

Hacking from the Inside Out

Or: Own a router via a browser

AITP

Jordan Wiens

Wednesday, October 23th



Overview

- Firewalls, shmirewalls
- Basic web application security
- Recent advances
- Demo
- Defenses



How the web was designed

- Pages link to pages
- Content is on those pages
- Only “programs” are on the server



How the web now works

- Applications built on the web
- Applications built on applications built on the web (Mashups)
- Apps³ (Ajax)



Session Management

- Or: how the web-server recognizes you
 - **Cookies**
 - **URLs with persistent values**
 - **Similarly, hidden variables in page**
 - **Some unique identifier that should only belong to your browser**



Why this matters?

- Designed for static content, now used for applications
 - Visiting a link (passive, or active?)
 - POST versus GET
 - Basis for Cross Site Request Forgery (otherwise known as “Hot-linking inherently dangerous)



Step Back

- No exploits here besides weak authentication
- Using the web the way it was designed to be used



On the other hand

- That was “web-server trusts the client”
- What about “client trusts the web-server”?



XSS

- Goal: get bad-guy input to come from the good web-server
- Think: bulletin boards
- Or any web application!



Basic web application security

- Browser flaws
 - **Plugins, extensions, other vulns**
- Same-origin policy
 - **DNS-Pinning**
- Cross-Site Request Forgery (CSRF)
 - **Exploit “server trusting browser”**
- Cross Site Scripting (XSS)
 - **Exploit “browser trusting server”**
- Application Flaws
 - **SQL Injection, weak session handling, lack of authentication, etc.**



Recent Advances

- History Stealing
 - Javascript+CSS
 - Pure CSS!
- Port scanning
 - Javascript
 - Without javascript!
- DNS Pining
 - Not stuck anymore...



Recent Advances contd.

- Identify internal address
 - **Java applet**
 - **Javascript via java calls (ff)**
- Blind server fingerprinting
 - **Cached images similar to evasion of HTTP Auth popups**



Device Hacking Recipe

- **Mix:**
 - **one part internal ip detection**
 - **two parts blind server fingerprinting**
 - **add heavy amounts of default usernames/passwords**
 - **cover in a thin layer of web vulnerabilities to taste**
- **Bake until done**



Demo!

SSID: weaklinksys

WEP: 40bit/hex

Key: aaaaaaaaaa

URL: <http://10.10.10.99/>



Server-Side Protections?

- Referrer checking
- Only accept POST requests
- Strip out bad characters



Protections (most of the time)

- **Browser**
 - **Disabling scripting**
 - **SafeCache/SafeHistory**
 - **LocalRodeo**
 - **Flush saved credentials!**
- **Server**
 - **Sanitize input/output**
 - **Nonce/CAPTCHAs**
 - **ReAuth**



Other protections?

- Browser developers!
- Better frameworks
- Web Application Firewalls
- Additional client tools
 - **RequestRodeo**
 - **Firekeeper**



Meta Slide

- Copyright Information
 - This presentation (sans AITP logo) is released under a Creative Commons BY-SA 3.0 license.
 - The AttackAPI libraries demonstrated are maintained by pdp at <http://www.gnucitizen.com/> and are under a GPLv2.0 license. (this does not include the HTML stylesheets which are under a CC BY-NC-SD 2.5 license and are used here with permission)
- More information
 - Copies of the slides, code, video of the presentation, and links for more information will be available online after the talk: wantingseed.com/sprout/presentations
- Acknowledgements
 - Much of this information is due to the research and hard work of: Jeremiah Grossman, RSnake, pdp, and many, many others. Thanks!

