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

...d...ots?

---

If you want others to be happy, practice compassion.

The Dalai Lama

How much less stress and misery do I find myself wanting... when I laugh off my problems and focus on what's going right? How many less wrinkles I get? I've started my life off by being optimistic... I'm learning to be realistic. To project more patience in my life would be a blessing, both to myself and those who need it with understanding is not needed because... think... understand too well. Maybe a little less of that and a little more faith.

Makes me feel a little more peaceful already.

PS: do not know why MySpace ate my "'s, however, it's kind of cute... if you think of it as a giant Hangman game. Patience patience with the technology
How can we find fraud when we don’t know what it looks like?
Mime type issues
HTML, JS, and CSS differences
Proprietary URL structure
Proprietary protocols (jar:, data:, etc…)
Etc…
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

- 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…
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”
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
  - …
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 (2)
Today’s Big Browser Threats (3)

- De-anonymization
  - 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
  - …
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
- ...

OVERKILL
Nothing succeeds like excess.
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)