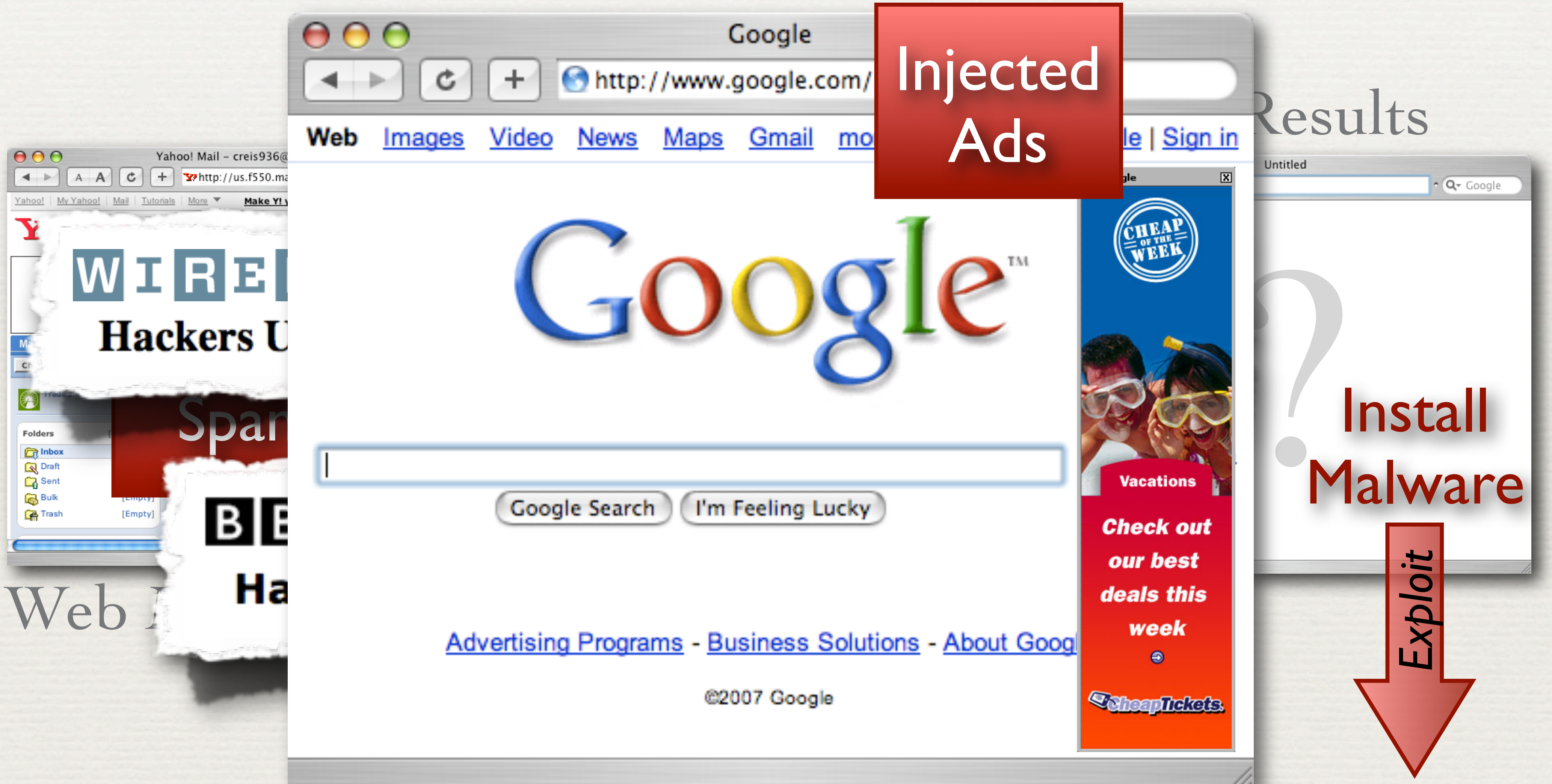
Building a Safer Web:
Web Tripwires &
A New Browser Architecture

Charles Reis
University of Washington

Web Browsing isn't Safe



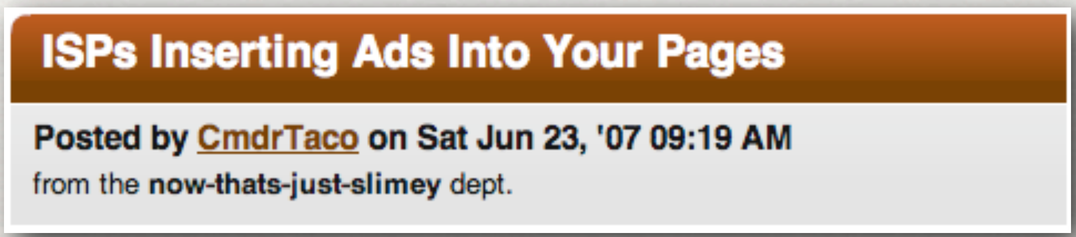
This Talk:

- Focus on one problem: **in-flight page changes**
  - Recent study shows undesirable changes
  - Publishers can detect with Web Tripwires
- Broader view of **safe browser architectures**
  - On-going research at UW CSE

1. In-Flight Page Changes & Web Tripwires

Joint work with
Steve Gribble, Yoshi Kohno, Nick Weaver

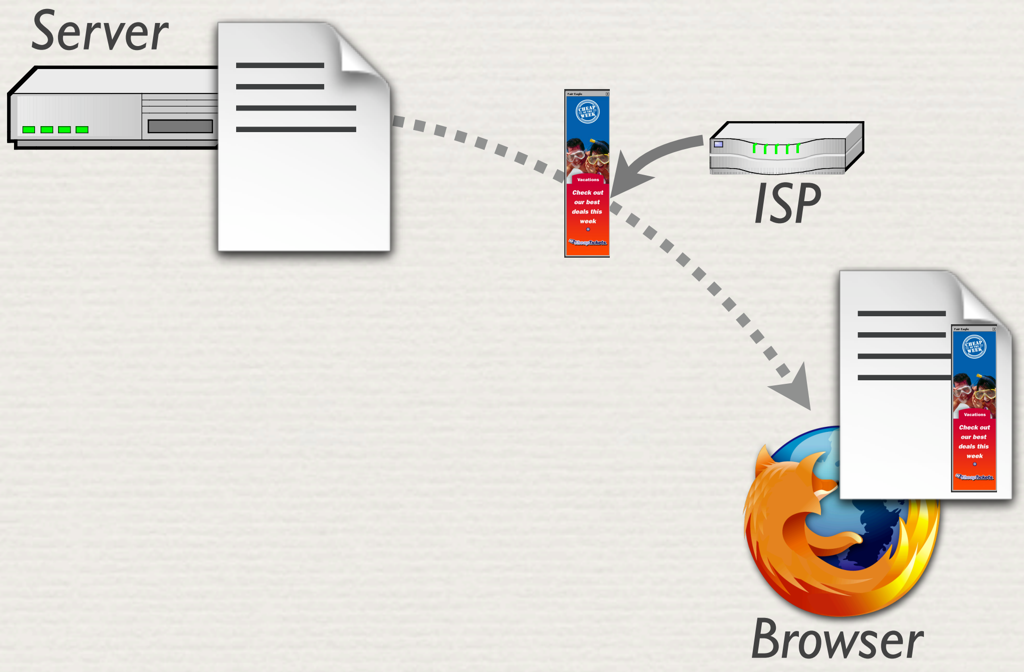
ISP-Injected Ads



ISPs Inserting Ads Into Your Pages

Posted by **CmdrTaco** **on Sat Jun 23, '07 09:19 AM**

from the **now-thats-just-slimey** dept.



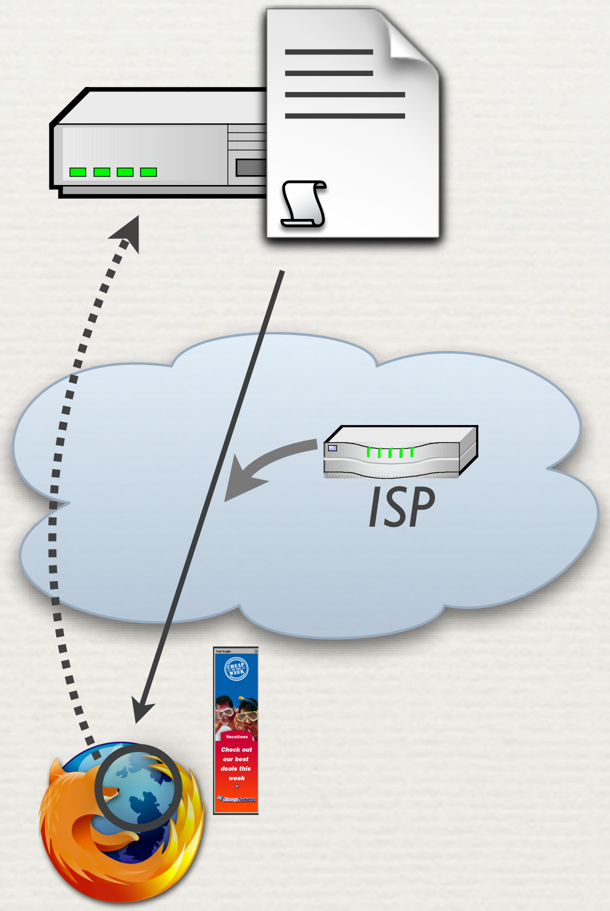
- Surprising reports of web page modifications
- How often does this occur?

Outline

- Detecting In-Flight Changes
- Measurement Results
- Dangerous Consequences
- Web Tripwires for Publishers

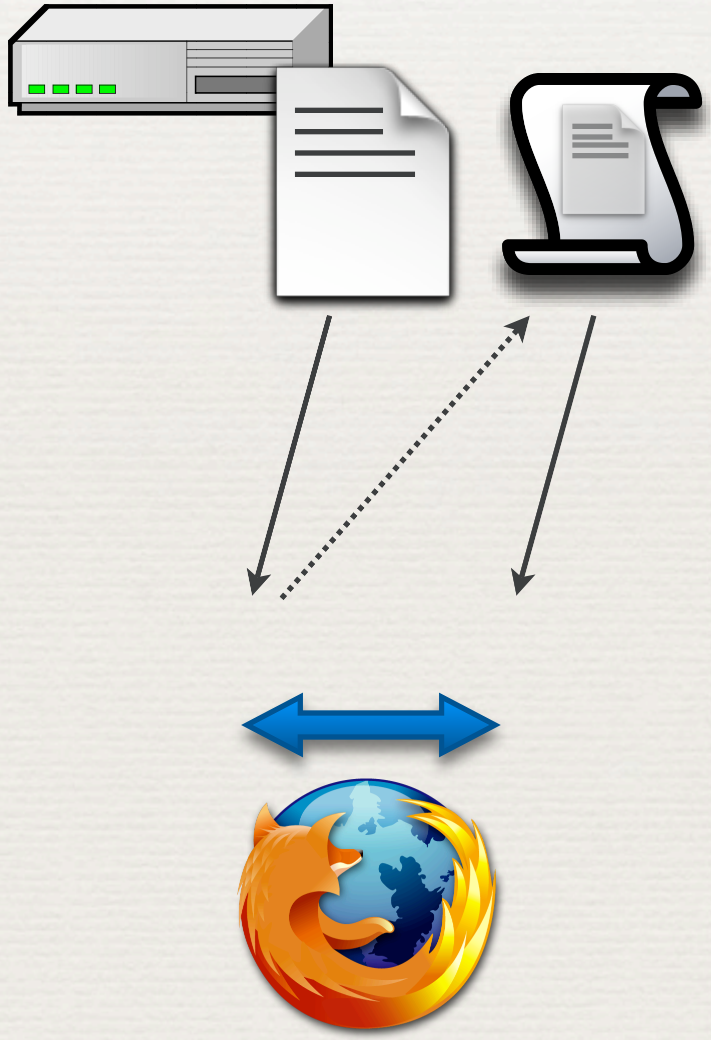
Detecting Page Changes

- Can detect with JavaScript
- Built a **Web Tripwire:**
  - Runs in client's browser
  - Finds most changes to HTML
  - Reports to user & server



http://vancouver.cs.washington.edu

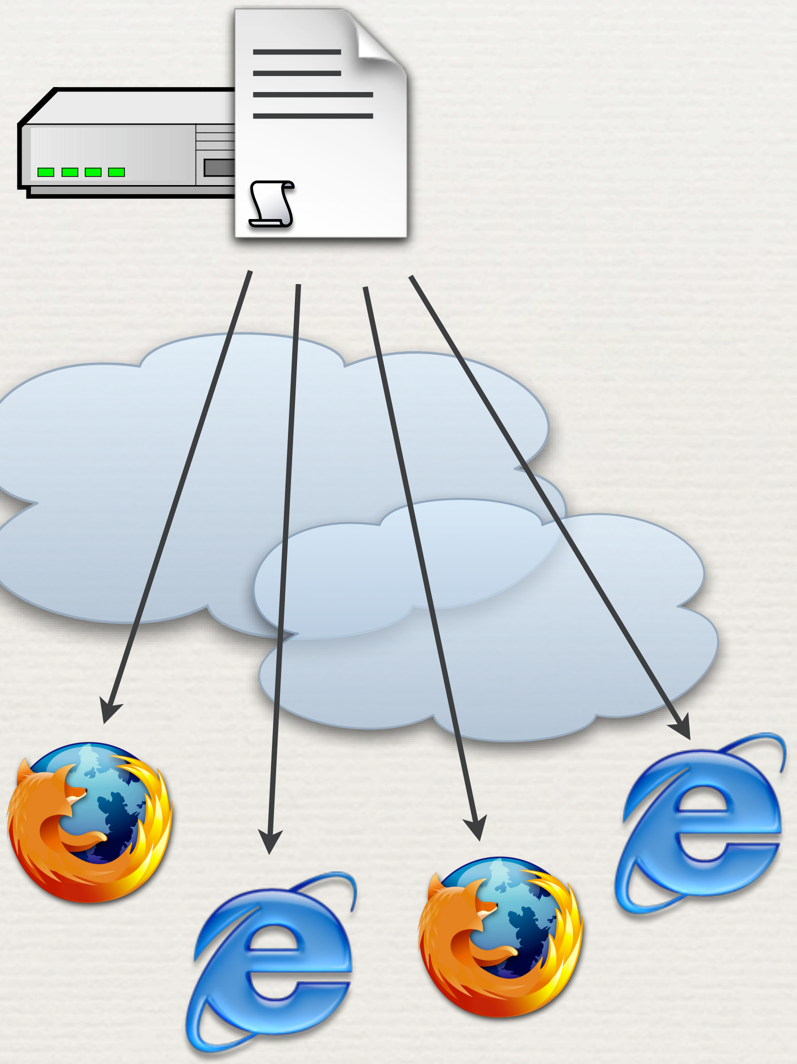
How it Works



- Fetch and render original page
- Fetch JavaScript code in background
  - Second, encoded copy of page
- Can't compare against DOM directly
  - Use XmlHttpRequest to fetch page's source code as a string

Attracting Visitors

- Wanted view of many clients on many networks
- Posted to **Slashdot**, **Digg**, etc.
  - Visits from over 50,000 unique IP addresses



Outline

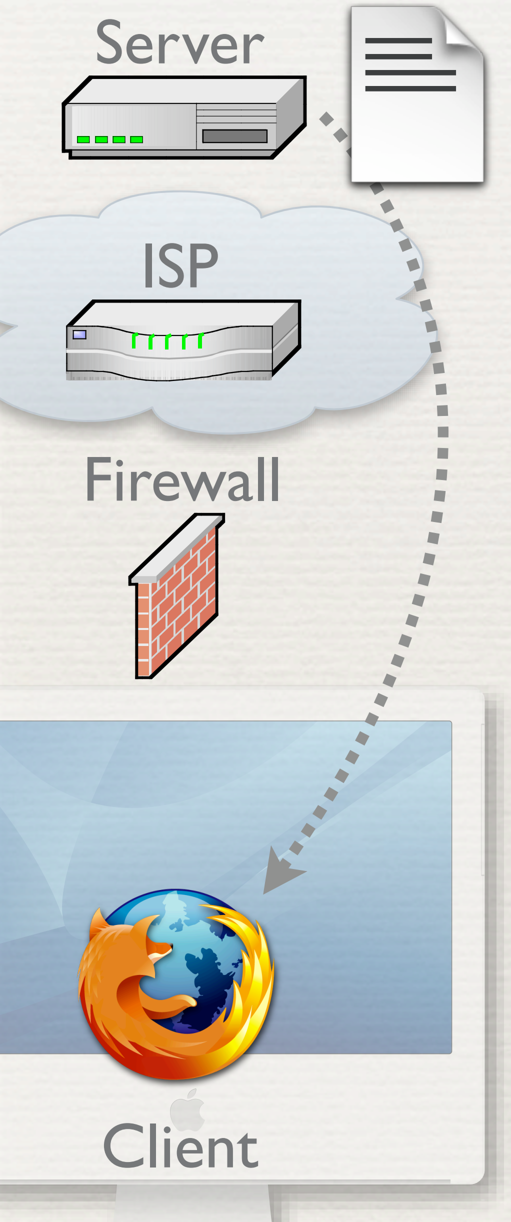
Detecting In-Flight Changes

Measurement Results

Dangerous Consequences

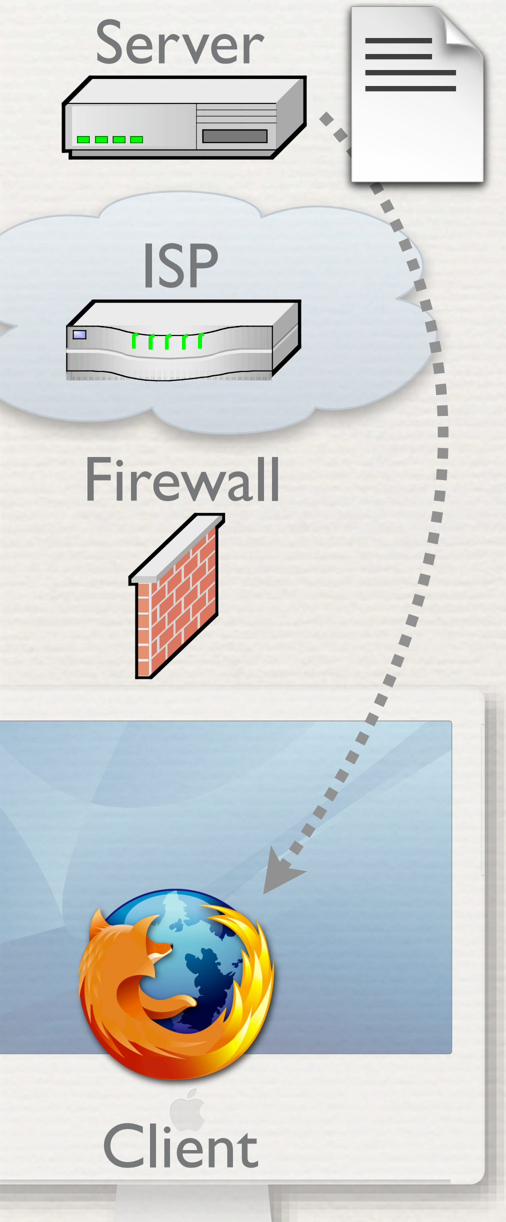
Web Tripwires for Publishers

Many Users Affected



- 657 clients saw changes (1.3%)
  - Many made by client software
  - Some made by agents in network
- Diverse incentives
- Often concerning for publishers

Many Types of Changes



Internet Service Providers

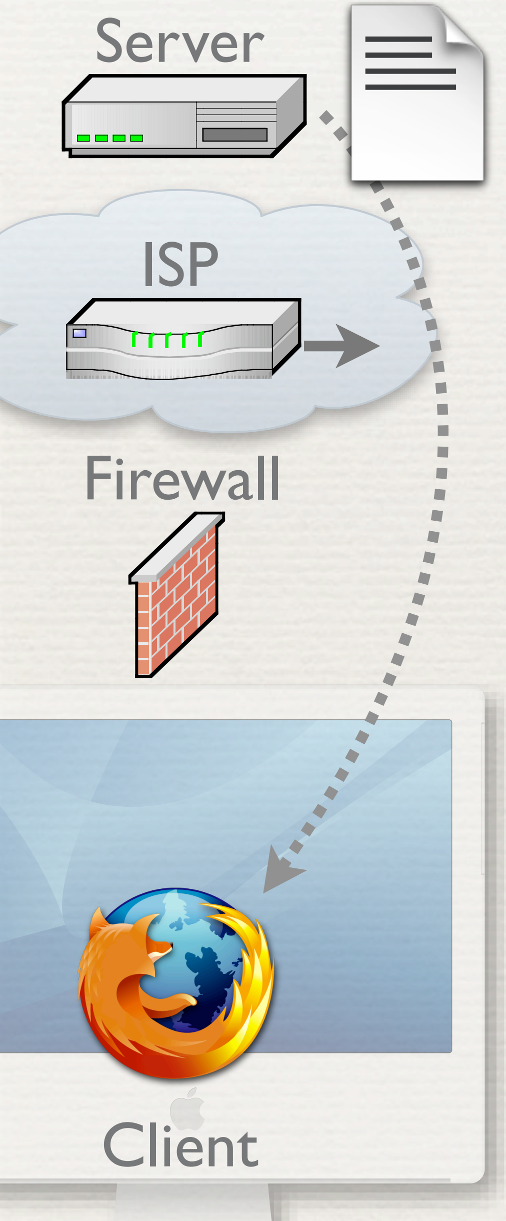
Enterprise Firewalls

Client Proxies

Malware

http://vancouver.cs.washington.edu

Changes by ISPs



- **Injected Advertisements** (2.4%)
  - NebuAd, MetroFi, LokBox, ...

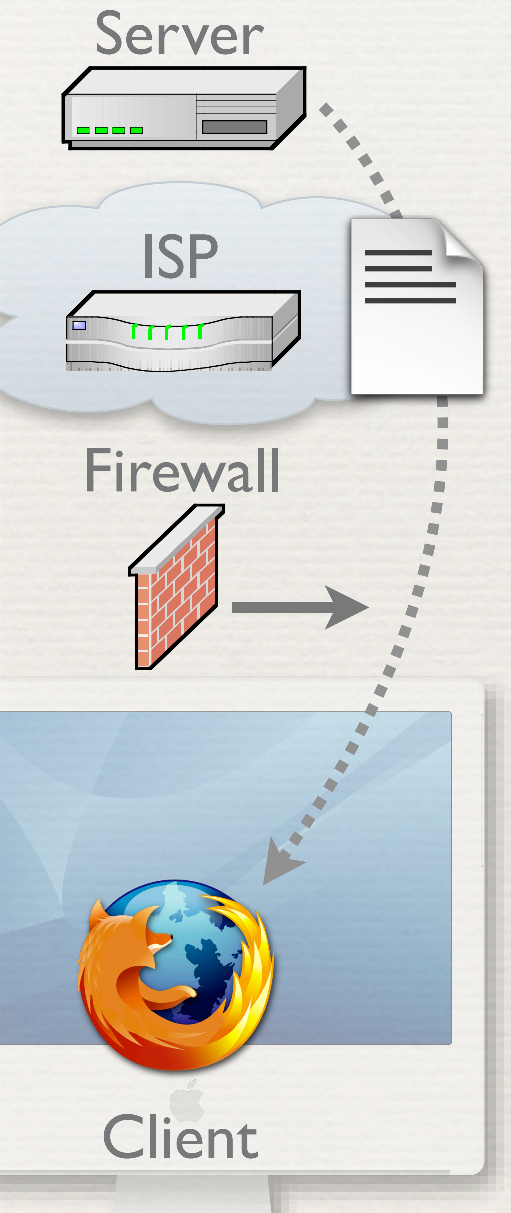
Revenue for ISP; annoy users

Growing Trend?
PerfTech, Front Porch, Adzilla, Phorm

- **Compression** (4.6%)

http://vancouver.cs.washington.edu

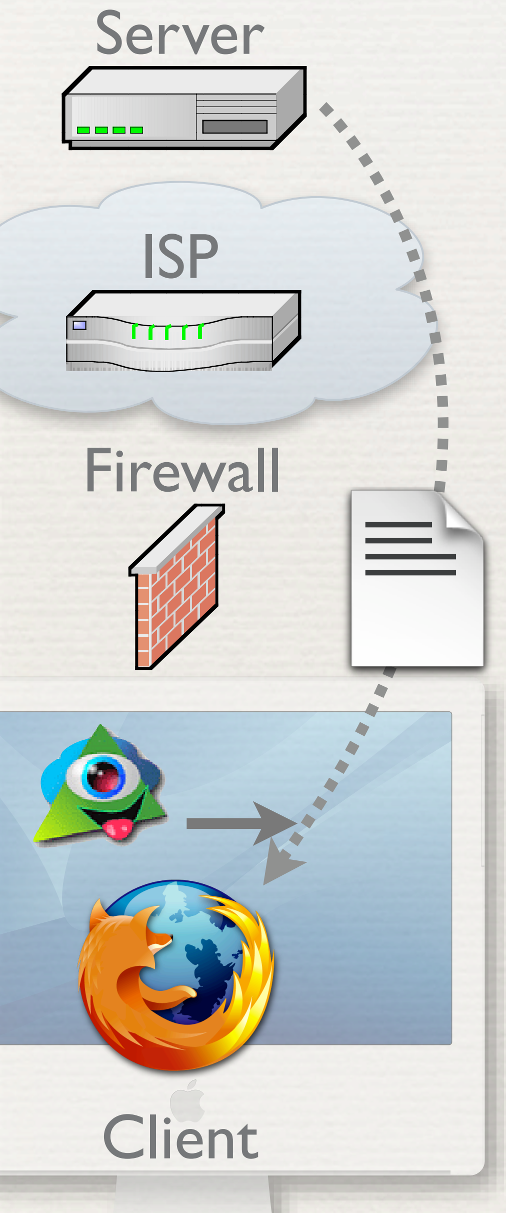
Changes by Enterprises



- **Security Checking Scripts** (2.3%)
  - BlueCoat Web Filter

Safer for clients; reduce risk

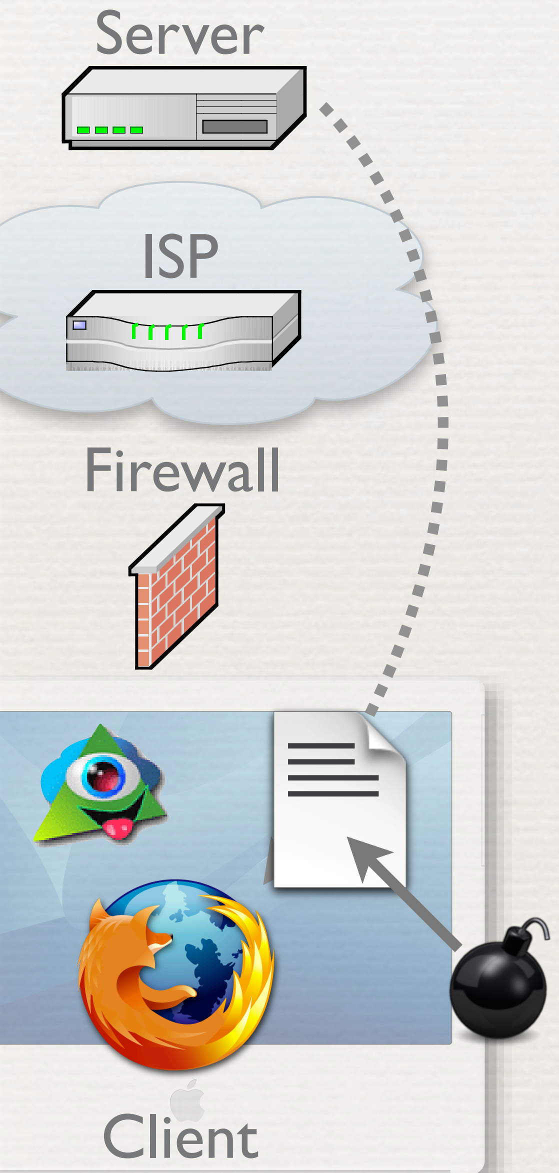
Changes by Client Proxies

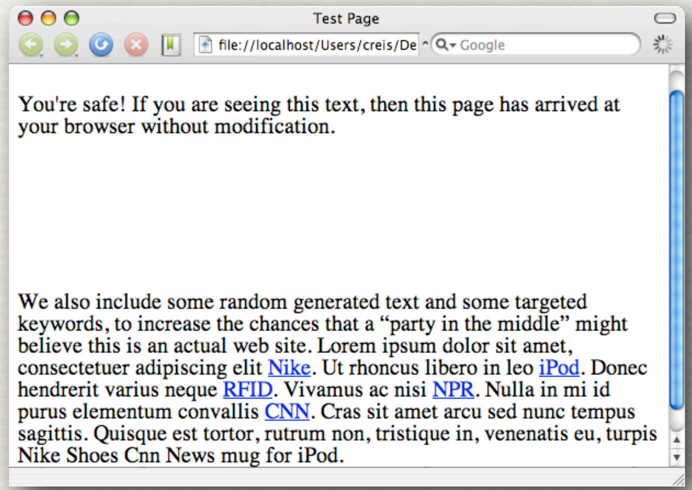


- **Popup & Ad Blockers** (71%)
  - Zone Alarm, Ad Muncher, ...

Less annoying; impact revenue

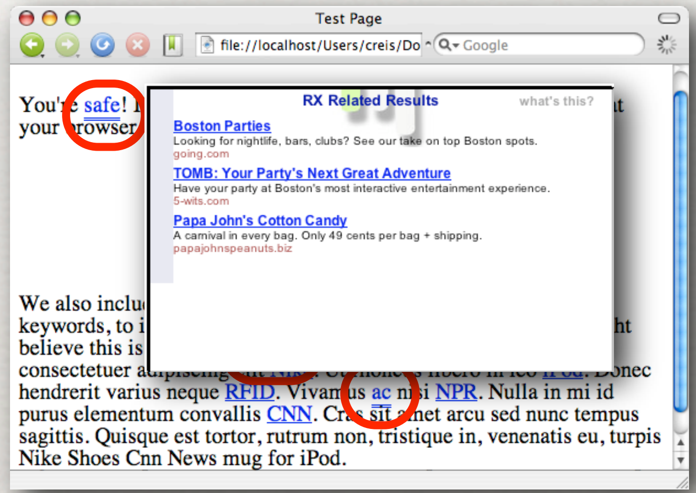
Changes by Malware





You're safe! If you are seeing this text, then this page has arrived at your browser without modification.

We also include some random generated text and some targeted keywords, to increase the chances that a "party in the middle" might believe this is an actual web site. Lorem ipsum dolor sit amet, consectetuer adipiscing elit Nike. Ut rhoncus libero in leo iPod. Donec hendrerit varius neque RFID. Vivamus ac nisi NPR. Nulla in mi id purus elementum convallis CNN. Cras sit amet arcu sed nunc tempus sagittis. Quisque est tortor, rutrum non, tristique in, venenatis eu, turpis Nike Shoes Cnn News mug for iPod.



Boston Parties
Looking for nightlife, bars, clubs? See our take on top Boston spots.
going.com

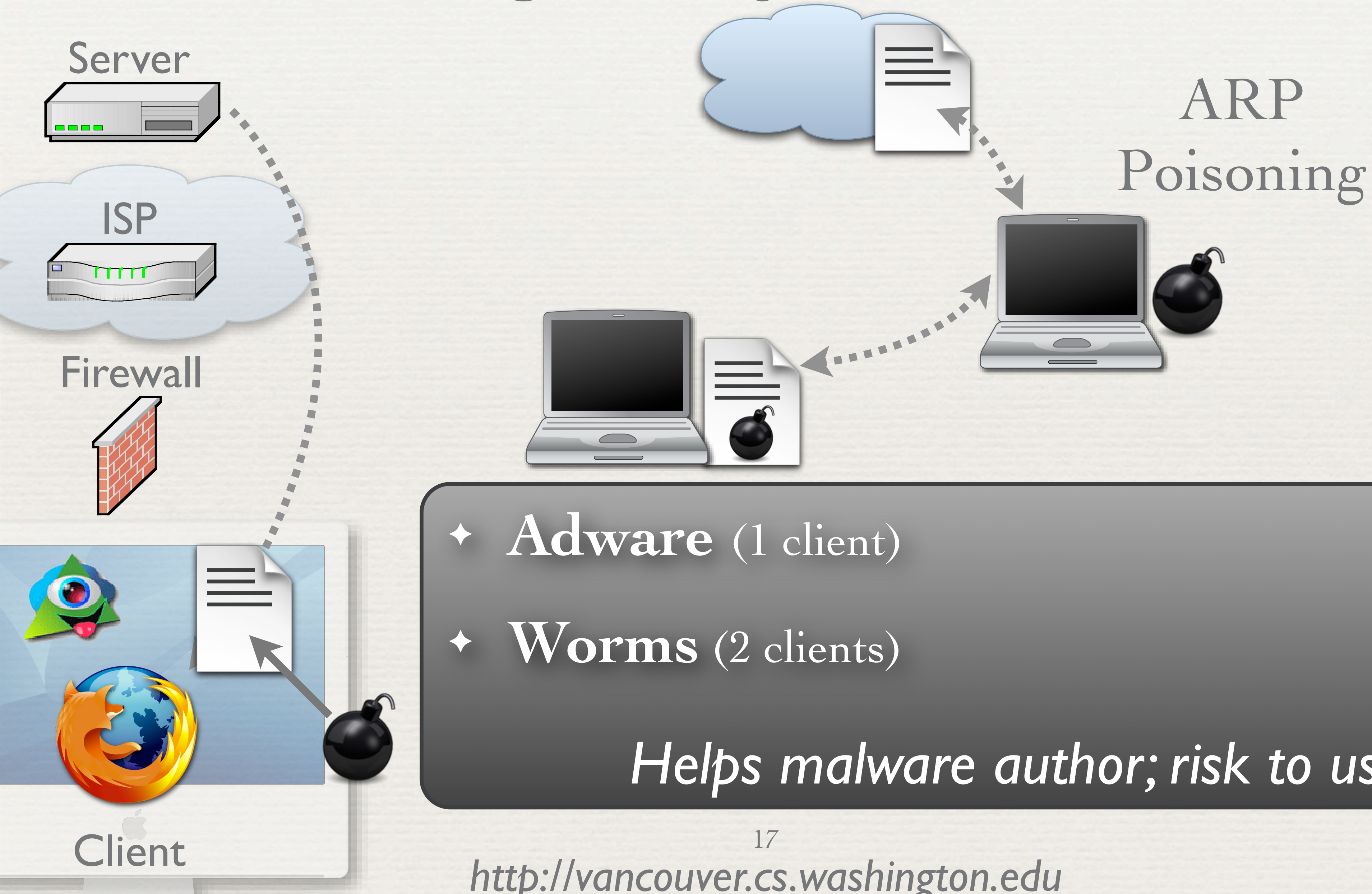
TOMB: Your Party's Next Great Adventure
Have your party at Boston's most interactive entertainment experience.
5-wits.com

Papa John's Cotton Candy
A carnival in every bag. Only 49 cents per bag + shipping.
papajohnspeanuts.biz

- **Adware** (1 client)

http://vancouver.cs.washington.edu

Changes by Malware



- **Adware** (1 client)
- **Worms** (2 clients)

Helps malware author; risk to user

http://vancouver.cs.washington.edu

Outline

Detecting In-Flight Changes

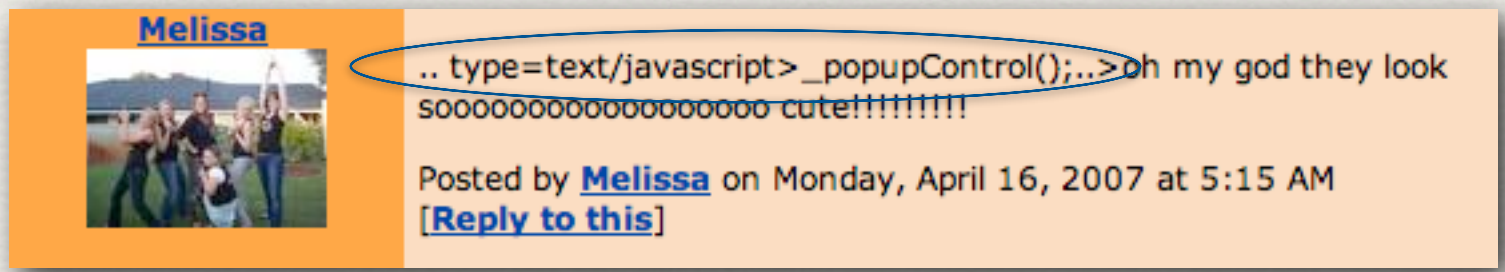
Measurement Results

Dangerous Consequences

Web Tripwires for Publishers

Unanticipated Impact

- Some changes **inadvertently** broke pages
  - JavaScript errors
  - Interfered with MySpace / forum posts



Melissa

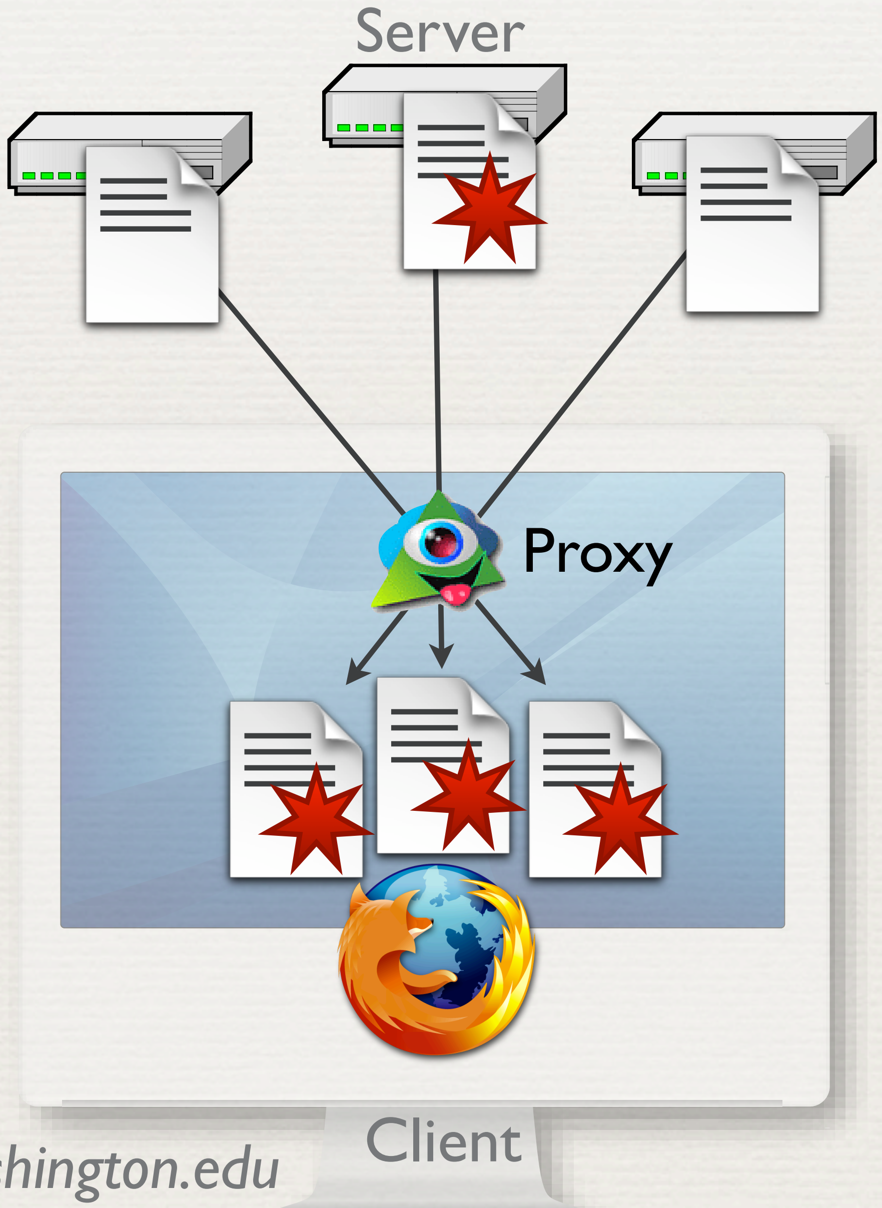
.. type=text/javascript>_popupControl();..>oh my god they look soooooooooooooooooo cute!!!!!!!!!

Posted by Melissa on Monday, April 16, 2007 at 5:15 AM
[Reply to this]

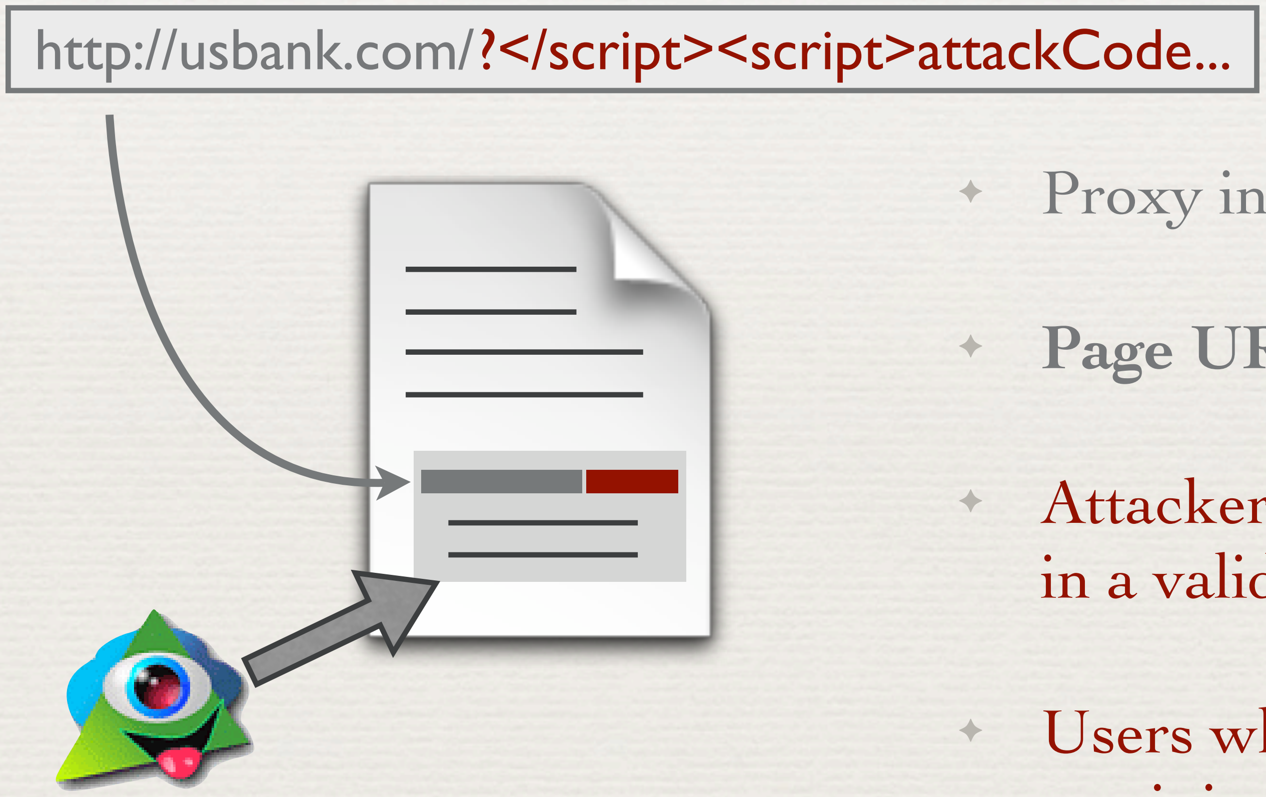
http://vancouver.cs.washington.edu

Introduced Vulnerabilities

- **XSS** allows script injection
  - Usually fixed at server
- Some proxies made otherwise safe pages vulnerable
  - Ad Muncher, Proxomitron
- Affected most HTTP pages
  - Like a **root exploit**

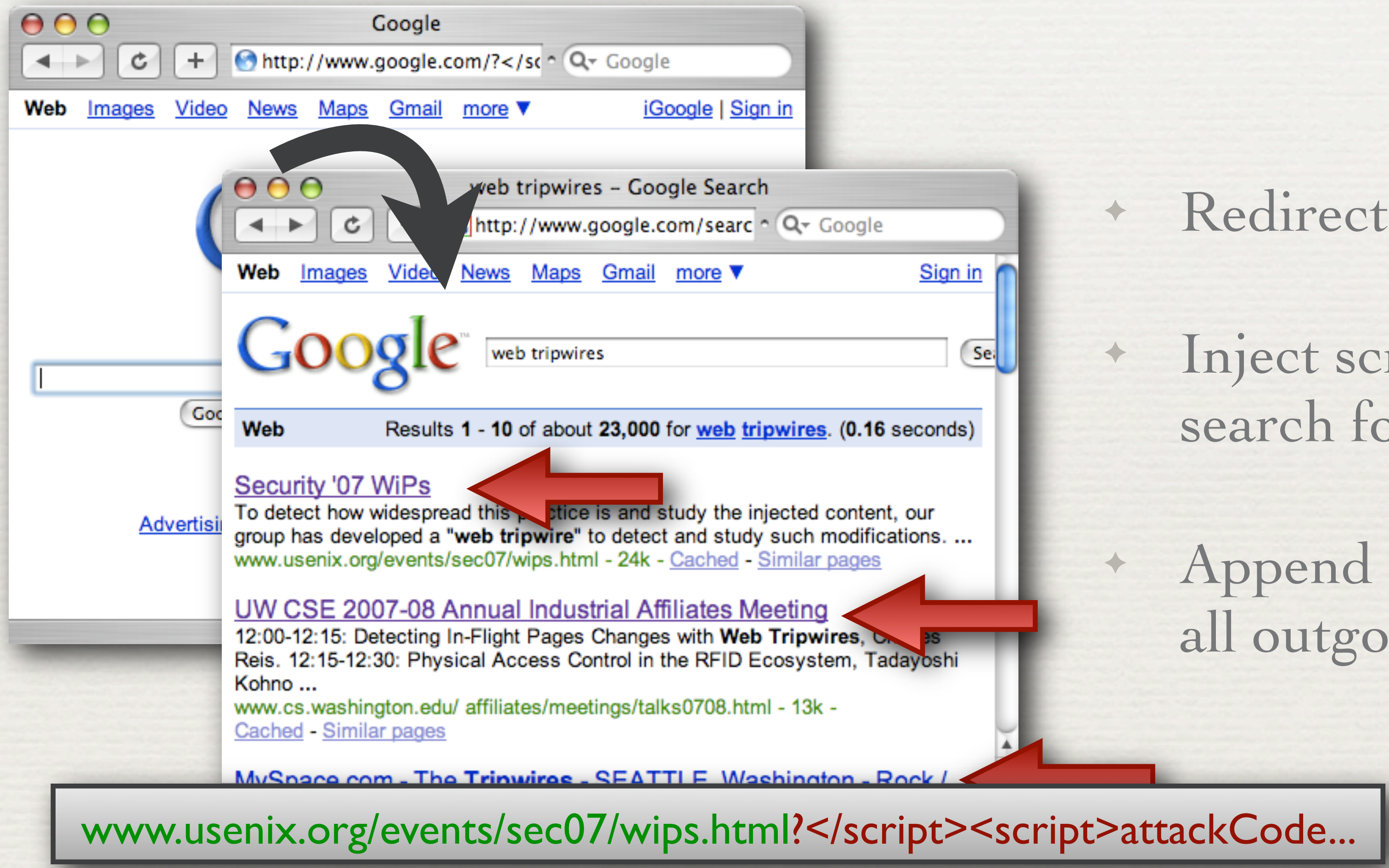


XSS via Proxy



- Proxy injected script code
- **Page URL** was included in code
- Attacker could place script code in a valid URL
- Users who follow the URL run injected code

Example Exploit



Google

http://www.google.com/?</sc

Web Images Video News Maps Gmail more ▼ iGoogle | Sign in

web tripwires – Google Search

http://www.google.com/searc

Web Images Video News Maps Gmail more ▼ Sign in

web tripwires

Web Results 1 - 10 of about 23,000 for web tripwires. (0.16 seconds)

Security '07 WiPs
To detect how widespread this practice is and study the injected content, our group has developed a "web tripwire" to detect and study such modifications. ...
www.usenix.org/events/sec07/wips.html - 24k - Cached - Similar pages

UW CSE 2007-08 Annual Industrial Affiliates Meeting
12:00-12:15: Detecting In-Flight Pages Changes with Web Tripwires, Charles Reis. 12:15-12:30: Physical Access Control in the RFID Ecosystem, Tadayoshi Kohno ...
www.cs.washington.edu/ affiliates/meetings/talks0708.html - 13k -
Cached - Similar pages

MySpace.com - The Tripwires - SEATTLE, Washington - Rock /

www.usenix.org/events/sec07/wips.html?</script><script>attackCode...

- Redirect user to Google
- Inject script code into search form
- Append exploit code to all outgoing links

Vulnerability Aftermath

- Reported vulnerabilities; now fixed
- Web tripwires can help find vulnerabilities
  - Search for URL in page changes

Outline

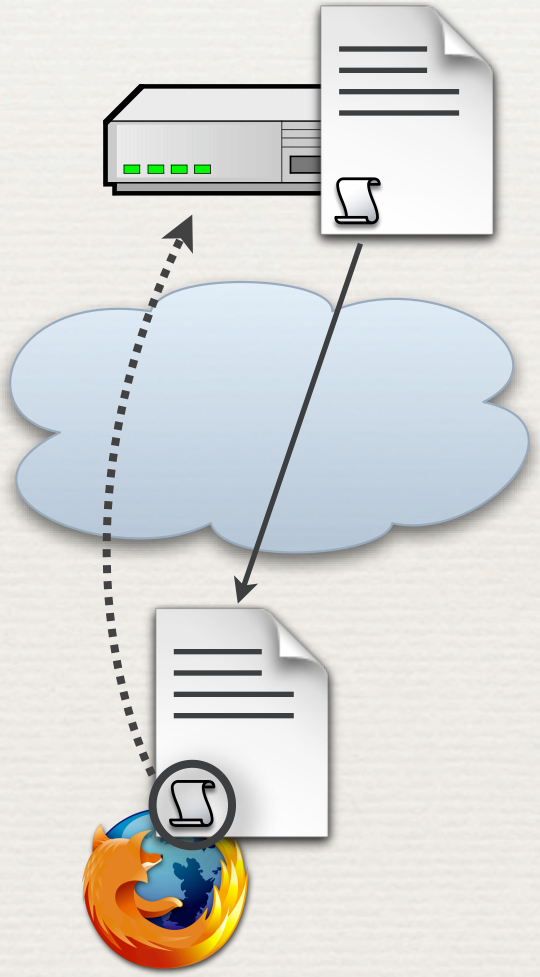
- Detecting In-Flight Changes
- Measurement Results
- Dangerous Consequences
- Web Tripwires for Publishers

How to React?

- Option 1: **Use HTTPS**
  - Encryption prevents in-flight changes
- But... costly and rigid
  - Can't allow security checks, caching, etc.

Web Tripwires

- JavaScript code to detect changes
- Easy for publishers to deploy
  - **Configurable toolkit**
  - **Web tripwire service**
- But... not cryptographically secure
- Can be robust in practice

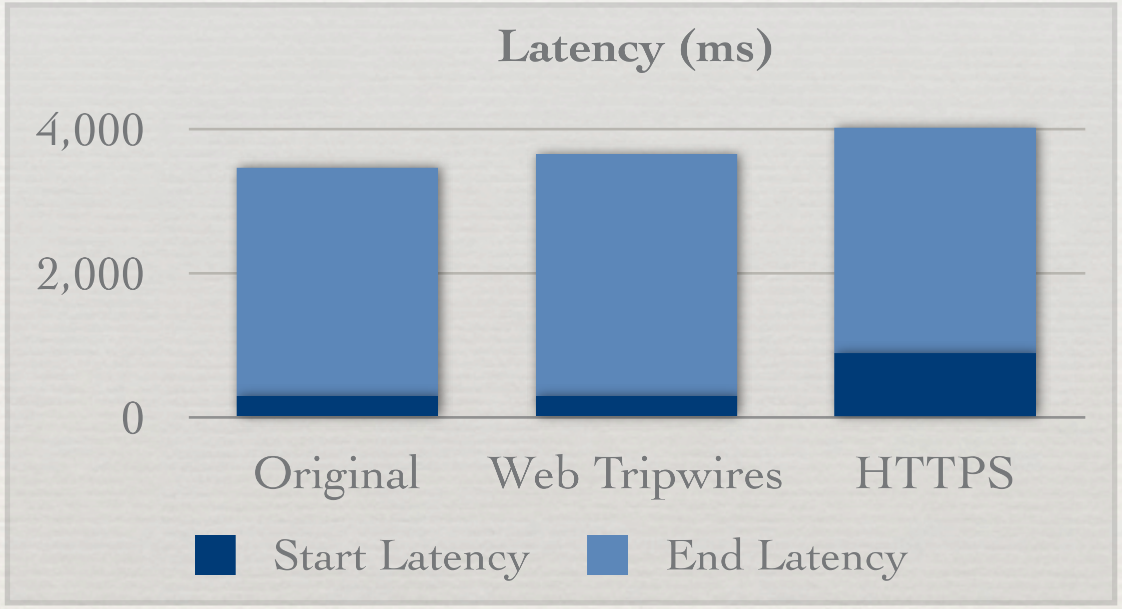


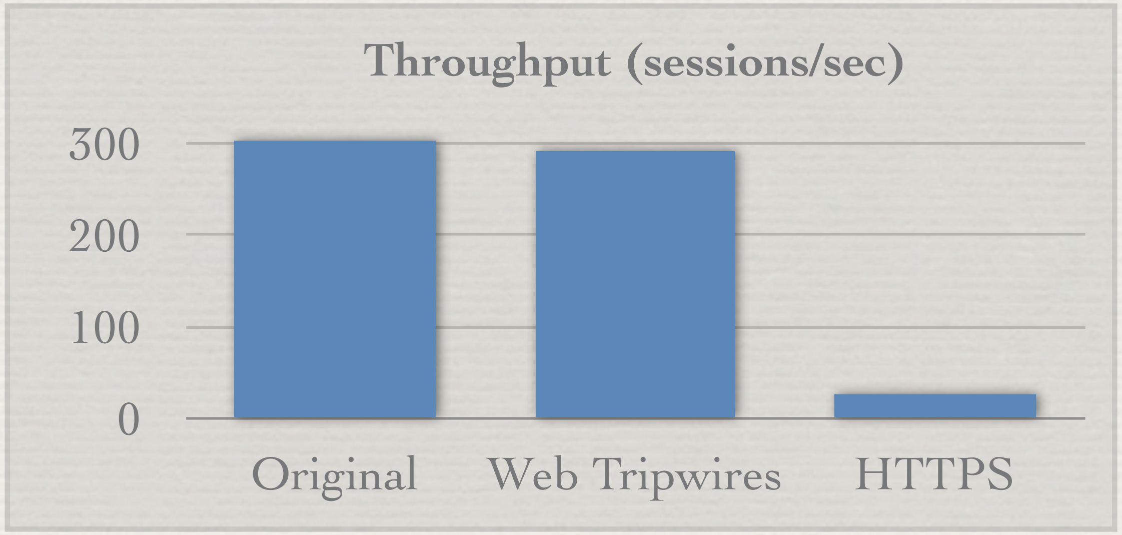
Tradeoffs

HTTPS	Web Tripwires
✦ Prevents in-flight changes, as well as some useful services	✦ Detects most in-flight changes
✦ Cryptographically robust	✦ Could face an arms race ✦ Obfuscation can challenge adversaries
✦ Expensive: certificates, computation, extra RTTs	✦ Inexpensive to deploy

Performance Impact

- Relative to HTTPS, web tripwires have:
    - Low latency
    - High throughput





http://vancouver.cs.washington.edu

Web Tripwire Summary

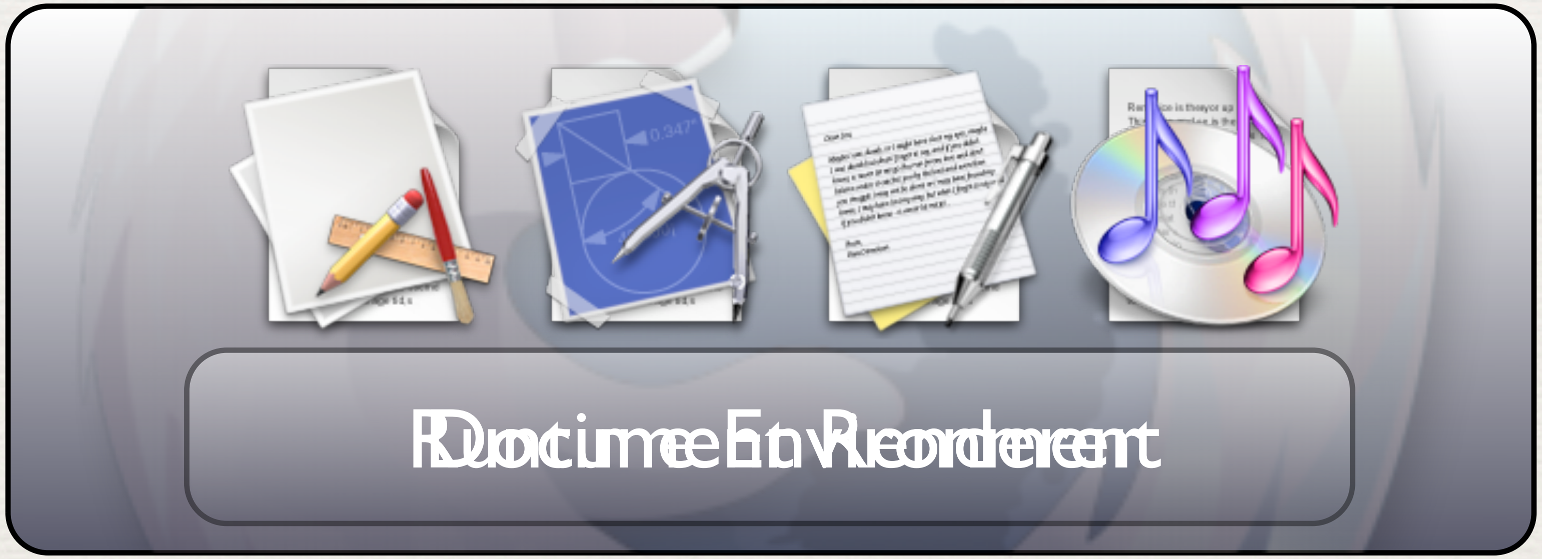
- HTTP web pages are being **changed in flight**
  - Real negative impact for publishers & users
  - Page rewriters have dangerous power
- **Web tripwires** can help publishers react

2. Safe Browser Architectures

Joint work with
Steve Gribble, Hank Levy

How did we get here?

- Web content has evolved



- Browser now analogous to OS
- Current architectures inadequate

Safety Threats

- Many more than in-flight page changes
    - Exploits, XSS, CSRF, interference
- Need better support for **web programs**
    - Must improve both **program definitions** and **browser architectures**

Outline

Defining **program boundaries**

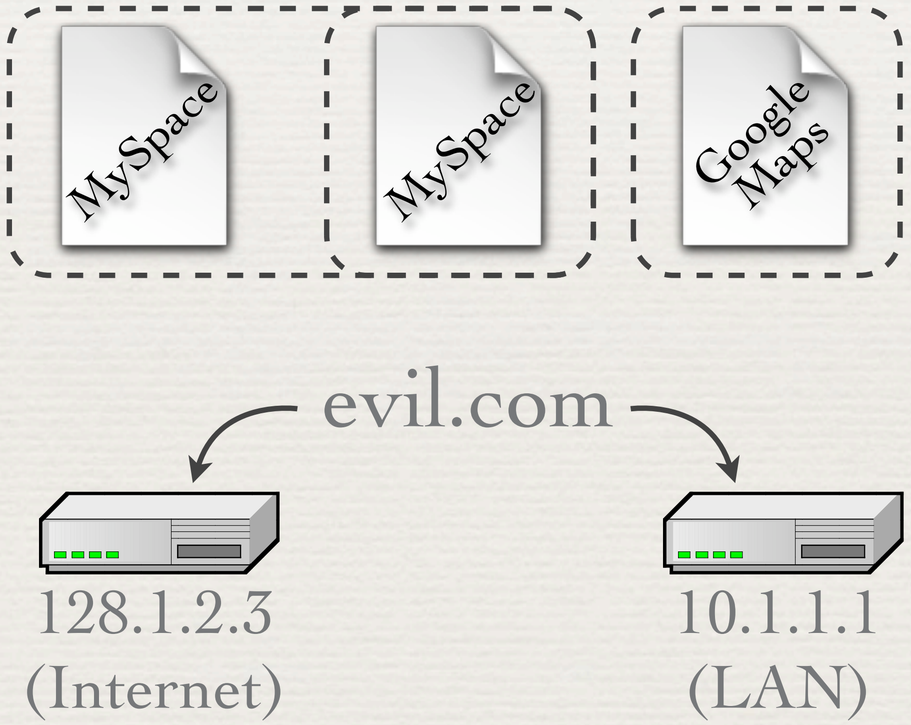
Preventing **unwanted code**

Isolating programs in browser

Applying **uniform policies**

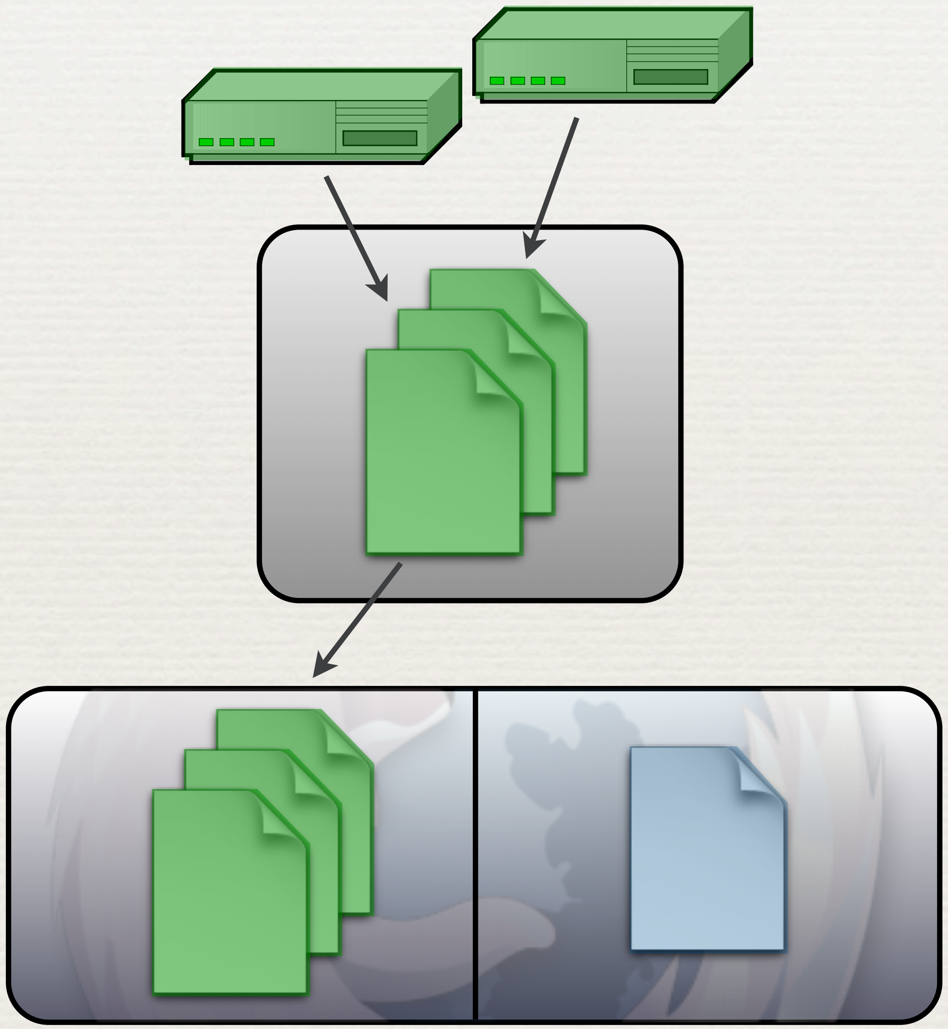
Reis, Gribble, Levy [HotNets '07]

Can't identify program boundaries



- **Same Origin Policy** provides current boundaries
- Flawed approach:
    - Too narrow
    - Too broad
    - Easily compromised

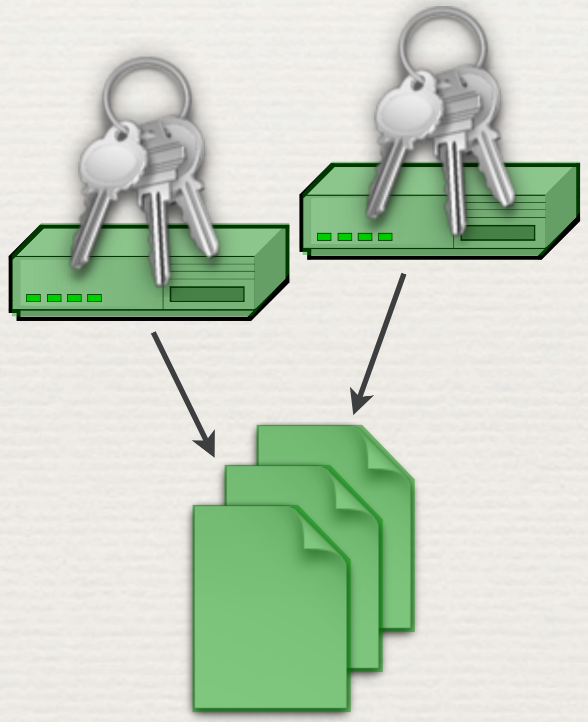
Program Boundaries



- New abstractions:
  - **Web program**
  - **Program instance**
- Must explicitly assign resources to programs

Keys as Boundaries

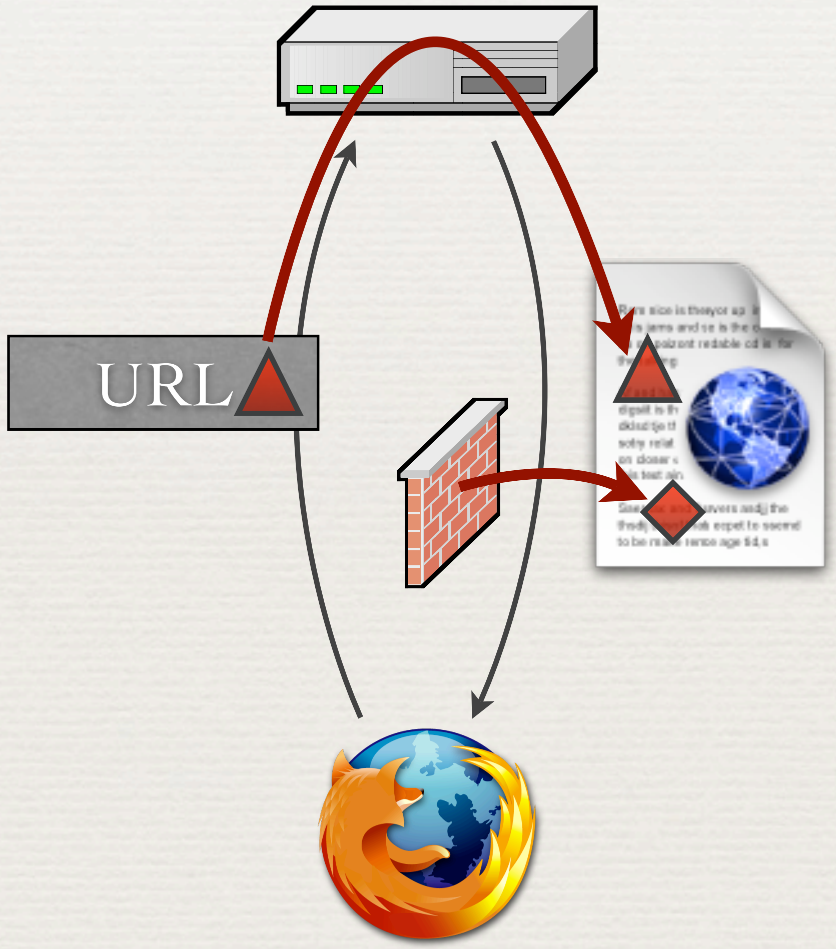
- Author holds a private key
- Web program:
  - Public key
  - Set of signed documents
- No PKI required



Outline

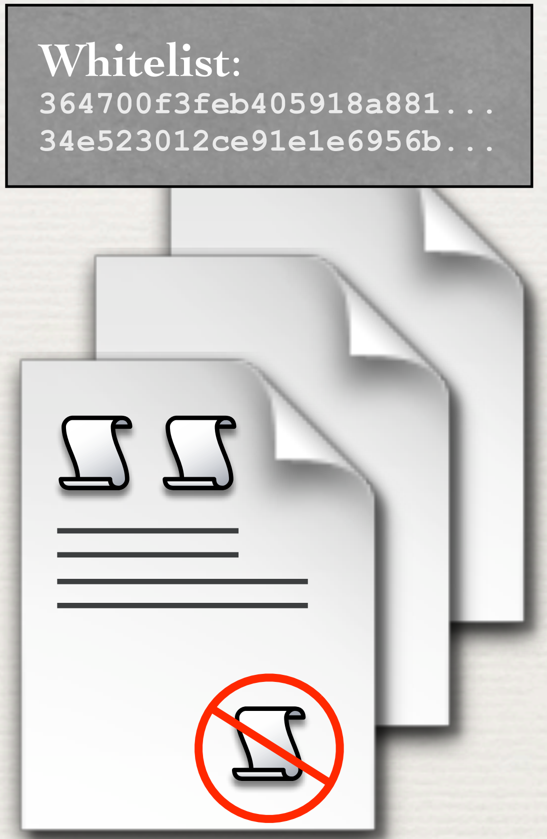
- Defining **program boundaries**
- Preventing **unwanted code**
- **Isolating programs** in browser
- Applying **uniform policies**

Can't prevent unwanted code



- Scripts injected via user input (XSS)
- Scripts injected in-flight

Authorized Code



- Need to authorize all web program code
- **Script Whitelists** are a start
  Jim, Swamy, Hicks [WWW '07];
  Reis, Gribble, Bershad, Levy
  - Browser ignores any script whose hash is not in list
  - Should apply to all active code; could sign whitelist
- Challenges for dynamic pages

Outline

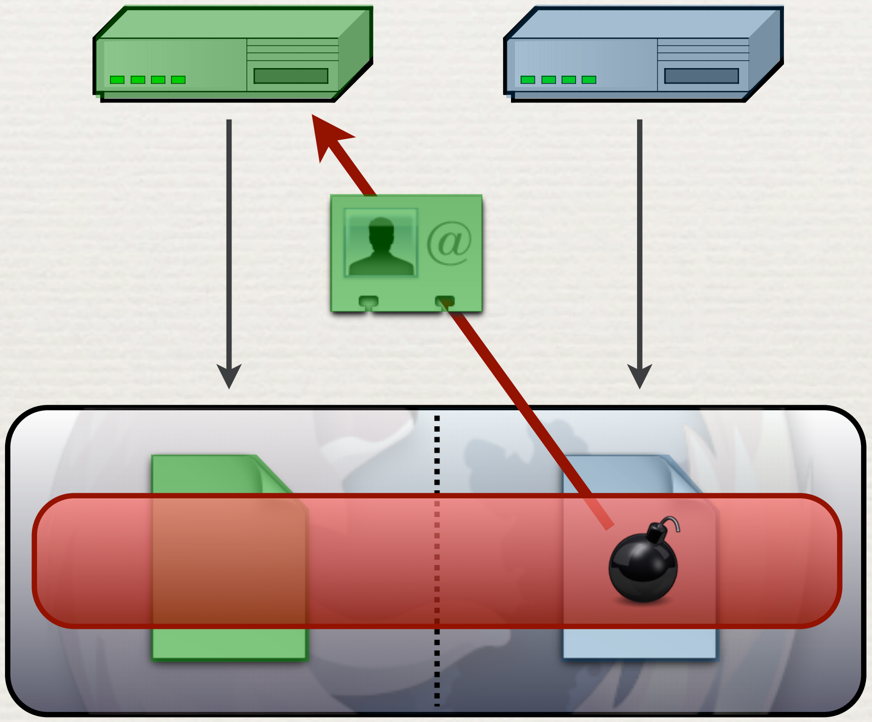
Defining **program boundaries**

Preventing **unwanted code**

Isolating programs in browser

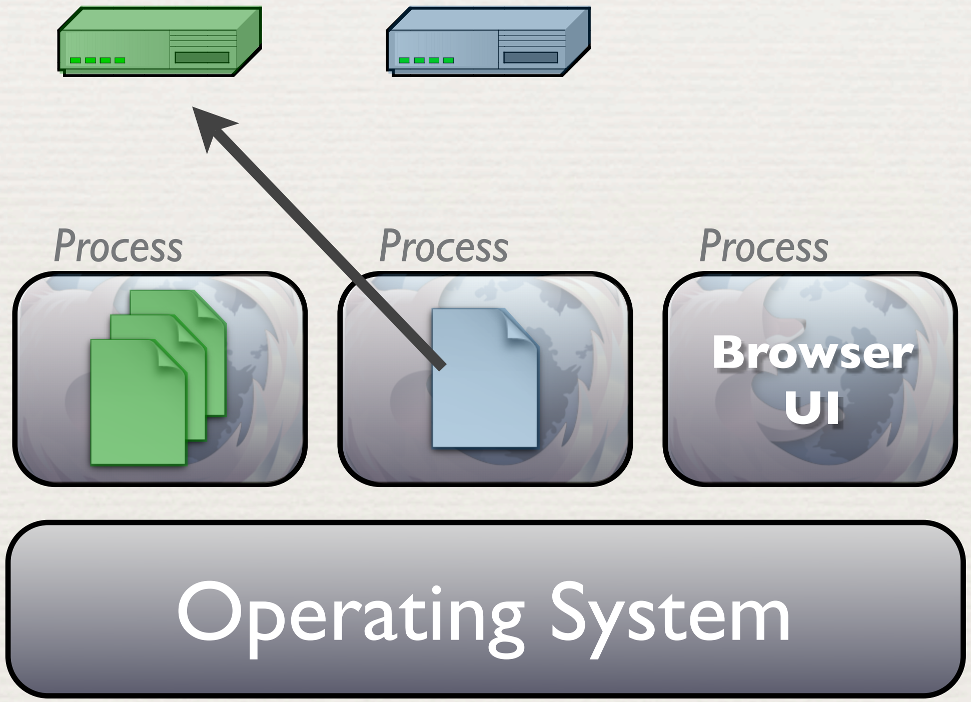
Applying **uniform policies**

Can't isolate programs in browser



- Can abuse credentials of other sites (CSRF)
- Failures, resource contention

Program Isolation



- **Privacy:**
  - Isolate credentials between instances
- **Robustness:**
  - OS process for each program instance

Reis et al. [UW Tech Report '07]

Outline

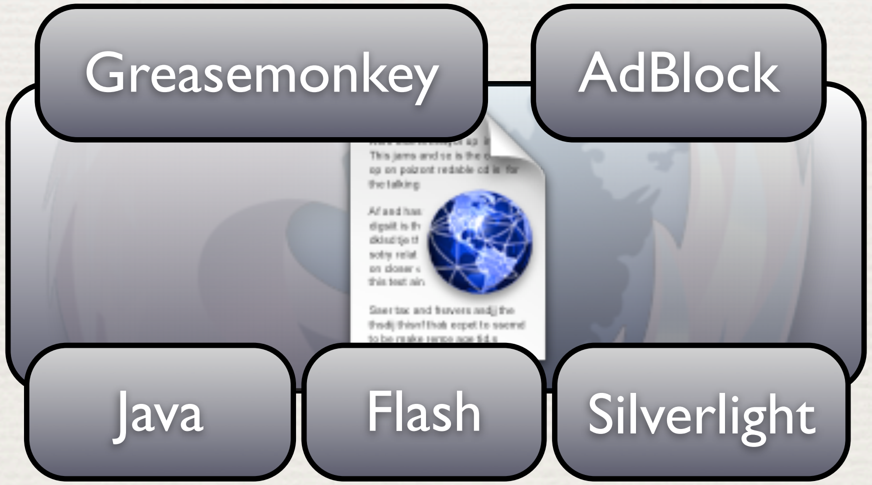
Defining **program boundaries**

Preventing **unwanted code**

Isolating programs in browser

Applying **uniform policies**

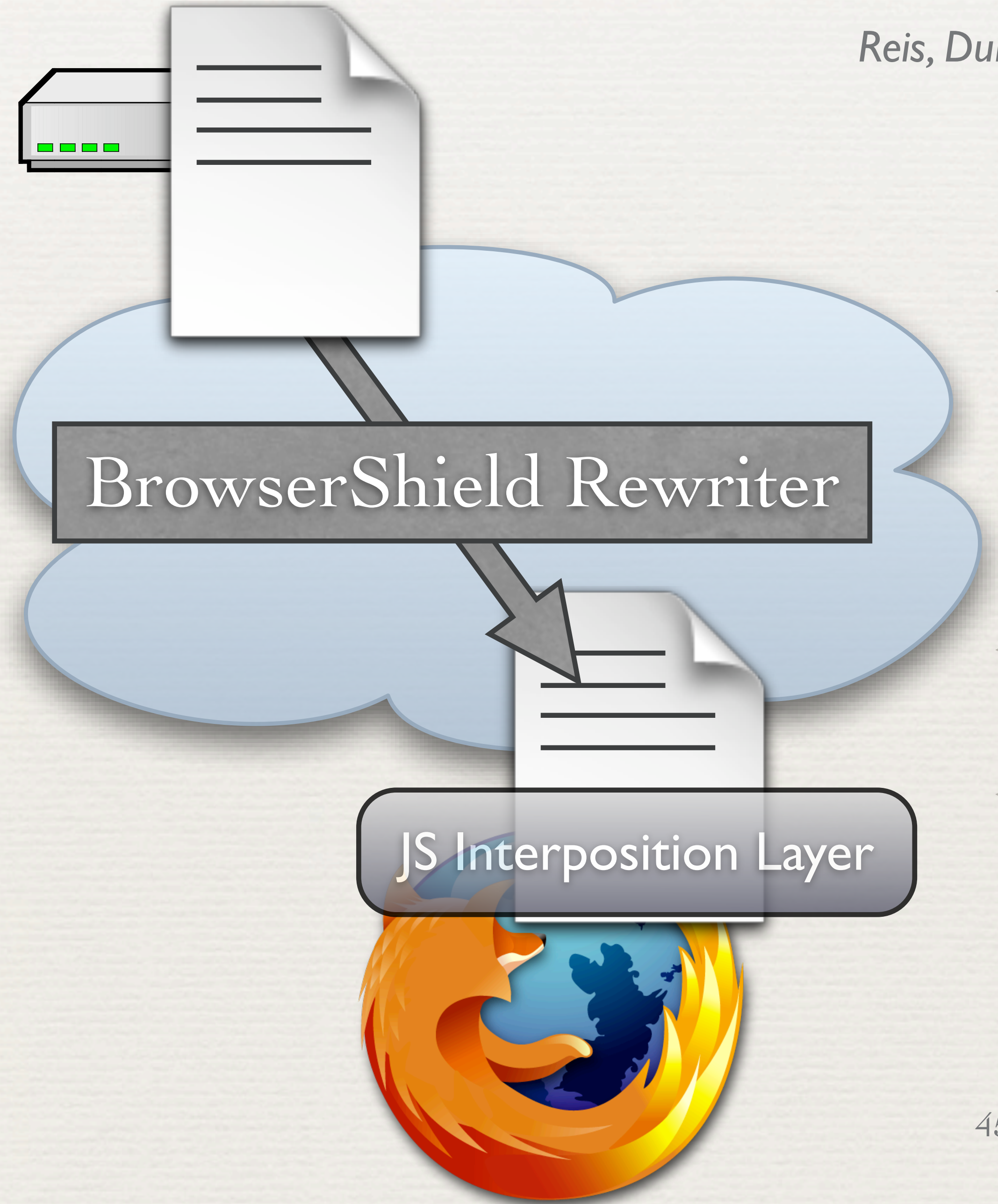
Can't apply uniform policies



- Each content type has its own security model
- No restrictions on browser extensions
- Can't reason about a web program's abilities

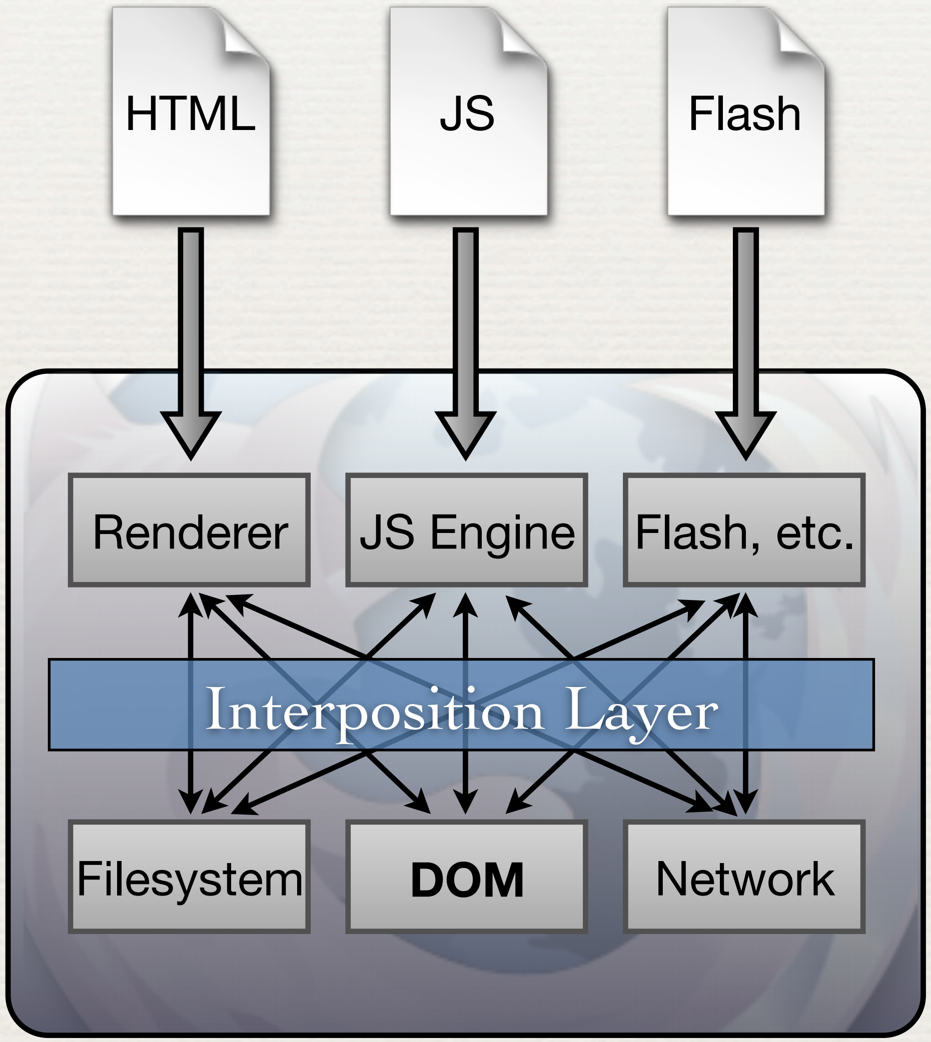
BrowserShield

Reis, Dunagan, Wang, Dubrovsky, Esmeir [OSDI '06]



- Interpose on JavaScript code
    - Prevent exploits of known vulnerabilities
- Rewrites JavaScript in-flight
- **Challenges**: HTTPS, other active content, browser quirks

Apply Uniform Policies



- Need to interpose on web content **within the browser**
  - Enforce same policies on all content types
  - Protect key resources (DOM, FS, network)

Conclusion

- Many threats in today's web
  - **In-flight page changes** pose risks
- **Web Tripwires** can help detect changes
- **Safer browser architectures** are needed
  - Program boundaries, authorized code, isolation, uniform policies