

Hacking from the Inside Out

Or: letting the browser do the work for you

GatorLUG
Jordan Wiens
Wednesday, June 21th



Overview

- Basic web application security
- Recent advances
- Demo
- Defenses



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/
-



Protections... or not?

- 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 GatorLUG 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!



Hacking from the Inside Out

Or: letting the browser do the work for you

GatorLUG
Jordan Wiens
Wednesday, June 21th



Some Rights Reserved. <http://creativecommons.org/licenses/by-sa/3.0/>

Overview

- Basic web application security
- Recent advances
- Demo
- Defenses



Some Rights Reserved. <http://creativecommons.org/licenses/by-sa/3.0/>

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.**



Some Rights Reserved. <http://creativecommons.org/licenses/by-sa/3.0/>

Firefox password manager plugin, Quicktime exploits, PDF exploits, IE activeX, etc

Description of same-origin policy and original attack.

Same origin: <http://www.mozilla.org/projects/security/components/same-origin.html>

Justin Schuh's blog entry on the topic is excellent:
<http://taossa.com/index.php/2007/02/08/same-origin-policy/>

Princeton attack: <http://www.cs.princeton.edu/sip/news/dns-scenario.html>

CSRF / XSS

Myspace worm and explanation of problem of filtering input, failed CSRF defenses, and XSS: <http://namb.la/popular/tech.html>

Recent Advances

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



Some Rights Reserved. <http://creativecommons.org/licenses/by-sa/3.0/>

History stealing: <http://jeremiahgrossman.blogspot.com/2006/08/i-know-where-youve-been.html>

<http://hackers.org/blog/20070228/steal-browser-history-without-javascript/> (nested CSS calls)

Port scanning: Trap error messages or timeout mechanisms (also, abuse about:cache)

W/o <http://jeremiahgrossman.blogspot.com/2006/11/browser-port-scanning-without.html>

DNS Pining Violation: <http://www.jumperz.net/index.php?i=2&a=1&b=7>
(closed port retry)

Limitations include Host name forced in attacker's domain
(avoid default virtualhost!)

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**



Some Rights Reserved. <http://creativecommons.org/licenses/by-sa/3.0/>

Java socket libraries allow `getHostAddress` calls on a socket:
`alert(java.net.InetAddress.getLocalHost())`

Or longer:

```
function natIP() {  
  var w = window.location;  
  var host = w.host;  
  var port = w.port || 80;  
  var Socket = (new  
java.net.Socket(host,port)).getLocalAddress().getHostAddress();  
  return Socket;  
}
```

Same origin still applies, but doesn't hurt us.

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



Some Rights Reserved. <http://creativecommons.org/licenses/by-sa/3.0/>

Phenoelit.de's default password list: <http://www.phenoelit.de/dpl/dpl.html>
Web vulnerabilities? Yikes. Too many to count.

Demo!

- SSID: weaklinksys
- WEP: 40bit/hex
- Key: aaaaaaaaaa
-
- URL: <http://10.10.10.99/>
-



Protections... or not?

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



Some Rights Reserved. <http://creativecommons.org/licenses/by-sa/3.0/>

Referrer checking -- recent flash vulnerability, not practical because of privacy issues, meta-refresh, javascript populated frame submission, generated popup

More information at:

<http://www.cs.kuleuven.ac.be/publicaties/rapporten/cw/CW448.abs.html>

POST only very marginally more difficult to implement

Sure you got all naughty characters? Harder and harder to keep track of all the various ways to embed attacks. Simpler to white-list valid characters than try to black-list bad ones:

<http://ha.ckers.org/xss> (XSS cheat sheet)

Protections (most of the time)

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



Some Rights Reserved. <http://creativecommons.org/licenses/by-nc/3.0/>

NoScript / don't forget other embedded tools like pdf, quicktime, and flash, which have recently had vulnerabilities, and often have scripting mechanisms of their own

<http://www.safecache.com/> and <http://www.safehistory.com/>

Local rodeo: <http://database.net/labs/localrodeo/>

Sanitize the “right” way, libraries, white-list only, multiple passes, as necessary.

Change default passwords internally (intranet brute-forcing via browser is slow!)

Nonce works sometimes (Samy)

Amazon approach -- auth again.

Other protections?

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



Some Rights Reserved. <http://creativecommons.org/licenses/by-sa/3.0/>

IE team meeting with Rsnake, definitely taking issues seriously, working on implementing better transparent mechanisms.

Firefox has the work of Gervase Markham
<http://www.gerv.net/security/content-restrictions/>

RequestRodeo -- inline proxy acting much like a web application firewall for the purpose of protecting clients instead of servers.

<http://www.nongnu.org/requestrodeo/>

Firekeeper -- snort for firefox (good idea -- use browser's own rendering engine, processing, not quite there yet practically)

<http://firekeeper.mozdev.org/>

Meta Slide

Copyright Information

- This presentation (sans GatorLUG 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!



Some Rights Reserved. <http://creativecommons.org/licenses/by-sa/3.0/>