The Security State of WordPress' Top 50 Plugins

Executive Summary

Checkmarx's research lab identified that more than 20% of the 50 most popular WordPress plugins are vulnerable to common Web attacks, such as SQL Injection. Furthermore, a concentrated research into e-commerce plugins revealed that 7 out of the 10 most popular e-commerce plugins contain vulnerabilities. This is the first time that such a comprehensive survey was prepared to test the state of security of the leading plugins. In total, 8 million vulnerable WordPress plugins were downloaded.

The impact? Hackers can exploit these vulnerable applications to access sensitive information such as personally identifiable information (PII), health records and financial details. Other vulnerabilities allow hackers to deface the sites or redirect them to another attacker-controlled site. In other cases, hackers can take control of the vulnerable sites and make them part of their botnet heeding to the attacker's instructions.

The issues we describe in this report lie within WordPress' extensive plugin offerings. These security gaps within the plugins allow hackers to use the platform as vehicles for mass infections and malware distribution. Since we do not focus on the security of the basic platform, our discussion can be applied to any marketplace that provides third-party extensions and applications.

In this report we present our findings of running multiple security scans against the source code of the plugins. In-between the scans a handful of plugins were completely fixed. However, it is our belief that websites containing the vulnerable versions still remain outdated due to lack of security knowledge, admin resources, and scheduling reasons. This report also contains recommendations and mitigation measures which plugin developers, Web admins and platform providers should take when developing and installing third-party extensions.

Background and Motivation

WordPress is the most popular blogging Content Management Systems (CMS). An open-source platform, WordPress powers more than 60 million websites and 18% of the Web[1]. The strength of WordPress lies in its ability to customize sites through its extensive offering of tens of thousands of plugins and themes. Types and functionalities of plugins vary - from adding a contact form to the site to optimizing WordPress blog for search engines and even publishing new posts to Facebook.

Currently, any developer can add a WordPress extension to enhance the basic blogging platform. Although there are some set of coding standards and recommendations, there is no security guidance or requirements that a plugin developer needs to adhere to.

[1] http://wordpress.org/

As such a widely-distributed system, WordPress is also a popular target for attacks. A vulnerability against a plugin propagates across millions of websites. A hacker exploiting a plugin's vulnerability can infect millions of websites as the security industry – and bloggers worldwide – have already witnessed. Examples of such mass infections include the TimThumb LFI vulnerability which compromised[2] 1.2 million websites and the redirection[3] of 200,000 WordPress based pages to rogue sites. More recently, the finding of a critical vulnerability in a plugin hosted by sites such as Mashable[4] made headline security news.

To be one step ahead of the hackers and identify the next risk targets, Checkmarx's research labs have conducted a comprehensive analysis of the source code of these third party WordPress plugins. The research did not focus on the security of the WordPress core platform, but rather on the security state of the most popular plugins.

Methodology

WordPress is an open-source community platform, and so are its plugins. Given the source code, Checkmarx ran an automated static code analysis scan against the most popular plugins. Checkmarx looked at the top 50 most downloaded (i.e., "popular") plugins, as well as the top 10 most downloaded plugins dedicated to e-commerce.

Checkmarx performed scans of the general top 50 most downloaded plugins on two separate occasions. The first scan was conducted in early January 2013. During this survey, Checkmarx found 18 vulnerable plugins which amounted to 18.5 million downloads. Due to the high-profile of some of these plugins, Checkmarx alerted the developers of four of the vulnerable plugins to their vulnerabilities and worked with them towards their fixes.

After 6-months, in early June 2013, the WordPress plugin repository had shown that all the 18 vulnerable plugins had updated versions. In order to test the security posture of these updated versions, Checkmarx re-ran the scan on the new versions of all 18 plugins. To note, these 18 plugins still remained in the top 50 most popular plugins.

Separately, in early June 2013, Checkmarx also took a look into the top 10 most downloaded e-commerce plugins.

For all scans, Checkmarx tuned the tool to alert whenever one of the following vulnerabilities existed in the code. Focus was made on these particular vulnerabilities due to their high risk and impact severity in a hosted environment.

- **SQL Injection (SQLi).** Allows the execution of commands on the back-end server, such as the extraction of sensitive information.

[2] http://www.darkreading.com/database/hackers-timthumb-their-noses-at-vulnerab/231902162
[3] http://www.networkworld.com/news/2012/030612-30000-wordpress-blogs-infected-to-256993.html
[4] http://www.theregister.co.uk/2012/12/27/wordpress_cache_plugin_vulnerable/

In a hosted environment, a SQL Injection attack can further be used as a stepping stone for attacks against non-vulnerable sites. The attacker in this environment can manipulate the database shared by both vulnerable and non-vulnerable sites.

- **Cross Site Scripting (XSS)**. Allows the running of a script on the client in order to bypass access controls. This kind of vulnerability targets all visitors to the site.
  In a hosted environment, this problem is exacerbated where an attack can also be used to steal the session cookies of the site admin. Effectively, it allows the attacker to imposter the admin.
- **Cross Site Request Forgery (CSRF).** Allows the attacker to perform an application-level transaction on behalf of the victim.
  In a hosted environment, an attacker can log onto the site on the admin's behalf and perform nefarious activities such as re-directing users or performing admin transactions.
- **Remote/ Local File Inclusion (RFI/ LFI)**. Allows the uploading of a potentially malicious file to the server.
  In a CMS environment this is a particularly critical vulnerability as the point of the CMS is to manage and deliver site-related files.
- **Path Traversal.** Allows the attacker to crawl the Web pages.
  In a CMS environment an attacker can exploit this type of vulnerability to access and map all hosted files.
  Checkmarx tested the plugins for Path Traversal vulnerabilities only during the second scan.

Findings

Findings of running a security scan on the source code of WordPress plugins in early June 2013:

1. **20% of the 50 most popular WordPress plugins are vulnerable to common Web attacks. This amounts to nearly 8 million downloads of vulnerable plugins.**
   Namely, these plugins are vulnerable to: SQL Injection (SQLi), Cross Site Scripting (XSS), Cross Site Request Forgery (CSRF), and Path Traversal (PT).
2. **7 out of top 10 most popular e-commerce plugins are vulnerable to common Web attacks. This amounts to more than 1.7 million downloads of vulnerable e-commerce plugins.**
   These plugins are vulnerable to SQLi, XSS, CSRF, RFI/ LFI and PT.
3. **There is no correlation between the number of Lines of Code (LOC) and the vulnerability level of the plugins.**
   Every line of code has the potential impact of introducing a vulnerability. But Checkmarx has found that the opposite does not hold true. Meaning, the smaller the code does not necessarily mean the safer the code. On the contrary – some plugins that included only a few thousand lines of code contained more types of vulnerabilities than plugins containing tens of thousands lines of code.
4. **Vulnerable top 50 general plugin types vary.**
   These include, but not limited to, plugins which are used for:
   - Ecommerce such as a shopping cart

  - Content management such as feed aggregators, related links and checking of broken links
  - Site development such as APIs for Web development and transforming a Web site to a mobile app
  - Social networks - from linking to Facebook to establishing an internal organization network.

5. **Only six plugins were completely fixed in a 6-month time period- although all plugins updated their versions during this time.**
   A first scan ran in January 2013 showed a higher rate of vulnerable plugins where more than a third (18 out of 50) of the plugins were vulnerable. In total, this meant that nearly 18.5 million vulnerable plugins were downloaded. Vulnerabilities in that first scan also presented the existence of RFI/ LFI vulnerabilities.
   The second scan, conducted in early June 2013, was performed on the updated versions of all plugins. However, only six of these updates were free of those previously found vulnerabilities. These were:
   - BuddyPress – creates a social network for the organization. # Downloads: 1,319,743. Alerted by Checkmarx to their vulnerabilities.
   - BBPress – forum software. # Downloads: 483,283. Alerted by Checkmarx to their vulnerabilities.
   - E-Commerce – shopping cart plugin. # Downloads: 2,209,352. Alerted by Checkmarx to their vulnerabilities.
   - Woo Commerce – an e-commerce store. # Downloads: 469,503. Alerted by Checkmarx to their vulnerabilities.
   - W3 Total Cache – site optimization by caching. # Downloads: 1,450,980. Most likely fixed as part of a security overhaul following an external full disclosure of some vulnerabilities[5].
   - Super Cache – site optimization by caching. # Downloads: 3,984,976. Most likely fixed as part of a security overhaul as with W3 Total Cache.

[5] http://seclists.org/fulldisclosure/2012/Dec/242

Plugin	LOC	# Downloads	SQLi	XSS	CSRF	PT
[redacted] Lists related entries	4,682	2,093,718	🟥			
[redacted] Tests the site for broken links and missing images	20,636	1,493,609			🟥	
[redacted] Add links to Facebook	8,857	1,029,626		🟥		
[redacted] A review system for comments	26,326	1,002,808	🟥	🟥	🟥	
[redacted] An RSS aggregator	15,481	622,894	🟥	🟥		
[redacted] Site backup	247,816	464,212		🟥		🟥
[redacted] Embeds Flash and HTML5 video	13,676	380,551		🟥		🟥
[redacted] Saves contact from data	22,591	372,150	🟥	🟥	🟥	
[redacted] An alternative WordPress editor	11,395	263,171		🟥		🟥
[redacted] Management of site statistics	3,593	152,467	🟥	🟥	🟥	🟥
[redacted] Transforms WordPress sites to mobile apps	3,820	84,863		🟥		

Table 1: A summary of the vulnerabilities found in the top 50 most popular general plugins (June 2013)

Plugin	LOC	# Downloads	SQLi	XSS	CSRF	PT	RFI/ LFI
[redacted] Shopping cart	22,277	519,462	🟥	🟥	🟥		
[redacted] Online store setup	39,950	380,800		🟥	🟥		
[redacted] Paypal shopping cart	1,302	274,273		🟥			
[redacted] Store management and performance	42,587	234,134		🟥	🟥		

[redacted] Store management	56,162	104,420	🟥	🟥	🟥	🟥	
[redacted] Shopping cart	42,073	98,521		🟥		🟥	🟥
[redacted] Shopping cart	19,885	93,537		🟥	🟥		

Table 2: A summary of the vulnerabilities found in the top 10 most popular e-commerce plugins (June 2013)

Mitigation

There are three major affected parties when it comes to WordPress plugin vulnerabilities: site admins, plugin developers and WordPress itself.

We provide recommendations for each of these parties, and in essence these can be extended also to other common platforms that provide extensions in the form of plugins and apps.

Recommendations for Web Admins

Whether a WordPress-based site admin for a large enterprise or a small business, here's what you can do:

1. **Download plugins only from reputable sources. For WordPress, this means WordPress.org**
   Since anyone can develop a WordPress plugin, hackers can also exploit this vulnerability to hide their own nefarious plugin. Although going through a reputable marketplace will not guarantee only harmless plugins[6], you should consider this as a first line of defense.
2. **Verify the security posture of the plugin by scanning it for security issues**
   If you have the source code – and most probably you do since the plugins are open-source - run a static source code analysis tool which will provide you with the plugin's "bill of health". Advanced scanners can even point you with the optimal and quickest fix recommendations. If you cannot manage the plugin's source code, you can run any of the WordPress dynamic security scanner plugins. The downside? These test only specific scenarios and so the scanners lose out on coverage.
3. **Ensure all your plugins are up to date**
   Do not ignore all those notification emails of an upgraded plugin version. You can even use a purposeful WordPress plugin that notifies admins on updates to other installed plugins. There are also third party services which provide a plugin update notification and management offering.
4. **Remove any unused plugins**
   The code of old, unused plugins remains on the server – even if the plugins are inactive. Schedule plugin spring cleaning as part of your WordPress site admin activities.

[6] http://blog.sucuri.net/2013/04/wordpress-plugin-social-media-widget.html

Recommendations for Plugin Developers

1. **Integrate security within the plugin development**
   The later a vulnerability is discovered, the higher the cost of fixing. But there's more: a simple fix does not guarantee that the plugin users updated their sites. And until they do, these admins are left with the vulnerable plugins. The solution? Bake security within the plugin development as part of your secure Software Development Life Cycle (SDLC) process.
2. **Run the plugin through a code scanner to ensure that it stands up to a security standard**
   Perform a source code analysis test during development and prior to release.

Recommendations for WordPress and Other Application Platform Providers

Although this research was conducted against WordPress plugins, it is important to recognize that many other platform providers suffer from similar problems. All app marketplaces – whether Web or mobile - which provide the platform for building plugins and extensions run the risk of vulnerable extensions on their site.

We suggest app marketplaces follow these security best practices:

1. **Enforce a security policy on apps that enter the marketplace.**
   Although the security of a plugin may seem to be the developer's problem, platform providers are held responsible in the light of users. Make security your competitive edge and build a marketplace where users can feel secure to return to.
2. **Authorize only apps that passed the security bar.**
   Provide certification to those apps that stand up to the security policy.

About Checkmarx

Checkmarx is the developer of next generation Static Application Security Testing (SAST) solutions.

Checkmarx provides the best way for organizations to introduce security into their Software Development Lifecycle (SDLC) which systematically eliminates software risk. The product enables developers and auditors to easily scan uncompiled / unbuilt code in all major coding languages. CxSuite's application security testing is available in both "On Premise" and "On Demand" configurations. The security testing scans for hundreds of the most prevalent security vulnerabilities as well as business logic flaws, best coding practices and more.

Checkmarx was recognized by Gartner as sole visionary in their latest SAST magic quadrant and as Cool vendor in application security.

Customers include hundreds of Fortune 500, government organizations and SMBs in over 30 countries.

For more information, visit www.checkmarx.com or call 1-917-470-9501.