Web Browser (In)-Security "Past, Present, and Future"

SecTheory

Internet Security

About Me

- Robert "RSnake" Hansen - CEO
    - SecTheory LLC
        - Bespoke Boutique Internet Security
        - Web Application/Browser Security
        - Network/OS Security
        - Advisory capacity to VCs/start-ups
        - "We solve tough problems."
        - http://www.sectheory.com/
- Founded the web application security lab
    - http://ha.ckers.org/ - the lab
    - http://sla.ckers.org/ - the forum

Sound Familiar?

“Frames have horrible usability since they break most of the navigation features in a web browser (bookmarks, backtrack, and going to a URL all stop working).” Jakob Nielsen (1996)

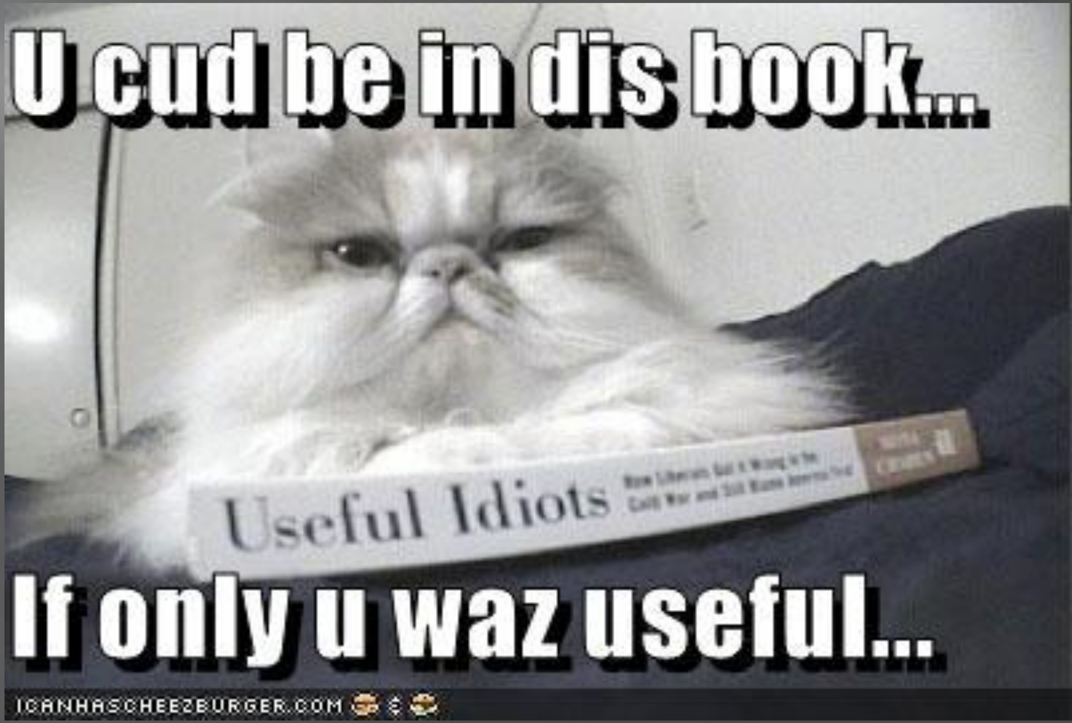
Remind you of XMLHTTPRequest?

Browser Security Then

- Text based, www and gopher
- Graphical Browsers:
  - Mosaic (1993 Nov)
  - Netscape(Oct 1994)
  - Internet Explorer (Aug 1995 – Spyglass license)
- 1995 comes with JavaScript support originally created by Brendan Eich at Netscape, which is soon adopted by all browsers.
- Microsoft answered with VBScript in 1996.

The Big Security Problems with HTTP in the Early 90's

- No session/state management
- Not encrypted
- Only supports basic auth which wasn't good for state management:
  - Two users could use the same userid
  - UI was terrible

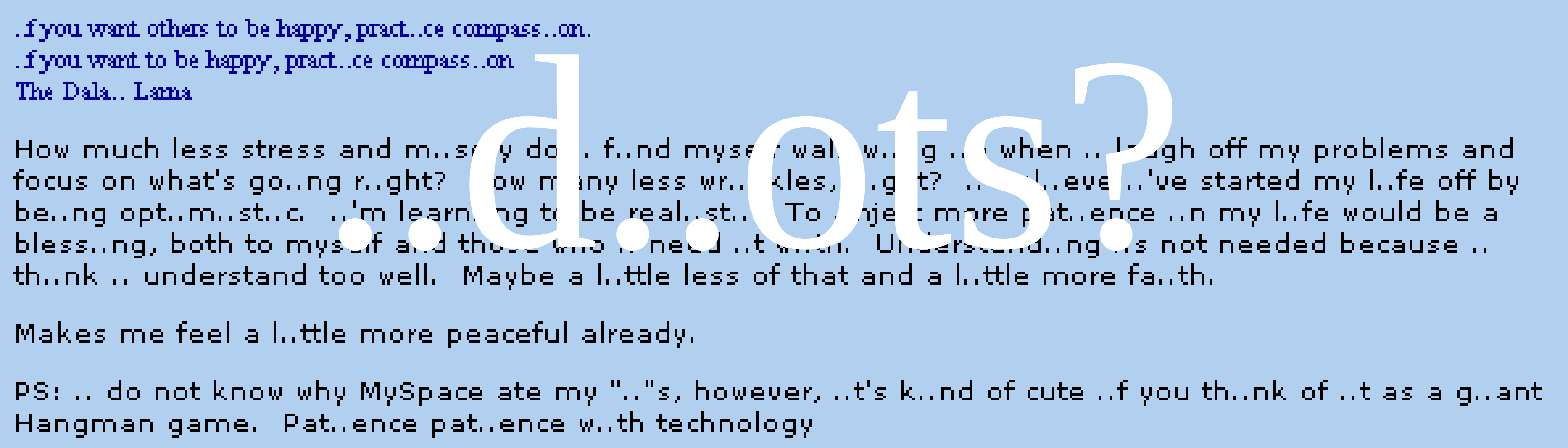


HTTP Security Add-ons

- Netscape implemented SSL (1994)
- Web developers were told to create their own credentials/state management and use secure cookies (rfc2109 in 1997)
- HTTP/1.1 brought digest auth which also sort of helped with state management, kind of (rfc2616 in 1999)

The Big Problems with Browsers 90's & Early 2000's

- Allowed to contact anything
- No easy upgrades
- Cross domain policy issues (MS XSS in 2000)
- Lots of exploits
- HTML TIMTOWTDI (Browser rendering)



..f you want others to be happy, pract..ce compass..on.
..f you want to be happy, pract..ce compass..on
The Dala.. Lama

How much less stress and m..sery do .. f..nd myself walk..ng ... when .. laugh off my problems and
focus on what's go..ng r..ght? How many less wr..nkles, .. g..t? .. bel..eve ..'ve started my l..fe off by
be..ng opt..m..st..c. ..'m learn..ng to be real..st... To ..nject more pat..ence ..n my l..fe would be a
bless..ng, both to myself and those who .. need ..t w..th. Understand..ng ..s not needed because ..
th..nk .. understand too well. Maybe a l..ttle less of that and a l..ttle more fa..th.

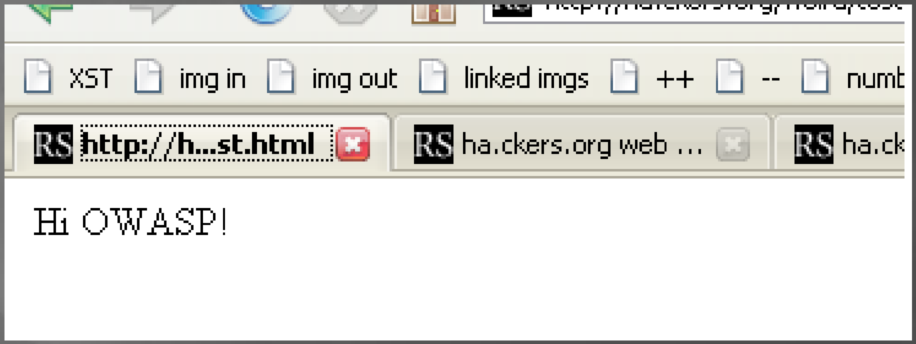
Makes me feel a l..ttle more peaceful already.

PS: .. do not know why MySpace ate my ".."s, however, ..t's k..nd of cute ..f you th..nk of ..t as a g..ant
Hangman game. Pat..ence pat..ence w..th technology

..d..ots?

HTML TIMTOWTDI

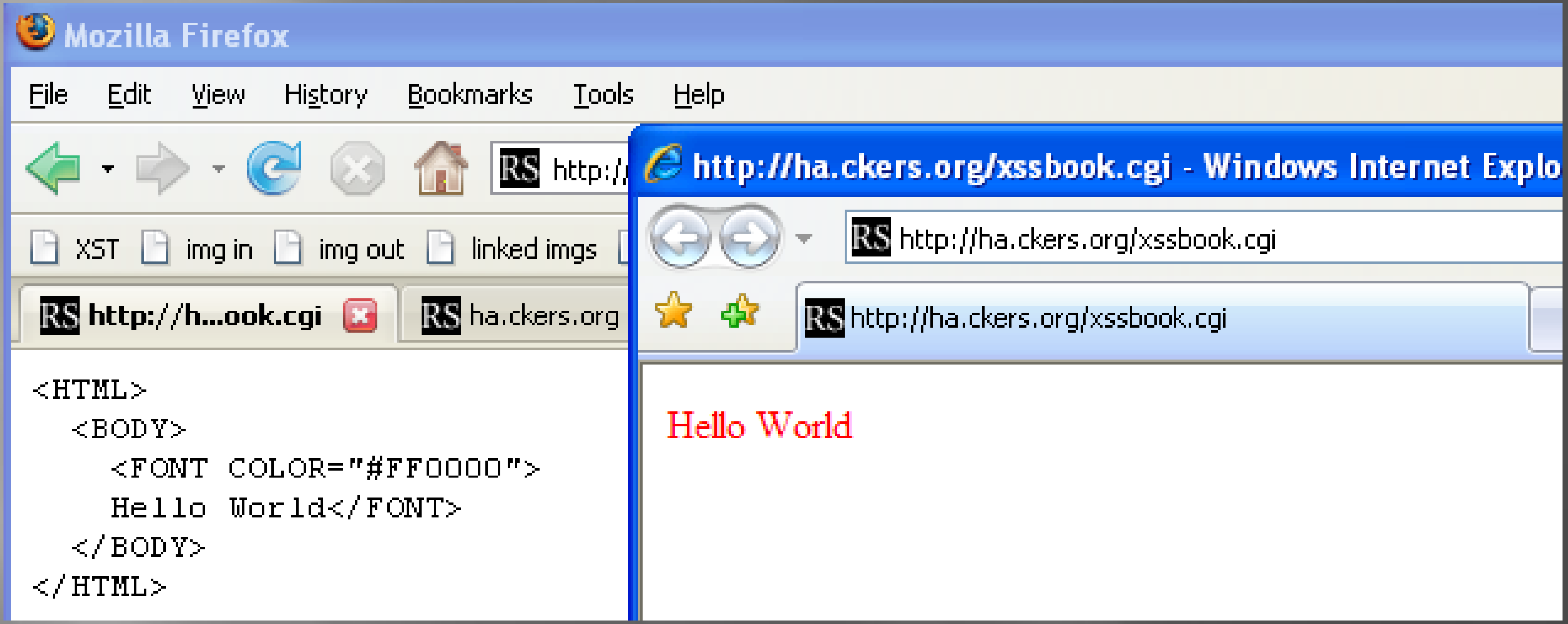
```
&#XFEFF;&#160;&#x48;<FONT
HUH=">'>`>SUP"></FONT>&#XFEFF;&#105;</table
><A STYLE="&#x63&#x6F&#x6C&#x6F&#x72&#x3A
&#x72&#x67&#x62&#x28&#x32&#x35&#x35&#x2C&#x32&#x35&#x3
5&#x2C&#x32&#x35&#x35
&#x29">I&#X00FeFF;</A><BDO
DIR="rtl">S&#XFEFF;AW<HRM>O</BDO>P!
```



Hi OWASP!

How can we find fraud when we don’t know what it looks like?

Browser Differences, Beyond UI


Mozilla Firefox
File
Edit
View
History
Bookmarks
Tools
Help
http://
XST
img in
img out
linked imgs
http://h...ook.cgi
ha.ckers.org
http://ha.ckers.org/xssbook.cgi - Windows Internet Explo
http://ha.ckers.org/xssbook.cgi
http://ha.ckers.org/xssbook.cgi

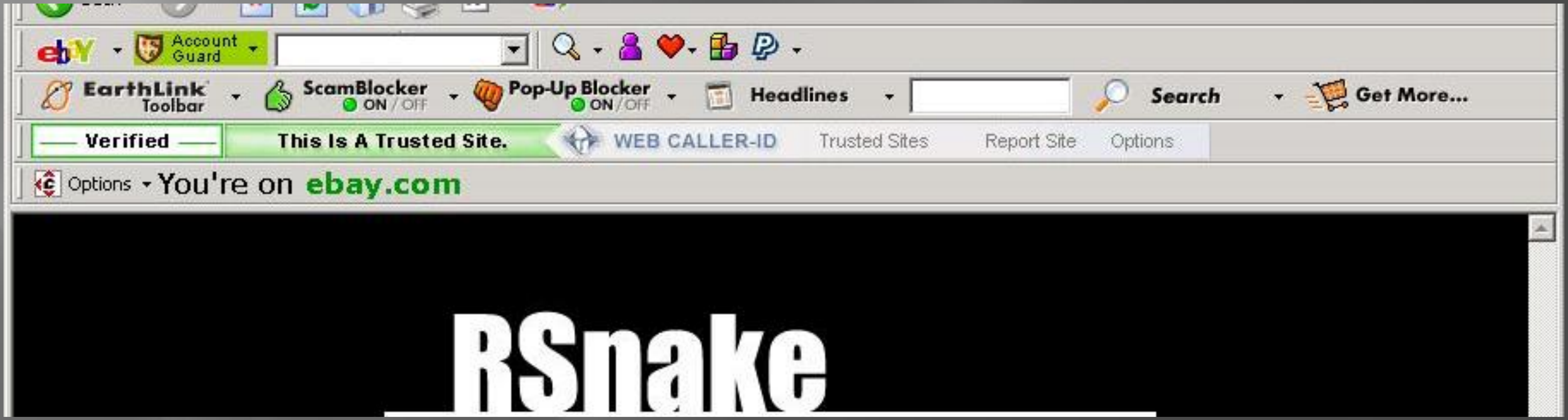
```
<HTML>
  <BODY>
    <FONT COLOR="#FF0000">
    Hello World</FONT>
  </BODY>
</HTML>
```

Hello World

- Mime type issues
- HTML, JS, and CSS differences
- Proprietary URL structure
- Proprietary protocols (jar:, data:, etc…)
- Etc…

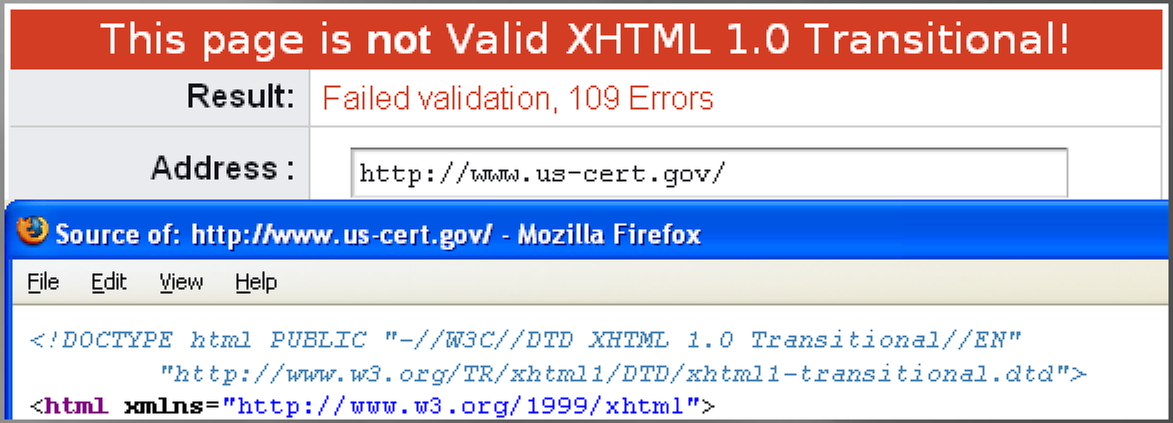
User's Concerns in the Early 2000's

- In the early 2000's bad guys could:
  - Compromise the desktop
  - Use DoubleClick "cookie" malware
  - Spoof the location bar
  - Phish/identity theft
- Toolbars move faster than browsers:



The Browser Community Reacts

- XHTML comes out of WAP2.0 in 2001



This page is not Valid XHTML 1.0 Transitional!

Result: Failed validation, 109 Errors

Address : http://www.us-cert.gov/

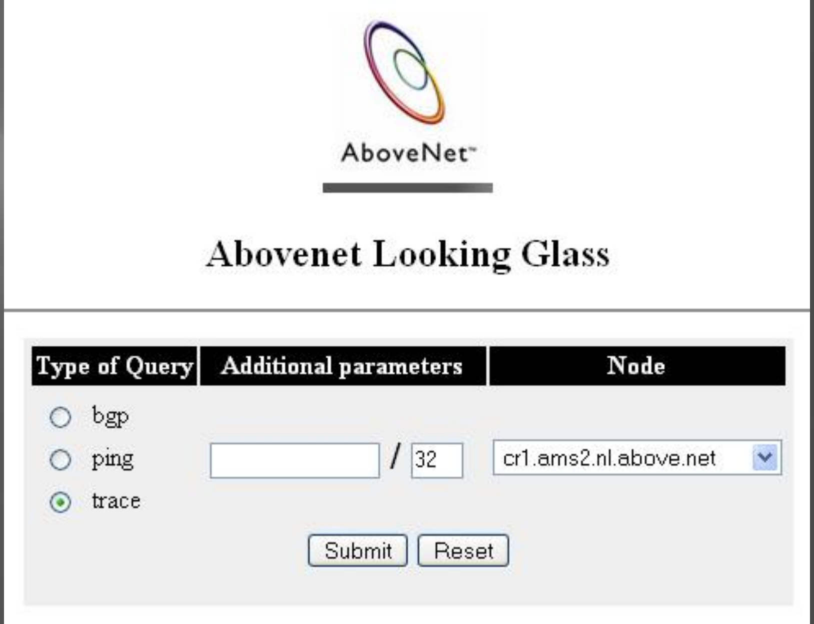
Source of: http://www.us-cert.gov/ - Mozilla Firefox

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
        "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```

- Firefox implements port blocking and allows users to quickly clean history/cookie data
- Anti-phishing filters built in – finally!
- Browsers begin to auto-update and generally begin to fix the major 0days quickly…
- …with the exception of Netscape who always lags.

2005 & 2006 Escalates The Problem

- Samy Worm (Oct 2005)
- It started with a router
  - Intranet port scanning
    - Combining XSS with CSRF
    - Bypassing port restrictions
- Browser history theft
- Exponential XSS
  - Nduja worm
- Desktop compromises…

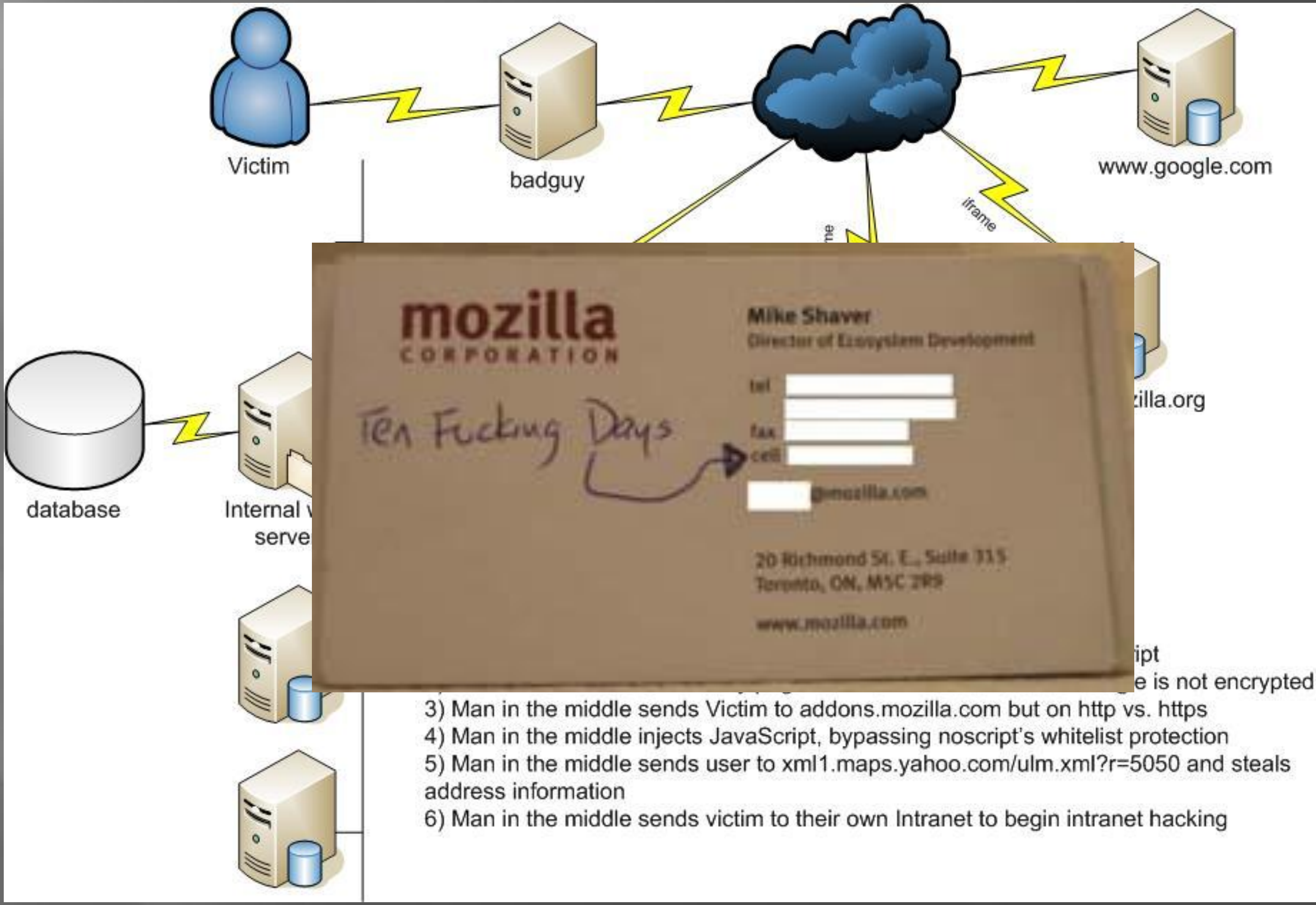


Abovenet Looking Glass

Type of Query	Additional parameters	Node
○ bgp		
○ ping	/ 32	cr1.ams2.nl.above.net
● trace		

2007 Goes Retro

- JavaScript-less:
  - Port Scanning
  - CSS history theft
  - CSRF
- Non HTTPS updates own browsers
- Intranet hacking through split VPN tunnels
- HTTPOnly becomes a standard in 2007 – sorta.
- “10 *ing days”



mozilla
CORPORATION

Ten Fucking Days

Mike Shaver
Director of Ecosystem Development

tel [redacted]
[redacted]
fax [redacted]
cell [redacted]
[redacted]@mozilla.com

20 Richmond St. E., Suite 315
Toronto, ON, M5C 2R9

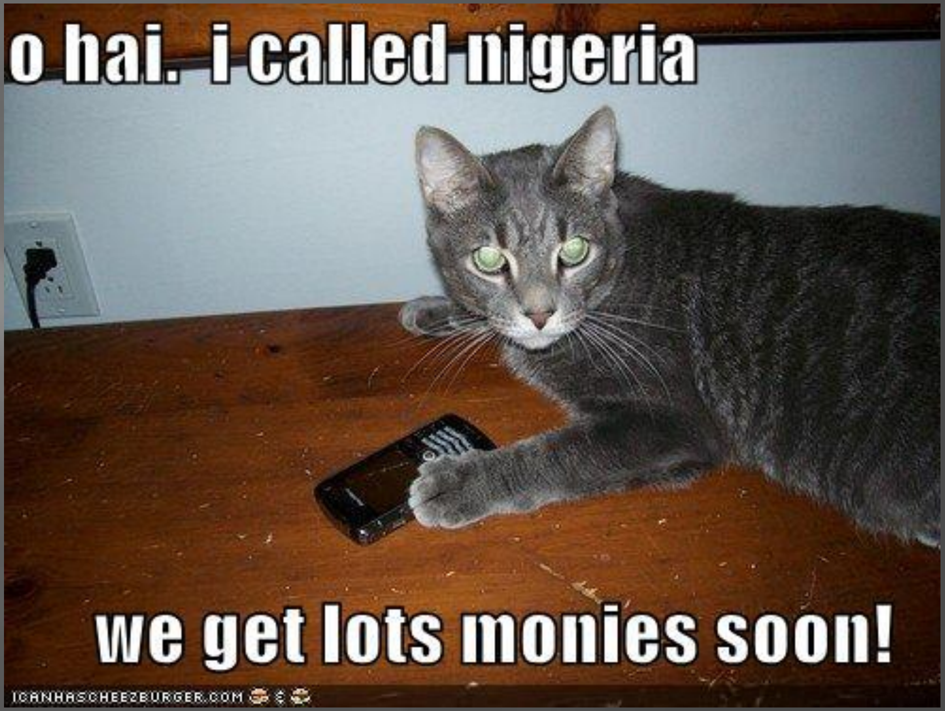
www.mozilla.com

ipt
e is not encrypted
3) Man in the middle sends Victim to addons.mozilla.com but on http vs. https
4) Man in the middle injects JavaScript, bypassing noscript’s whitelist protection
5) Man in the middle sends user to xml1.maps.yahoo.com/ulm.xml?r=5050 and steals address information
6) Man in the middle sends victim to their own Intranet to begin intranet hacking

Our Current and Unfortunate Business Rules

- We cannot encrypt everything (because we don’t own everything)
- We must allow rich HTML (consumers demand it)
- We cannot fix all XSS and CSRF holes (too many)
- Our employees must use the same (all) browsers as our consumers and they must be able to access the website (for QA)
- We cannot block on IP addresses (doesn’t work and alerts the bad guys to what we know)

Today’s Big Browser Threats



Cross domain leakage
Same site/CSRF
De-anonmization
Identity Theft/Fraud
Third party plugins/toolbars – off limits for this talk

Today's Big Browser Threats

- Cross domain leakage
  - XSS (70-90% of sites)
  - Unicode (IE vs FF – RFC war)
  - Referrers (Except file:, meta, etc…)
  - Status Bar
  - Remote CSS, JS, Flash, Java
  - CSS History
  - Onload & Onerror & Image size
  - XML errors
  - Timing attacks
  - DNS Rebinding (NTLM)
  - crossdomainpolicy.xml & Flash
  - …

Today's Big Browser Threats (2)

- Cross/same site request forgeries
  - IMG
  - LINK
  - IFRAME/FRAME
  - OBJECT/EMBED/APPLET
  - BGSOUND
  - SCRIPT
  - Hovering iframes
  - Client side apps
  - X-domain XHR
  - Redirection of URLs
  - Subversive JS file sharing!
  - …

Today's Big Browser Threats (3)

- De-anonmization
  - Cookies/Flash cookies
  - Browser caching (eTag)
  - IE & JS Persistence
  - Machine fingerprinting
  - TCP/OS fingerprinting
  - TCP/clock skew timing
  - CSS history/referrers
  - Offline enabled apps
  - Java Sockets & file:///
  - Statistical observation/MITM
  - …

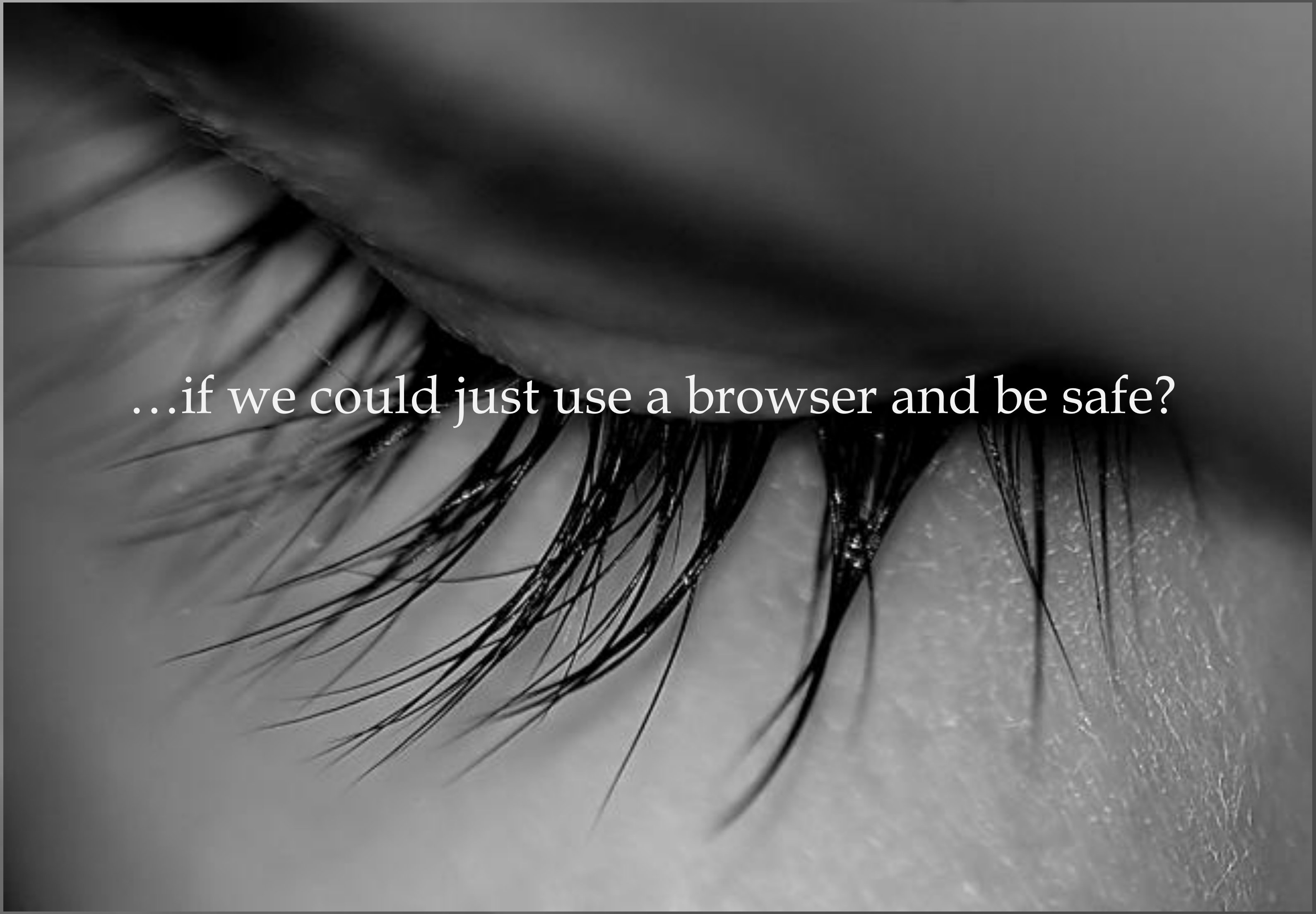
Today's Big Browser Threats (4)

- Identity Theft
  - Phishing on remote domains
  - XSS phish on white listed sites
  - IDN/Punycode
  - Credential theft
  - Embedded basic auth
  - CSS overlay of forms
  - DNS Pharming
  - Keystroke logging/malware
  - MITM
  - Obfuscated HTML
  - Password manager hijacking… FIXED? Uh… no!
  - …

How we tend to convey the "solution"

- Don't use JS
  - Use JS for auth pages
- Don't install anything
  - Install Patches
  - Use plugins (Eg: noscript)
- Don't use social networks
  - Use separate browsers
- Pick secure passwords
  - Don't re-use passwords
- Type the URL
  - Look for the green bar
  - .bank TLD
  - Look for the lock
- …

Wouldn’t It Be Simpler…



…if we could just use a browser and be safe?

Tomorrow's Security Recommendations

- "This sentence is a lie." -Spock. "If you trust me, trust me when I tell you to distrust me." –RSnake.
  - On-page sandboxing/Content restrictions
- Secured "Zones"
- Stop the "Forrest Gump referrer" problem
- Requestor context (click vs img)
- APIs (Callback's, intercept network data, network call to dom mapping)
- Protected/untainted JavaScript
- Standardized Authentication (eg: auto log-out)
- Browser-war truce - Browsersec (standards!)

Questions? Comments?

- Robert Hansen
  - robert _at_ sectheory d0t c0m
  - http://www.sectheory.com/
  - http://ha.ckers.org/
  - XSS Book: XSS Exploits and Defense
    - ISBN: 1597491543
  - Detecting Malice – O'Reilly (TBD)