers from that store.
--- http://www.google.com/privacy/ads/

Google Dominates Search Advertising With 80% Market Share Unaffected By The Rise Of Bing

Posted on June 21, 2011 by Advanced Media Productions

Session handling and login



GET /index.html →

← Set-Cookie: AnonSessID=134fds1431

Protocol is HTTPS. Elsewhere just HTTP

POST /login.html?name=bob&pw=12345 →

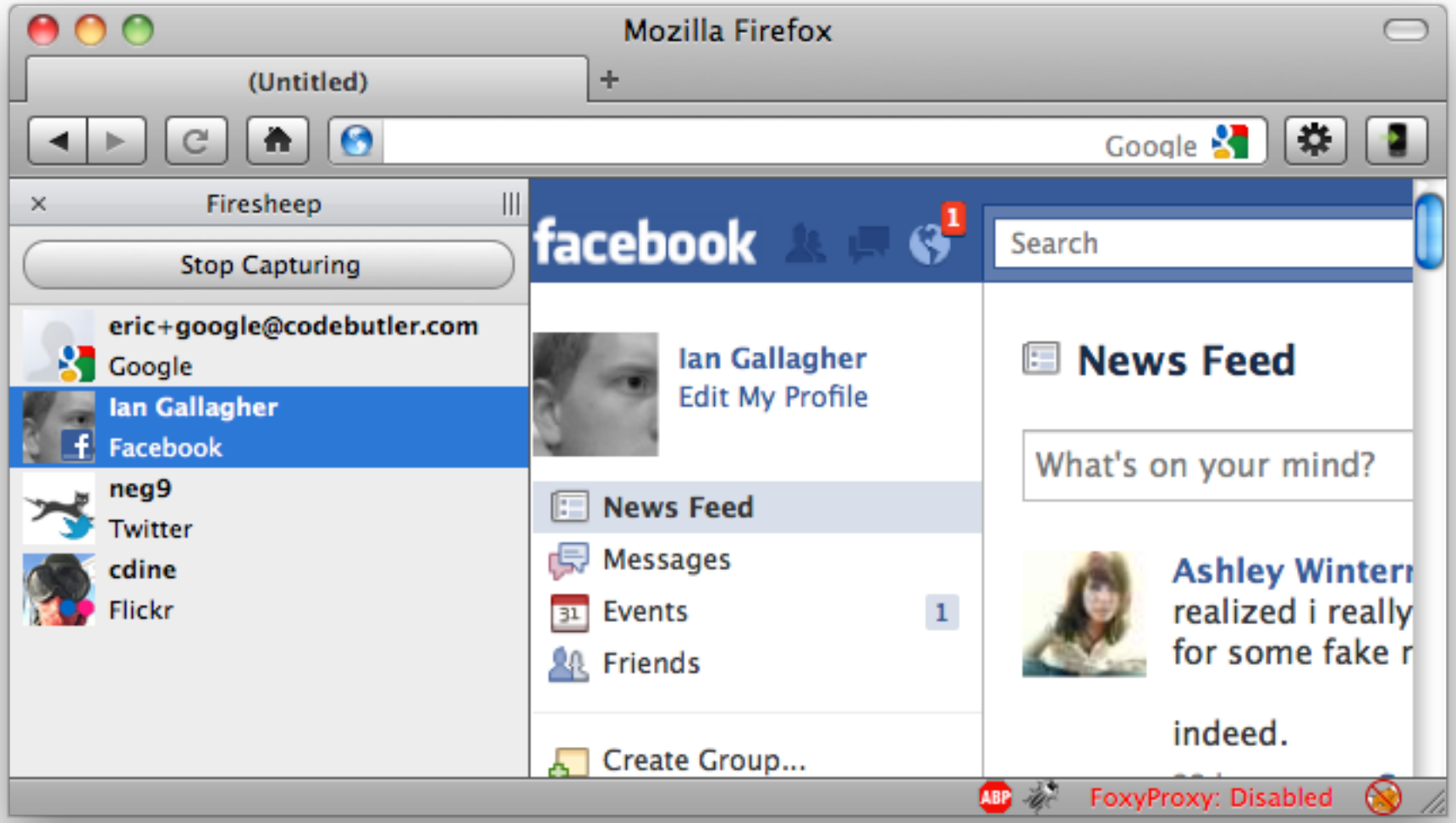
Cookie: AnonSessID=134fds1431

← Set-Cookie: SessID=83431Adf

GET /account.html →

Cookie: SessID=83431Adf

Session Hijacking



From http://codebutler.com/firesheep

Towards preventing hijacking

- Use encryption when setting session cookies
- SessID = Enc(K,info) where :
  - K is server-side secret key
  - Enc is Encrypt-then-MAC encryption scheme
  - info contains:
    - user id
    - expiration time
    - other data
- Server should record if user logs out
- Does this prevent Firesheep hijacking?
  - No
  - include in data machine-specific information
  - turn on HTTPS always

On Wednesday…

- Three prominent classes of web vulnerabilities
  - Cross-site scripting
  - Cross-site request forgery
  - SQL injection

Have a great Monday!