

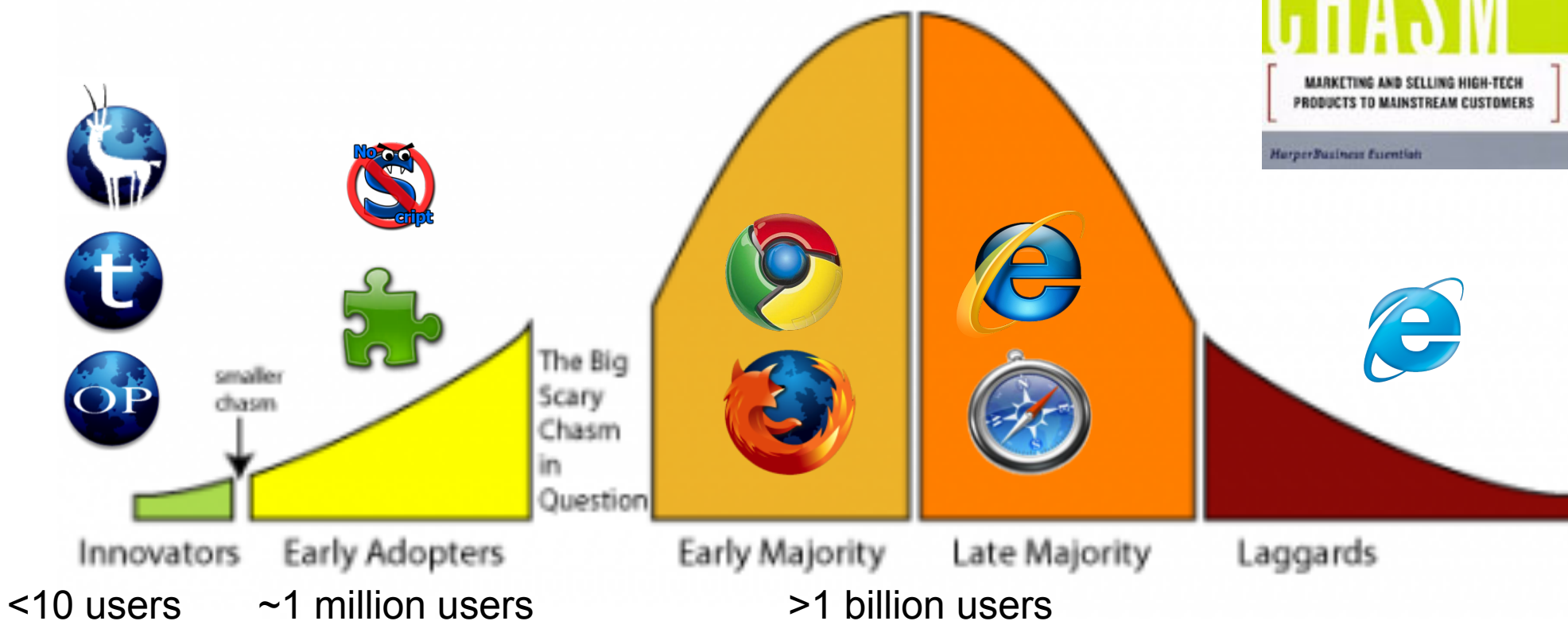
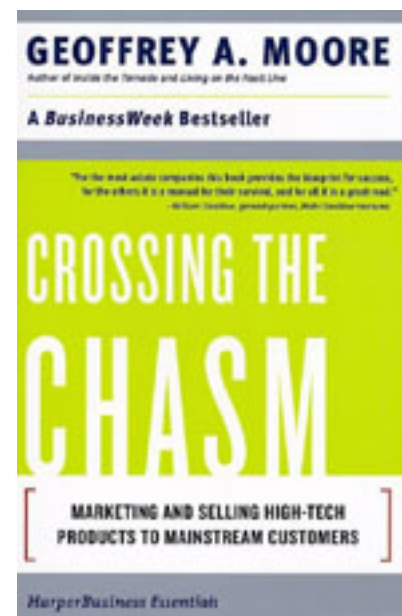
# Crossing the Chasm

Pitching Security Research to Mainstream  
Browser Vendors

Collin Jackson  
Carnegie Mellon University

# Why a security feature is like a startup

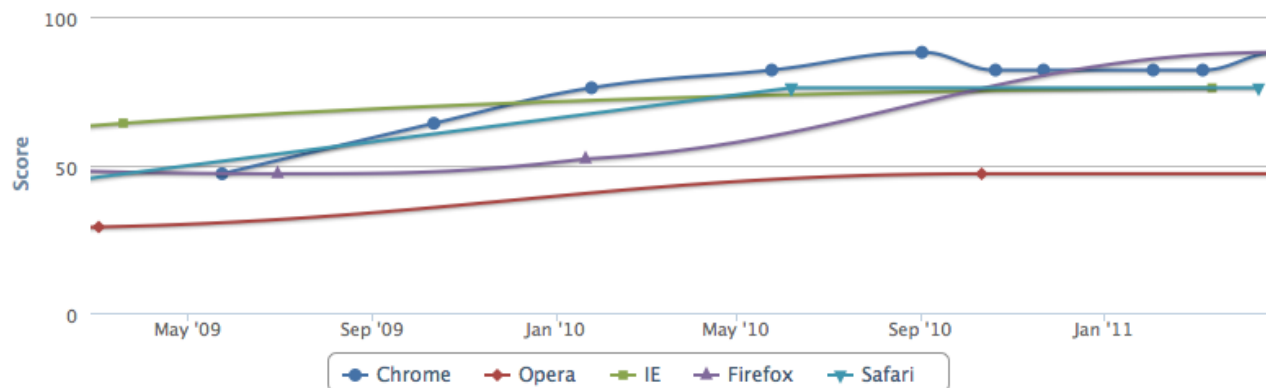
Geoffrey Moore's 'Crossing the Chasm' diagram  
circa 1991



# Ideas trying to cross the chasm



Security Score Timeline



Top Browsers		postMessage	JSON.parse	toStaticHTML	httpOnly cookies	X-Frame-Options	X-Content-Type-Options	Block reflected XSS	Block location spoofing	Block JSON hijacking	Block XSS in CSS	Sandbox attribute	Origin header	Strict Transport Security	Block cross-origin CSS attacks	Cross Origin Resource Sharing	Block visited link sniffing	Content Security Policy
<input checked="" type="checkbox"/>	Chrome 13 →	15/17	yes	yes	no	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	no
<input type="checkbox"/>	Chrome 14 →	15/17	yes	yes	no	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	no
<input type="checkbox"/>	RockMelt 0.9 →	15/17	yes	yes	no	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	no
<input type="checkbox"/>	IE 9 →	13/17	yes	yes	yes	yes	yes	yes	yes	yes	yes	no	no	no	yes	yes	yes	no
<input type="checkbox"/>	iPhone 4.2 →	13/17	yes	yes	no	yes	yes	no	yes	yes	yes	yes	yes	no	yes	yes	yes	no
<input type="checkbox"/>	Safari 5.0 →	13/17	yes	yes	no	yes	yes	no	yes	yes	yes	yes	yes	no	yes	yes	yes	no
<input type="checkbox"/>	Android 3.1 →	12/17	yes	yes	no	yes	yes	no	no	yes	yes	yes	yes	no	yes	yes	yes	no
<input type="checkbox"/>	Firefox 5 →	12/17	yes	yes	no	yes	yes	no	no	yes	yes	no	no	yes	yes	yes	yes	yes
<input type="checkbox"/>	Firefox 6 →	12/17	yes	yes	no	yes	yes	no	no	yes	yes	no	no	yes	yes	yes	yes	yes
<input type="checkbox"/>	Firefox Beta 7 →	12/17	yes	yes	no	yes	yes	no	no	yes	yes	no	no	yes	yes	yes	yes	yes
<input type="checkbox"/>	IE 8 →	11/17	yes	yes	yes	yes	yes	yes	no	yes	yes	no	no	no	yes	yes	no	no
<input type="checkbox"/>	Android 2.3 →	10/17	yes	yes	no	no	yes	no	no	yes	yes	yes	yes	no	yes	yes	no	no
<input type="checkbox"/>	iPhone 3.1 →	8/17	yes	no	no	yes	yes	no	no	yes	yes	no	yes	no	no	no	yes	no
<input type="checkbox"/>	Opera 11 →	8/17	yes	yes	no	yes	yes	no	no	yes	yes	no	no	no	yes	no	no	no

For every idea here there are 100 that never got **any** adoption

# Good ideas get adopted very quickly

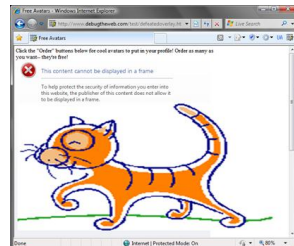
## X-Frame-Options

## History Privacy

Top Browsers	score ↓	postMessage	JSON.parse	toStaticHTML	httpOnly cookies	X-Frame-Options	X-Content-Type-Options	Block reflected XSS	Block location spoofing	Block JSON hijacking	Block XSS in CSS	Sandbox attribute	Origin header	Strict Transport Security	Block cross-origin CSS attacks	Cross Origin Resource Sharing	Block visited link sniffing	Content Security Policy
<input checked="" type="checkbox"/> Chrome 13 →	15/17	yes	yes	no	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	no
<input type="checkbox"/> Chrome 14 →	15/17	yes	yes	no	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	no
<input type="checkbox"/> RockMelt 0.9 →	15/17	yes	yes	no	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	no
<input type="checkbox"/> IE 9 →	13/17	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	no	no	no	yes	yes	yes	no
<input type="checkbox"/> iPhone 4.2 →	13/17	yes	yes	no	yes	yes	no	yes	yes	yes	yes	yes	yes	no	yes	yes	yes	no
<input type="checkbox"/> Safari 5.0 →	13/17	yes	yes	no	yes	yes	no	yes	yes	yes	yes	yes	yes	no	yes	yes	yes	no
<input type="checkbox"/> Android 3.1 →	12/17	yes	yes	no	yes	yes	no	no	yes	yes	yes	yes	yes	no	yes	yes	yes	no
<input type="checkbox"/> Firefox 5 →	12/17	yes	yes	no	yes	yes	no	no	yes	yes	yes	no	no	yes	yes	yes	yes	yes
<input type="checkbox"/> Firefox 6 →	12/17	yes	yes	no	yes	yes	no	no	yes	yes	yes	no	no	yes	yes	yes	yes	yes
<input type="checkbox"/> Firefox Beta 7 →	12/17	yes	yes	no	yes	yes	no	no	yes	yes	yes	no	no	yes	yes	yes	yes	yes
<input type="checkbox"/> IE 8 →	11/17	yes	yes	yes	yes	yes	yes	yes	no	yes	yes	no	no	no	yes	yes	no	no
<input type="checkbox"/> Android 2.3 →	10/17	yes	yes	no	no	yes	no	no	yes	yes	yes	yes	yes	no	yes	yes	no	no
<input type="checkbox"/> iPhone 3.1 →	8/17	yes	no	no	yes	yes	no	no	yes	yes	yes	no	yes	no	no	no	yes	no
<input type="checkbox"/> Opera 11 →	8/17	yes	yes	no	yes	yes	no	no	yes	yes	yes	no	no	no	yes	no	no	no

Two years after

One year after

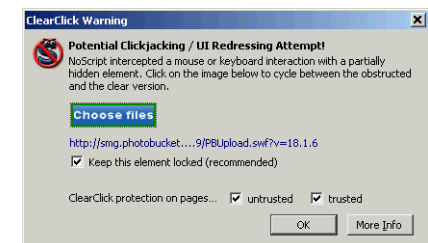
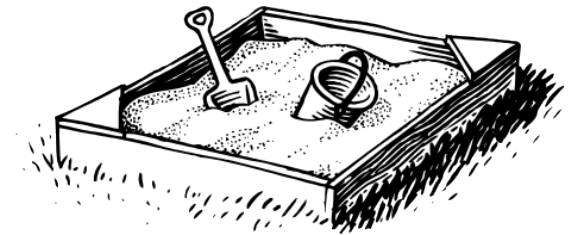


Preventing attacks on a user's history through CSS :visited selectors

*L. David Baron, Mozilla Corporation*

# Not all ideas are so lucky...

- Browser-based identity management
  - Password generators
  - Client certs
  - PAKE
- Fine-grained sandbox architectures
  - Plugin isolation
  - Origin isolation
- Automatic clickjacking protection
  - Wait, what?



# NoScript has 90m downloads!



- Less than  $<0.1\%$  of active internet users
- Dumping ground for chasm-challenged features
- Fundamentally different outlook than mainstream browsers
  - Extensive user interaction
  - Highly complex behavior
  - Breaks sites... by design!



[flattr.com/profile/ma1](http://flattr.com/profile/ma1)





# Are browser vendors too conservative?



- Features are not free!
  - Simplicity as a selling point
  - Rely on addons for niche functionality
- Breakage is **very** expensive
  - Web sites slow to adapt
  - Switching costs are low

## *Browser Not Supported*

---

The browser you are using is not supported by this application. If you wish to use this application, please use one of the links below to download and install the current version of a supported browser:

[Microsoft Internet Explorer](#)



[Netscape](#)



[Firefox](#)



# What program committees care about

- Novel
  - Not substantially similar to previous work
  - Opens new avenues of research
  - Unconstrained by conventional thinking
- Non-trivial
  - Makes clever use of advanced tools and techniques
  - Substantial work involved in system implementation



These will get you a conference paper...

... but they **actively harm** a proposal's mainstream appeal



# What browser vendors care about

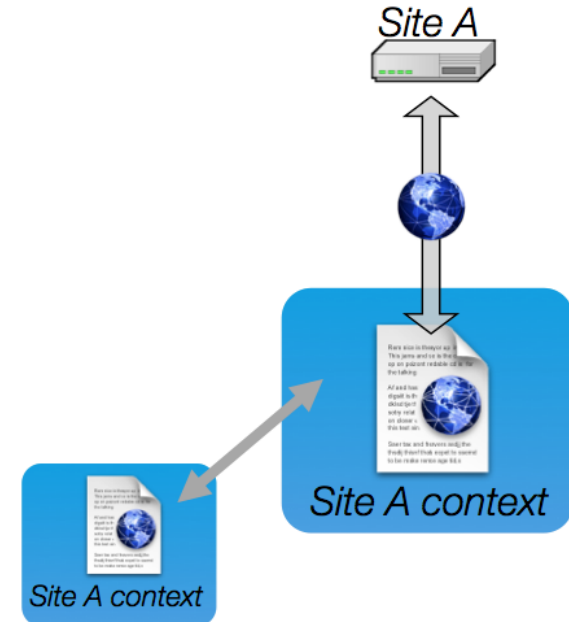
- Must-have
  - Replaces broken, band-aid approaches that are nevertheless *already being widely used*
  - No browser wants to be the only one without it
- Easy
  - Deployable *unilaterally*, with little effort
  - Everyone can implement in the same way
  - Can determine if implementation is correct
- Low-risk
  - Doesn't break anything important, even in the long tail
  - Any change that's not opt-in is risky

# Make your proposal a *must-have*

- Can always find **someone** who likes your idea...
  - Early adoption not a sure-fire sign of mainstream need
- Addons are a final resting place for many niche features
  - A vendor needs to be embarrassed **not** to have it
  - Browser vendors are like dominos
- Marketing
  - Compelling demos
  - Mainstream press
  - Large web sites who will champion it

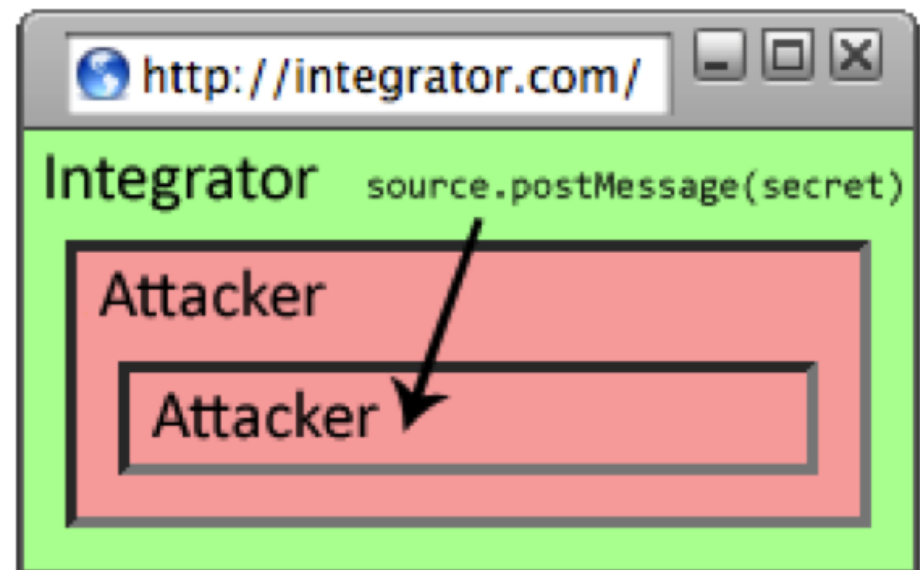
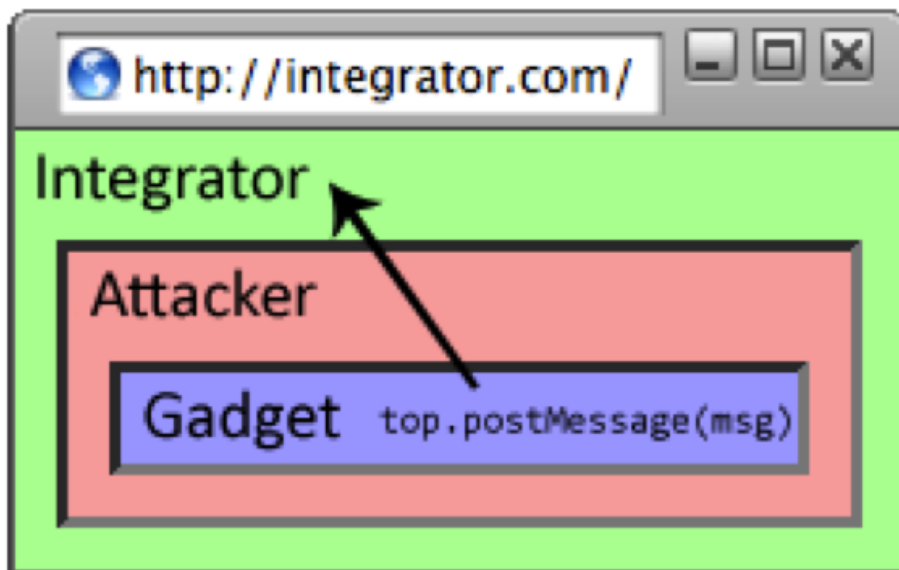
# Must-have #1: Same-origin policy

- Origin = protocol://host:port
- Full access to same origin
  - Full network access
  - Read/write DOM
  - Storage
- Limited interaction with other origins
  - Import of library resources (e.g. scripts)
  - Forms, hyperlinks
- Introduced by Netscape in 1996 in response to media reports of cross-origin scripting attacks



# How postMessage became must-have

- Allows client-side messaging between origins
- Increasingly popular web sites like Facebook build mechanisms around hacks (fragment identifier messaging)
- Microsoft decided it was safe, implemented in IE8
- Firefox wanted HTML5 feature parity with IE
- Safari wanted HTML5 feature parity with Firefox/IE
- By the time we dropped this bomb, it was too late to stop it



# How history privacy became must-have

## WHAT THE INTERNET KNOWS ABOUT YOU

- Compelling demos
- Real-world attacks
- Lawmakers and media interested

Perfect ingredients for competition among browser vendors

- Only partial solution but easy and low-risk

### YouPorn sued for sniffing browser history



By: Stephen Shankland

DECEMBER 6, 2010 8:12 AM PST

 Print  E-mail

 Recommend

128

 Tweet

146

 +1

1

 Share

 48 comments

A site for sharing pornographic content is the target of a lawsuit accusing it of improperly checking what other Web sites visitors had used.



Plaintiffs David Pitner and Jared Reagan, both of Newport Beach, Calif., accuse YouPorn operator Midstream Media of the Netherlands of violating the U.S. Computer Fraud and Abuse Act and California's computer crime law; of engaging in deceptive and unfair business practices; and of unlawful and unfair competition.

The suit, filed Friday in U.S. District Court for the central district of California, accuses YouPorn of, among other things, "intentionally accessing plaintiffs'...computers without authorization." The plaintiffs are seeking class-action status, an injunction to stop the history sniffing practice, and payment for damages.

YouPorn didn't immediately respond to a request for comment.

The nub of the issue, as mentioned above, is a practice called history sniffing. Browsers generally keep track of what Web sites a person has visited, showing the links in different colors depending on whether they've visited or not. Browser sniffing essentially asks the browser what color should be used for various links; the answer can reveal the browsing history.

# Make your proposal easy

Strongly preferable:

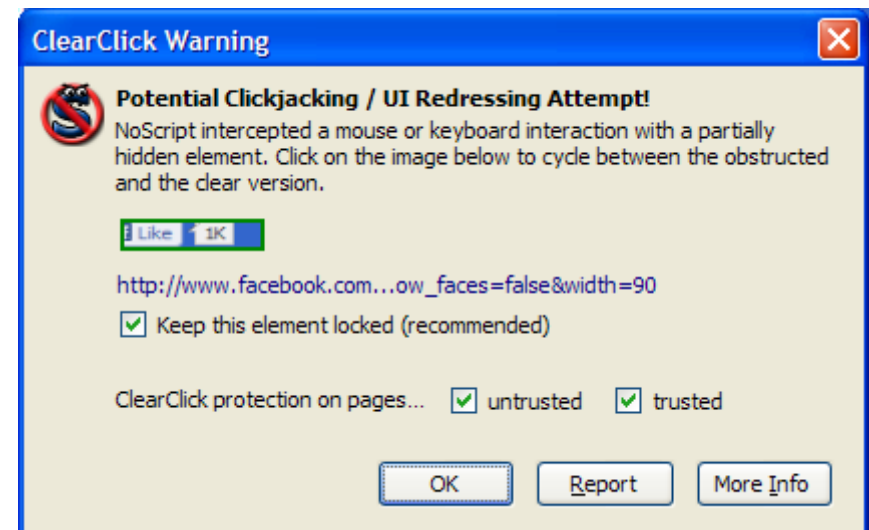
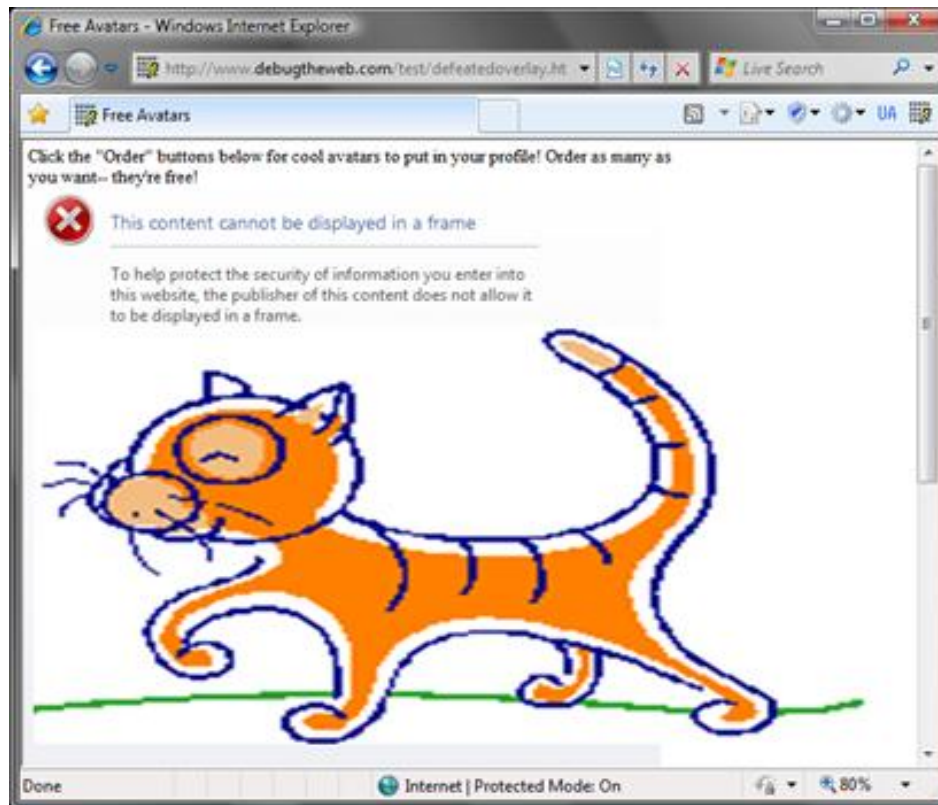
- Deployable unilaterally: doesn't require cooperation among multiple vendors
- Web sites don't have to adopt right away
- Everyone can implement it exactly the same

Non-examples

- Taint tracking
- toStaticHTML
- DNSSEC



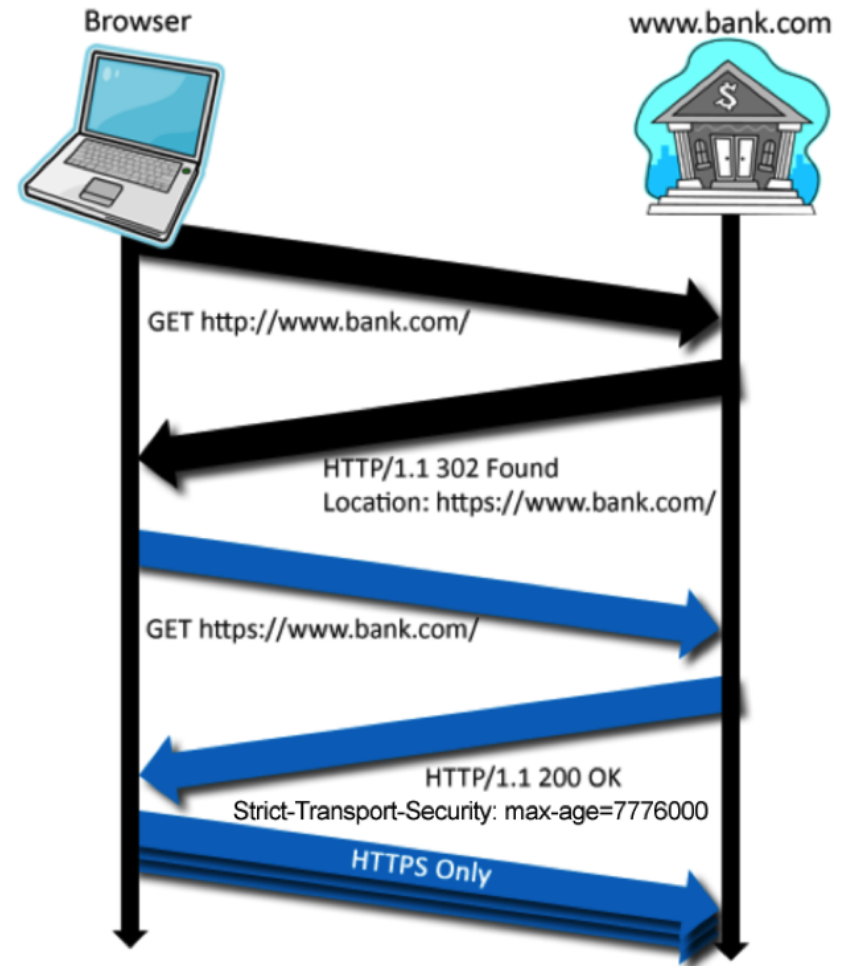
# X-Frame-Options versus ClearClick



# Strict Transport Security

Original ForceHTTPS involved

- Cookies
- User-configurable options
- Mixed content protection



Stripped down proposal to make it easier to implement

# Make your proposal *low-risk*

- Does it break functionality?
- Does it slow things down?
- Does it interfere with getting stuff done?
- Are you making more people sad than happy?



# De-risking a security proposal

Choose one:

1. Make the security opt-in

- Huge evangelism cost
- Yet another thing to forget to do

1. Create brand new functionality

- Sidesteps legacy considerations
- Adoption barrier?

**2. Very** thorough performance & compatibility evaluation

- Often ~5x harder than the actual implementation
- Some features just weren't meant to be!

# Opt-in security

X-Secure-Me-Harder: yes!

- Extremely popular approach! X-Frame-Options, X-Content-Type-Options, Strict-Transport-Security, etc.
- Header bloat problem

How many opt-in features had an impact on the world?

- Trickling down from the PayPals and Twitters
- Long tail takes many years

Alternative policy delivery mechanisms

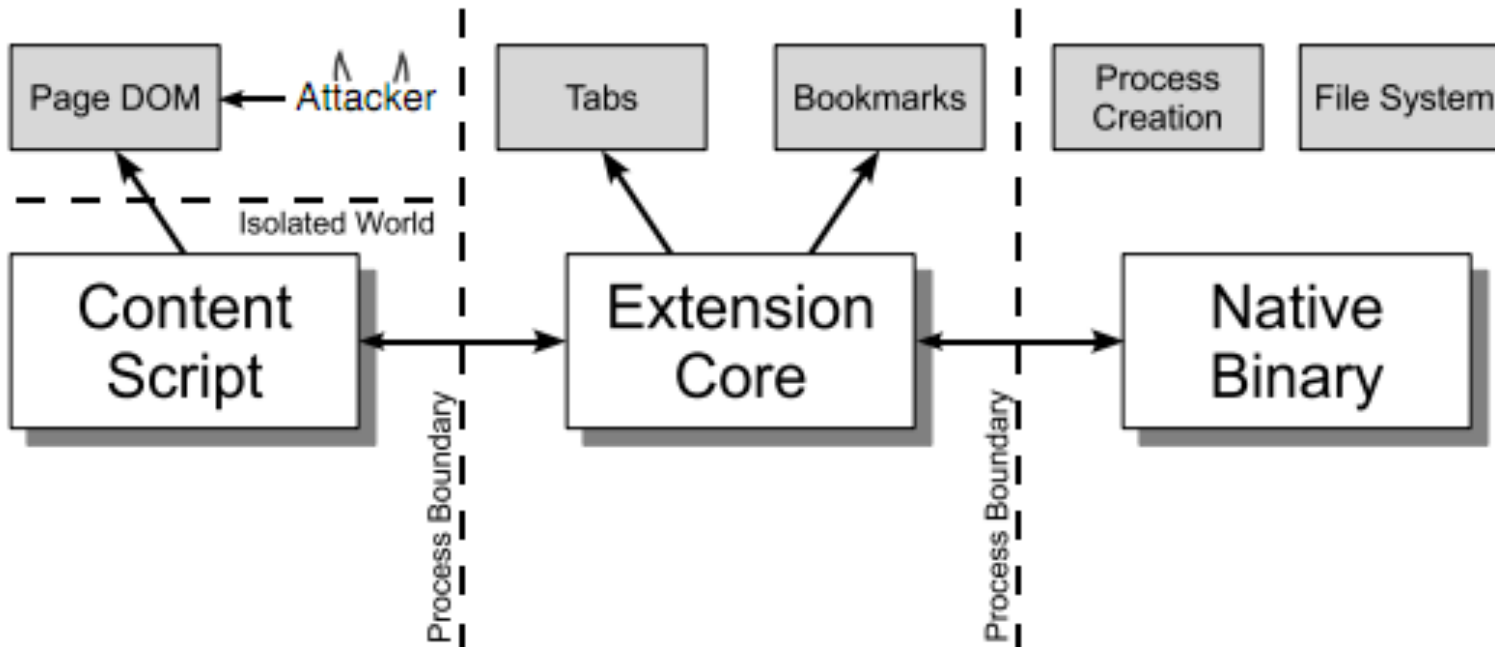
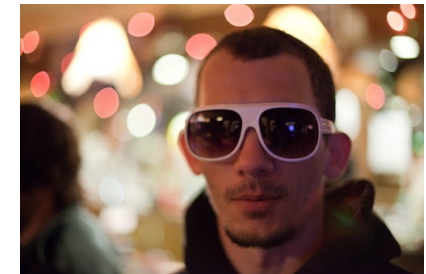
- Host-meta
- New HTML tags/attributes
- Content Security Policy

# New platforms

 chrome web store

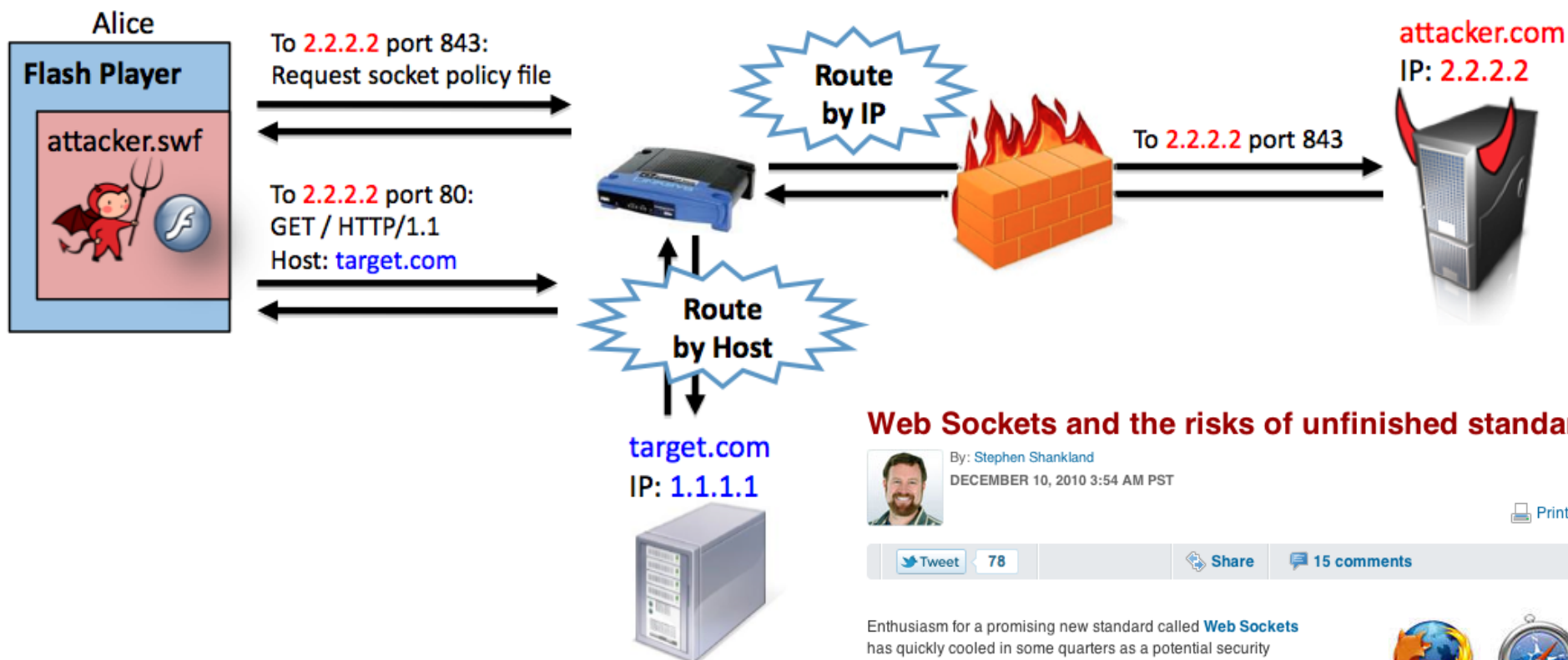
collin.jackson@gmail.com ▾

- Apps
- Education
- Entertainment
- Family
- Games
- Lifestyle
- News & weather
- Productivity
- Shopping
- Social & communication
- Utilities





# Web Sockets



## Web Sockets and the risks of unfinished standards



By: Stephen Shankland  
DECEMBER 10, 2010 3:54 AM PST

 Print  E-mail

 Tweet 78

 Share

 15 comments

Enthusiasm for a promising new standard called **Web Sockets** has quickly cooled in some quarters as a potential security problem led some browser makers to hastily postpone support.

The Web Sockets technology, which opens up a live communication link between a browser and a server, remains an important part of plans to make the Web a home for more dynamic, interactive sites. It could, for example, speed up Google Instant searching and multiplayer games. But **Mozilla** and **Opera** put their Web Socket plans on hold this week until the wrinkles are ironed out.

The reversal is only the latest difficulty, though. Web Sockets development already had become somewhat contentious as eager browser makers--Google in particular--began including support for a specification they knew wasn't done. Overall, the Web Sockets history illustrates some pitfalls of the style and pace of Web standards development.



# On-by-default security? Yikes.

- Things fail mysteriously, and more often than you'd think
- Failures are (usually) not attacks
- For every bug filed, how many users just give up or switch browsers?

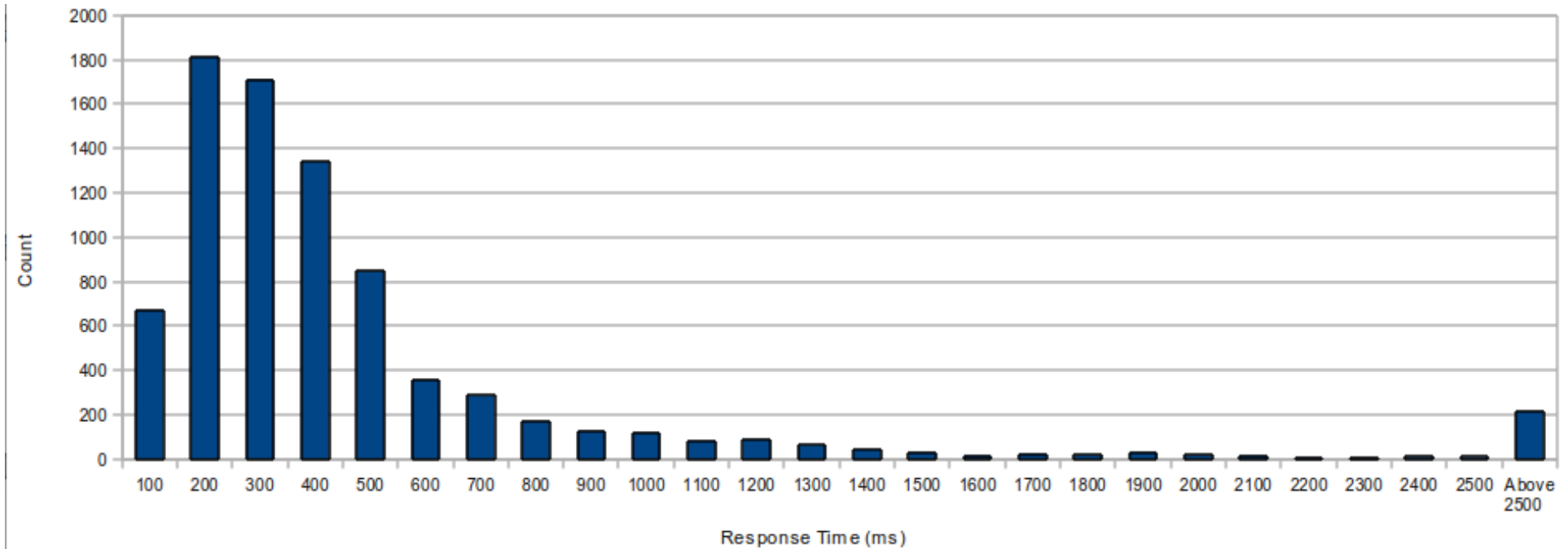


# How securing Gmail ruined my Korean class

- Get a website to host your SWF  
`http://victim.com/attack.swf`
- User logs in to victim.com
- Get user to visit  
`http://attack.com/`
- Embed the SWF and hijack the session  
`<embed src="http://victim.com/attack.swf"/>`



# Another on-by-default fail: OCSP



- Validating certificate takes >1sec for 10% of HTTPS requests
- Adds to initial page load time dramatically when dependent scripts, images, etc. are on other hostnames
- Must-have, yet high-risk. Browsers don't enforce
- Defeating OCSP with the number 3

# Compatibility Evaluation Failures

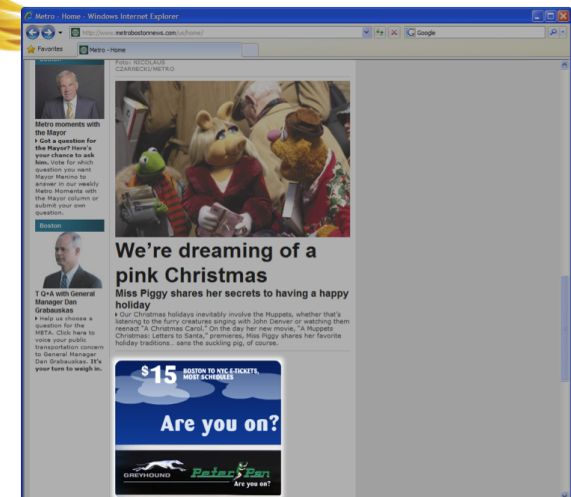
"I checked the Alexa top 100"

"I changed the plugin security policy and I played a YouTube video"

"I went to 10 websites and only 2 of them broke"

# Better ideas

- *Deep crawl*
  - *Get beyond login pages*
  - *Execute JavaScript (Kudzu)*
- *Client-side measurement*
  - *Google Chrome User Metrics*
  - *Firefox Test Pilot*
- *Ad networks*
  - *Flash Player ads*
  - *Iframe ads*





# What a real evaluation looks like

## Content Sniffing Algorithm

- Searched the entire Google crawl index for common mime type mismatches; eliminated unused sniffing rules
- QA team visited the top 500 sites and tested extensively while logged in
- Google Chrome user metrics study found less than 0.004% compatibility impact



# If you don't have the Google index...

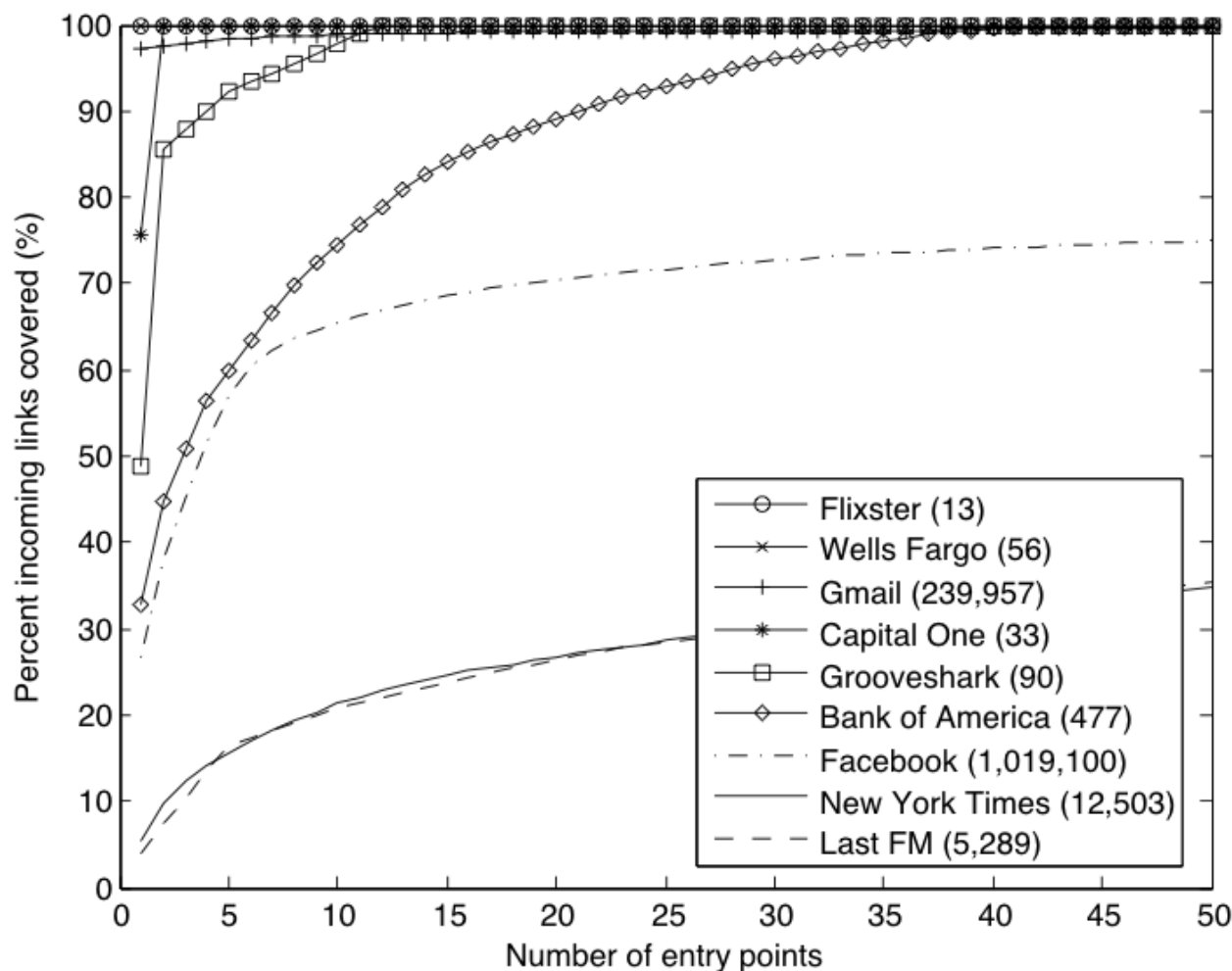


Alexa top 100,000

Requesting Server	Rendering Mode	Total	HTTP Error	Correct Type		Incorrect Type	
				Well-formed	Mal-formed	Well-formed	Mal-formed
Same-origin	Standards	180,445	1,497	178,017	506	424	1
	Quirks	25,606	455	24,445	332	304	59
Cross-origin	Standards	47,943	347	47,345	104	147	0
	Quirks	6,075	53	5,891	57	74	0
Total		260,069	2,363	255,698	999	949	60

CSS references blocked by **strict enforcement** or **minimal enforcement**

# If you don't have your own browser...

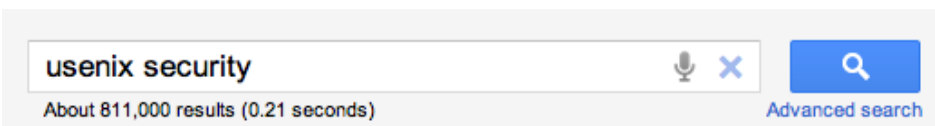




# Sometimes the answer is "no"

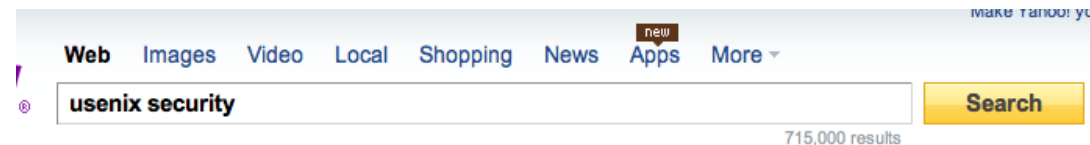


## Stanford SafeHistory

- Only showed links as visited if you visited from the current site
- Perfect protection from attack, but at what cost?



- [USENIX Security '11](#)  
- [www.usenix.org/sec11/](http://www.usenix.org/sec11/) - Cached
- Don't miss the 20th **USENIX Security** Symposium, August 8–12, 2011, in San Francisco, CA. **USENIX Security '11** brings together researchers, practitioners, ...
- |                               |                                 |
|-------------------------------|---------------------------------|
| <a href="#">Tech Sessions</a> | <a href="#">Call for Papers</a> |
| <a href="#">Registration</a>  | <a href="#">Organizers</a>      |
| <a href="#">Workshops</a>     | <a href="#">At a Glance</a>     |
| <a href="#">Hotel/Travel</a>  | <a href="#">Training</a>        |
- [More results from usenix.org »](#)



- Also try: [usenix security symposium](#), [usenix security 2011](#), [more...](#)
- [USENIX: The Advanced Computing Systems Association](#)  
HotSec '11: 6th **USENIX** Workshop on Hot Topics in **Security** Program now available!  
MetriCon 6.0: Sixth Workshop on **Security** Metrics; Register online today for **USENIX Security** ...  
[usenix.org](http://usenix.org) - Cached
- [18th USENIX Security Symposium \(USENIX Security '09\)](#)  
Thank You for Coming! Thanks to those of you who joined us in Montreal for **USENIX Security '09**. Check out the videos of the talks, online proceedings, and slides here.  
[www.usenix.org/events/sec09](http://www.usenix.org/events/sec09) - Cached

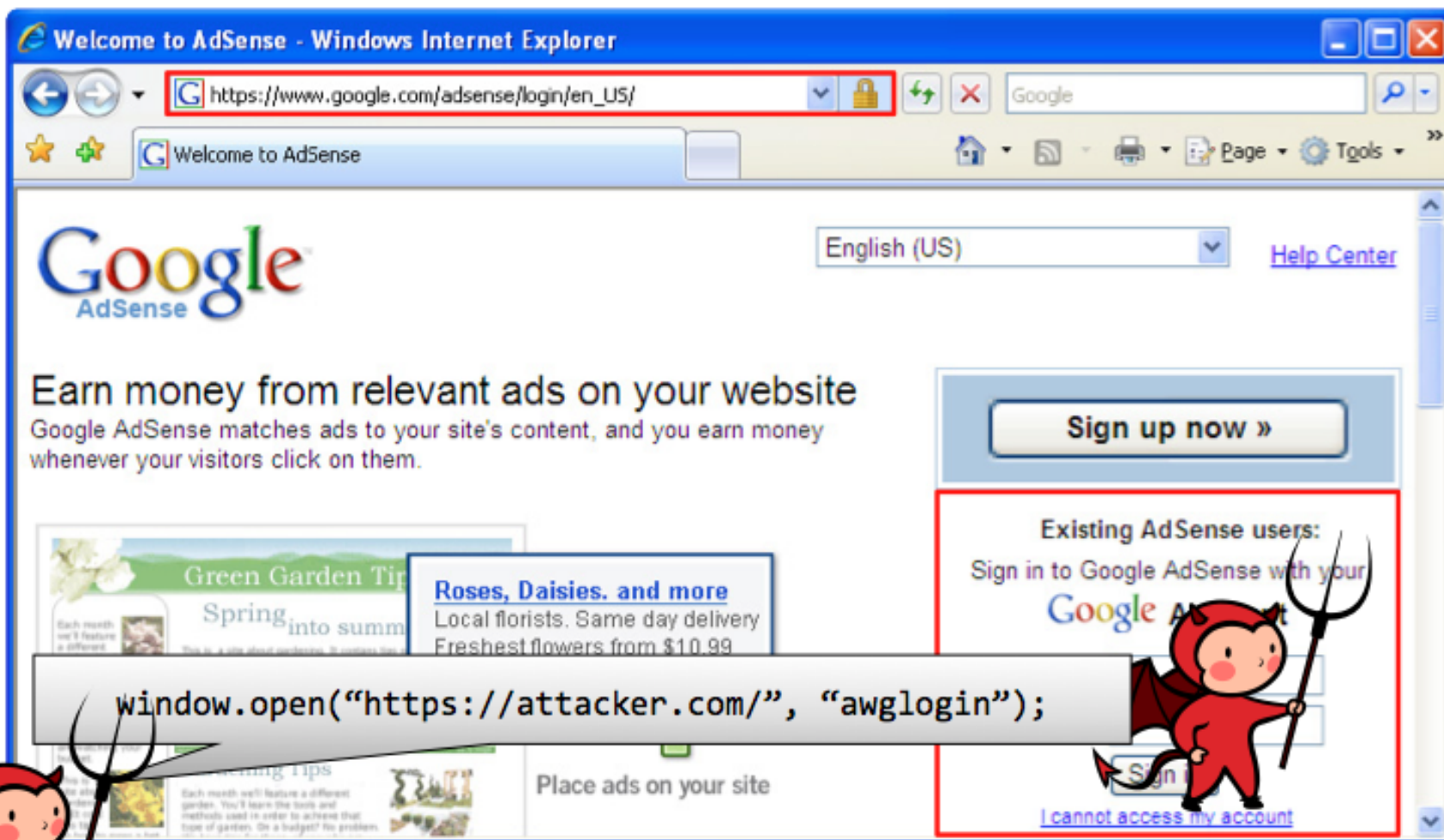
# Compatibility numbers aren't good?

An unlikely savior...





# Frame navigation



Welcome to AdSense - Windows Internet Explorer

https://www.google.com/adsense/login/en\_US/

Welcome to AdSense

Google AdSense

English (US) Help Center

Earn money from relevant ads on your website  
Google AdSense matches ads to your site's content, and you earn money whenever your visitors click on them.

Sign up now »

Existing AdSense users:  
Sign in to Google AdSense with your Google Account

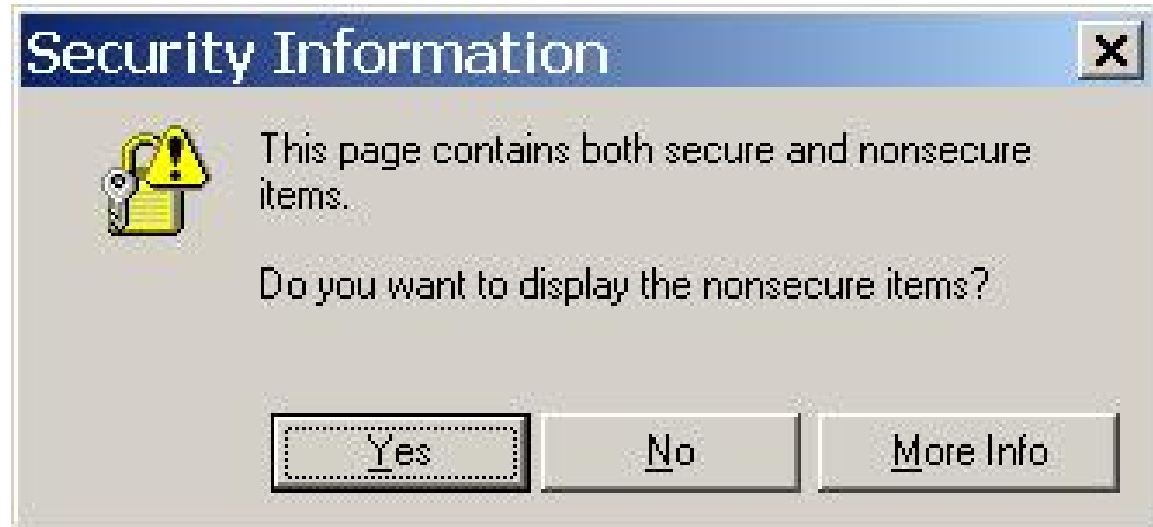
```
window.open("https://attacker.com/", "awglogin");
```

Sign in

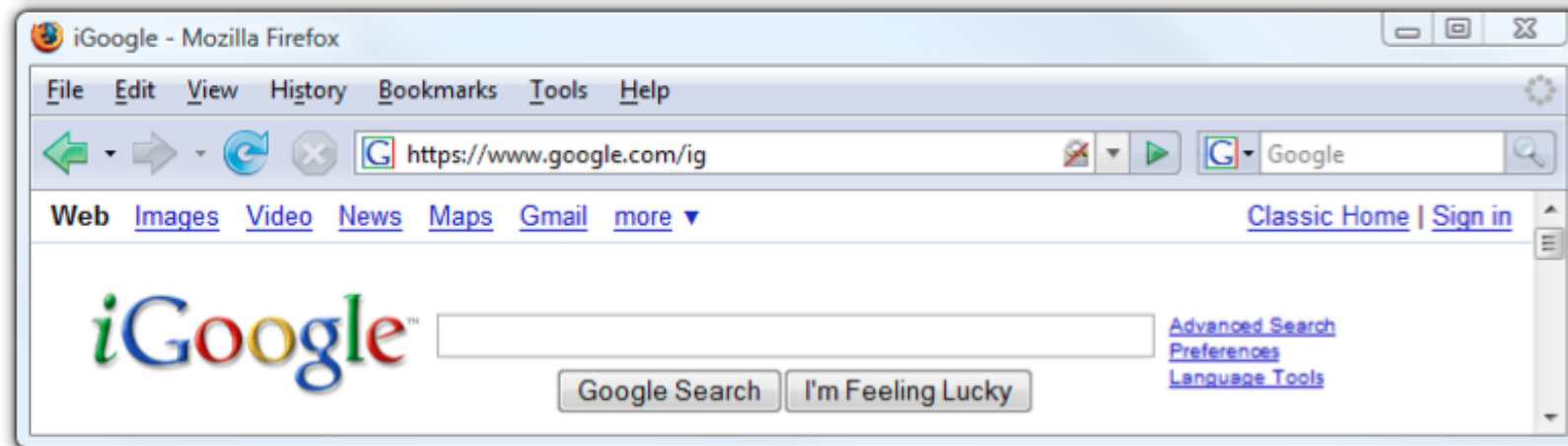
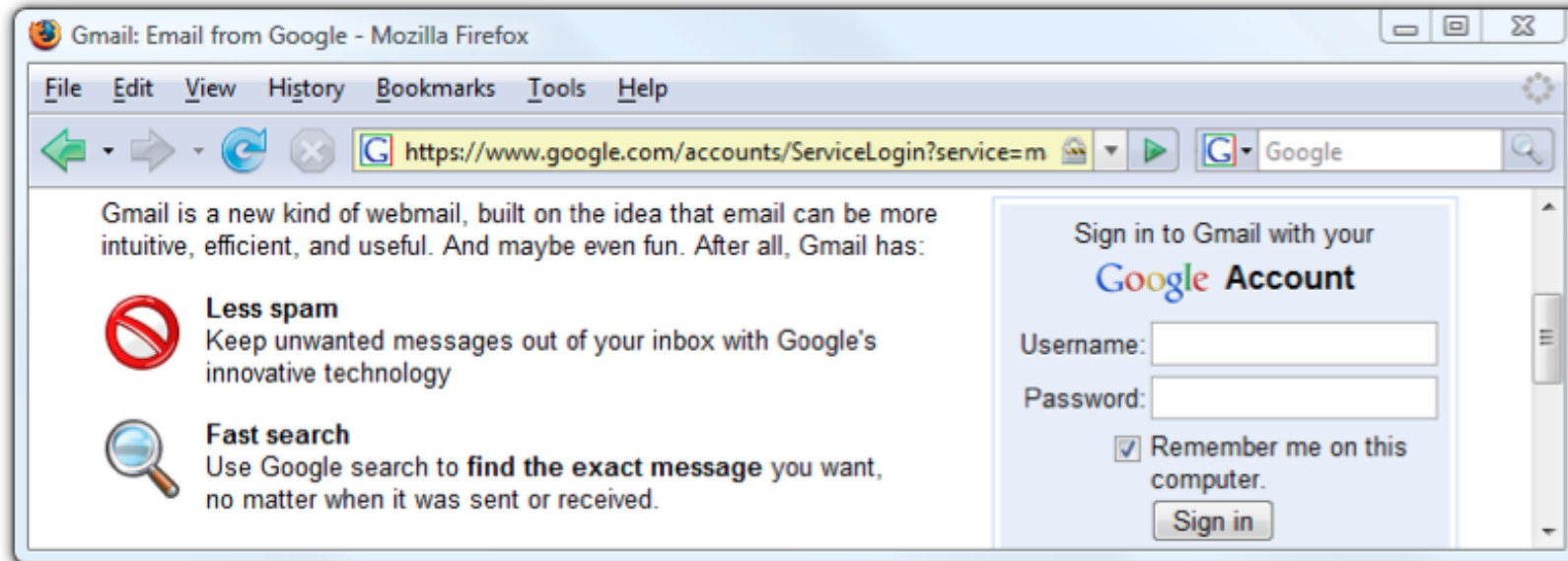
I cannot access my account



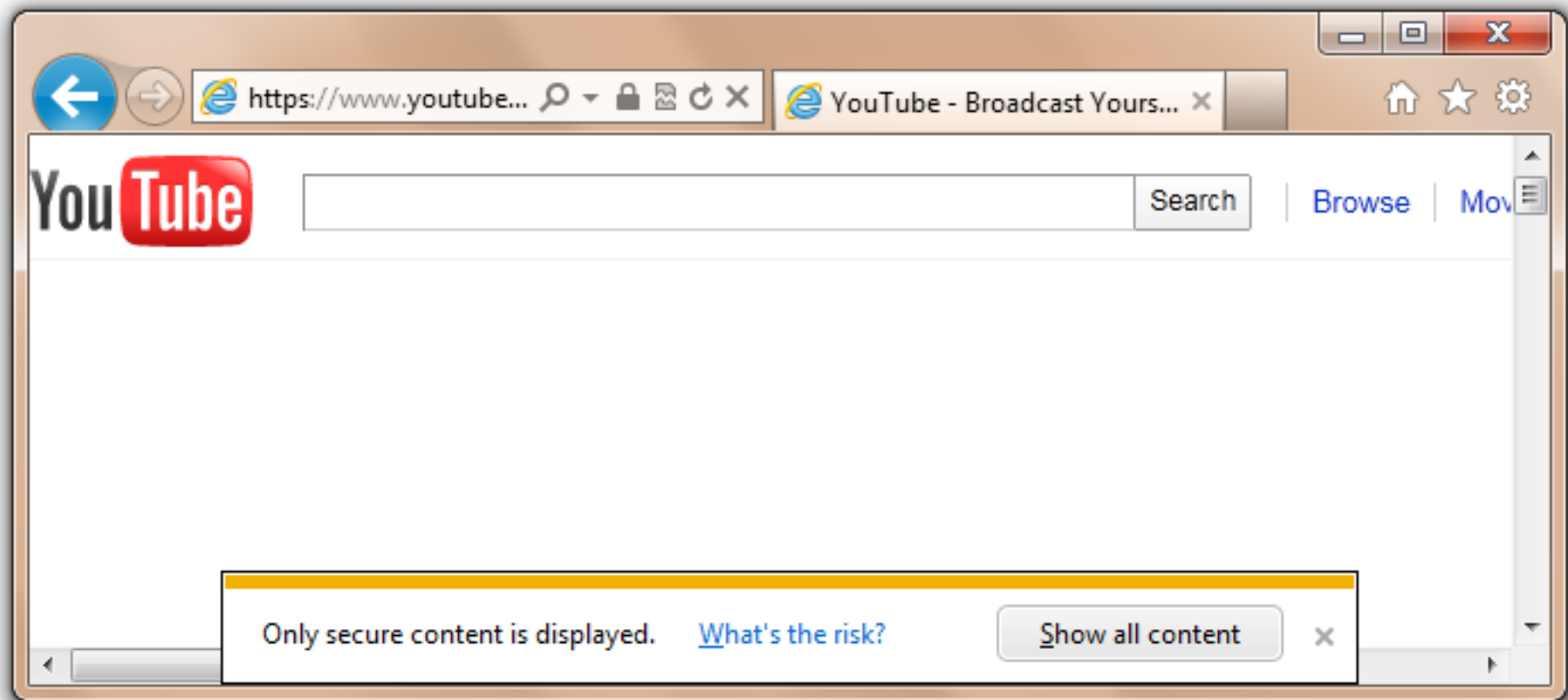
# Mixed content



# Origin contamination

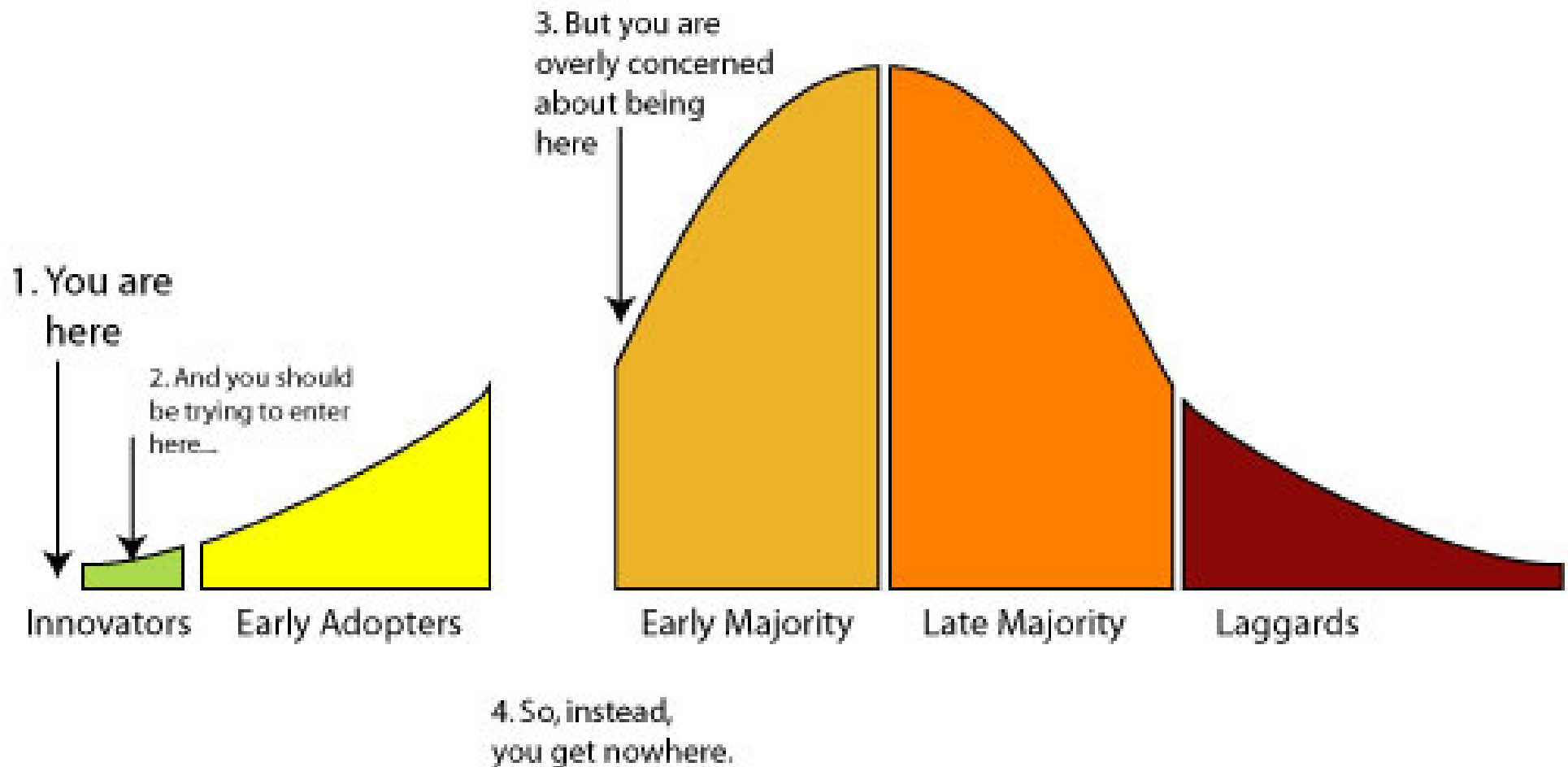


# Taking one for the team



# Dangers of chasm thinking

Miss Rogue's Currently Frustrating Chasm Dialogue  
circa 2006



Should researchers bother with

nice-to-have,  
difficult,  
risky

ideas?

# Yes!

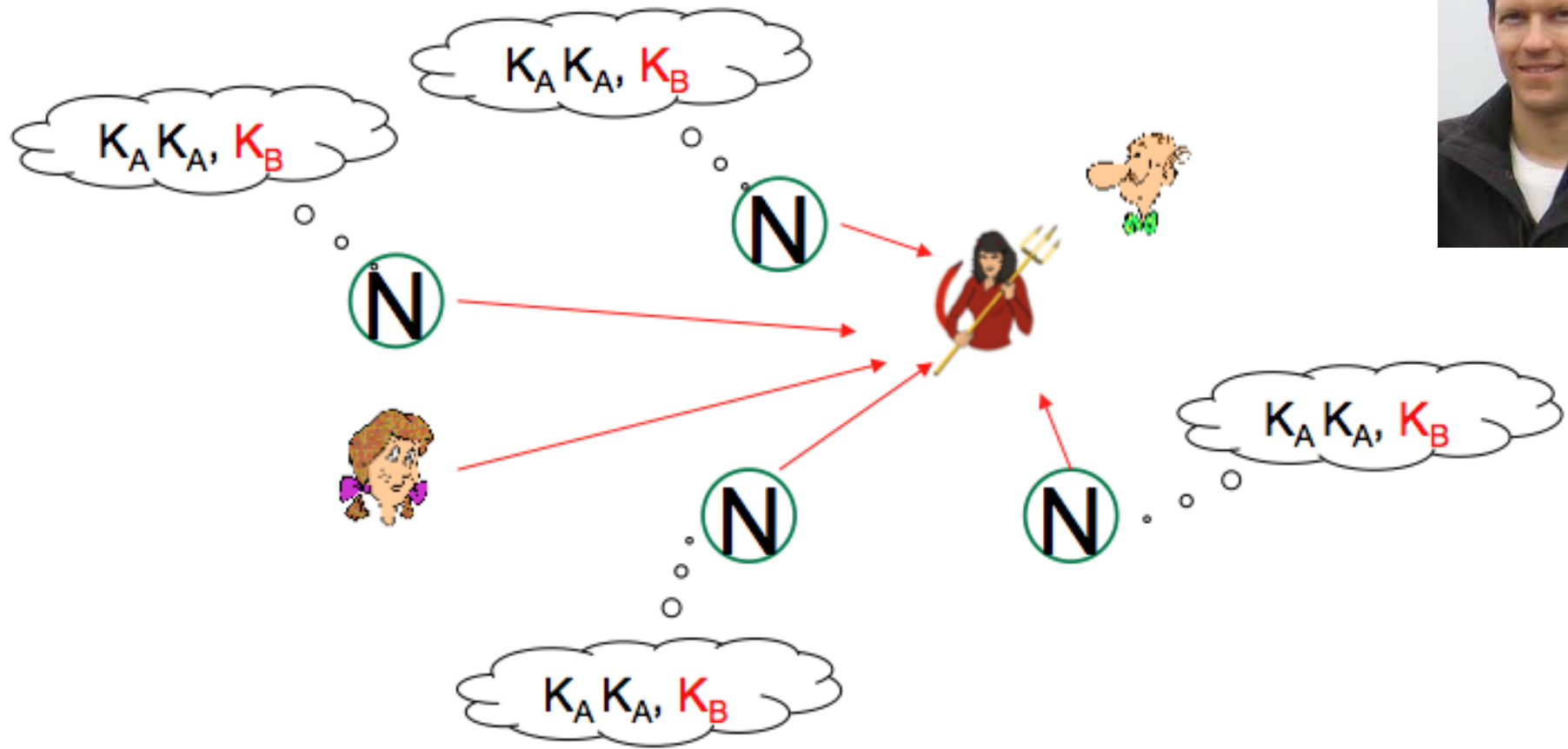
but...



# Let your idea be hacked apart

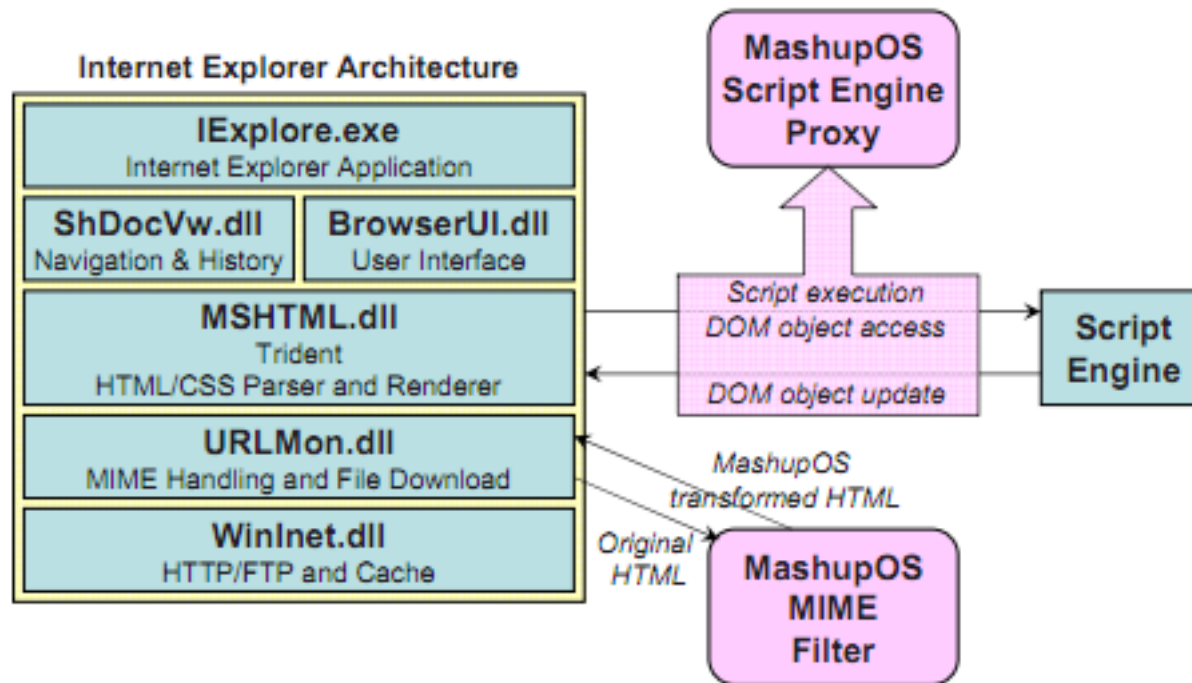
- Don't expect the final solution to resemble the original form
  - Rebranded
  - User interface changed/removed
  - Unnecessary complexity dropped
- The best ideas are easily tweaked and repurposed
- Sometimes just a problem statement is a contribution
- Celebrate indirect impact!

# Perspectives



- Firefox addon topped out at ~10,000 users
- Crossed the chasm in another form:  
~100,000,000 Chrome users benefiting from HTTPS monitoring

# MashupOS



- Never went anywhere in original form
- Key ideas survived
  - `postMessage(message, targetOrigin)`
  - `text/html-sandboxed` MIME type
- Gazelle may find a similar fate

# What you can do right now to help

- Analyze existing new proposals in standardization
- Catch problems before legacy concerns creep in
- **WebRTC** - direct network communication between web clients
- **Component Model** - lets you construct your DOM out of mutually distrusting components with security boundaries between them
- **ECMAScript 6** - have untrusted JavaScript run in your page
- **Content Security Policy** - protect the developer from themselves
- **Web Intents** - allow one web application to invoke another

The time to get involved is now!

# Show up!

- Meet the decision-makers
  - Many are in this room!
  - Many Mozilla meetings are open
- Join mailing lists
  - WHATWG
  - W3C public-web-security
  - IETF WebRTC
  - IETF Web Security Working Group
- Write code!
  - Firefox, Google Chrome, and most of Safari are open source
  - Nothing says "implement me now" like a patch ready for approval





collin.jackson@sv.cmu.edu  
<http://websec.sv.cmu.edu/>

# Controversial things I just said

- NoScript is a niche browser... not the browser of the future
- Program committees actively harm good ideas
- OCSP is risky
- Taint tracking is hard
- SafeHistory is undeployable
- Breaking web sockets for 6 months was not a mistake
- You should crash Mozilla team meetings