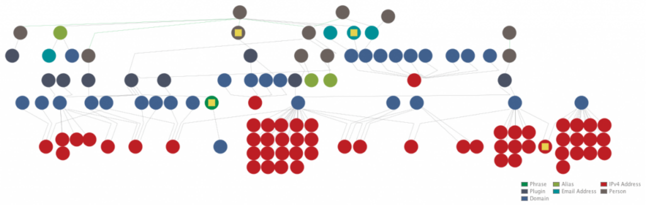
9 WordPress Plugins Targeted in Coordinated 4.5-Year Spam Campaign

This entry was posted in WordPress Security on September 20, 2017 by Mark Maunder   52 Replies



On Tuesday last week we published a post that described how someone had released an update to the Display Widgets plugin which contained a backdoor that allowed them to publish content to any site using the plugin. We also described how they exploited that backdoor to publish spam.

We then wrote a follow-up piece which we published on Wednesday identifying the person who modified Display Widgets as Mason Soiza.

In today’s post, we are publishing research showing a coordinated effort by the same spammer that targeted WordPress plugins over a 4.5-year period. In some cases, site owners opted in to a vague agreement that didn’t make it clear that their sites would be serving spam; in other cases, plugins were simply “backdoored” to allow posting without a site owner’s permission.

The content served from the ad network included ads for payday loans and escort services in the UK. The activity started in 2013 and ended this month. All nine plugins discussed today are linked to the same malicious actor, either through a financial trail or through the ad-serving domains used which share IP addresses.

Which Plugins Were Targeted And Their Current Status

We now have evidence that, in addition to the Display Widgets plugin, eight other plugins have been affected by spam in the past and all are linked in some way. These incidents range from May 2013 to September 2017.

In four cases we have also been able to uncover financial transactions linked to Mason Soiza, either directly or indirectly. In other cases, we have no transaction, but we share the data we have which helps establish attribution.

It is **important to note** that **all plugins listed here no longer have the ability to inject spam into websites.** They are either safe, or in two cases have been removed from the WordPress plugin repository.

Here is the list of plugins we discuss today and their current status. When we list a plugin as "safe," it means the plugin does not currently include any spam code that we are aware of.

- 404 to 301: Safe
- Display Widgets Plugin: Safe, but no longer maintained. Use Jetpack's Widget Visibility Module instead.
- WP Slimstat: Safe
- WP Maintenance Mode: Safe
- Menu Image: Safe
- NewStatPress: Safe
- Financial Calculator Plugin: Safe. Never included malicious code, but Soiza did have access for some time this year.
- Weptile Image: Removed from repository
- No Comment: Removed from repository

Some plugin authors spoke with me anonymously for this post, but most went on the record. My thanks to all authors who helped us compile this data, because they have done the WordPress community a service in helping us understand the impact of these spam networks and who they belong to.

Special thanks to the following plugin authors:

- Display Widgets original author Steph Wells. We spoke via Skype voice and IM. She opened the floodgates.
- 404 to 301 plugin author Joel James. We spoke via Skype IM.
- WP Slimstat author who goes by the alias Jason Crouse. We spoke on the phone.
- WP Maintenance Mode developer George and current owner Andrian. We chatted via email.
- Menu Image plugin developer Alex. We chatted via email.
- Financial Calculator plugin author Ciprian Popescu. We chatted via LinkedIn.

Supply Chain Attacks and Avoiding Bad Partners

What we are describing here today and in our previous posts is known as a supply chain attack. A few examples of supply chain attacks are:

- A hacker breaks into a software company and infects its code with malware, which then gets distributed via the "supply chain" to that company's customers.
- A developer's account is compromised and the hacker uses the account to release malicious code to customers.

I have managed to verify that in most of these cases a malicious actor convinced a developer or company to add their code, usually through a financial arrangement, to a WordPress plugin. The malicious actor then uses that gateway to inject spam into customer websites.

As developers, we have a giant target painted on our backs. We are constantly releasing code to customers, and if someone can inject their malicious code into our software, they can infect thousands or even millions of customers at once, rather than just a few at a time. It is very important for us to guard our login credentials and code and important for us to choose our partners wisely.

One could argue that the plugin authors should have been more careful about who they partnered with. I think that is 20/20 hindsight because most partnerships require trust at some point in the relationship. If you put Google ads on your website, you trust Google won't display offensive content. If another ad network pays you to include their code that displays their ads, you trust them not to include spam or malicious content. In this case, the trust was broken, and in some cases customers paid the price.

My hope in publishing this story, and through the collaboration of these plugin authors who have kindly come forward, spoken up and shared their story, is that we as a community will learn how to avoid bad partnerships and how to form partnerships that do work.

The rest of this post goes into detail about each plugin that we analyzed, how it is linked to the spammer we have been tracking and which versions were affected over what time frame.

Display Widgets Plugin: An Update

We wrote about Display Widgets in detail last week so I am going to summarize the status here and provide an update from the WP plugin team.

I chatted with original plugin author Stephanie Wells via Skype voice and IM. She sold the plugin to Soiza on May 19, 2017, and we've confirmed the transaction which came from pp@linkrocket.net.

On June 30, 2017, a backdoor that allowed remote unauthenticated content updates was added to the plugin and remained there until the plugin was removed from the repository on September 8, 2017. The affected versions are version 2.6.1 to version 2.6.3.

The domains used to serve spam were **geoip2.io and stopspam.io**. The WordPress.org plugin team have removed the historical versions that contained malicious code from the WP repository by the plugin team.

The WordPress.org plugin team have removed the offending code from the latest version of the Display Widgets plugin and incremented the version number so that sites can update to the clean code. I chatted with Otto, one of the plugin repository maintainers, who had this to say on WordPress Slack:

"*Version 2.7 is identical to the 2.05 version, codewise. This was done to remove the offending code from the plugin and to help eliminate it from sites that had updated it to the 2.6 series. The plugin is now closed and will receive no more updates. We recommend all users find an alternative plugin for their needs. Given what the intended functionality of the plugin is, I recommend using the Widget Visibility option in Jetpack instead.*"

You can find out more about Jetpack's Widget Visibility module on this page.

404 to 301 Plugin

We received data that Soiza was also involved in 404 to 301 and so we reached out to the author.

We previously wrote about 404 to 301 distributing spam in August of last year. We also did a follow-up post with more technical detail and to counter criticism we received.

I chatted with Joel James, the author of 404 to 301, for this post. He was kind enough to share transaction details with me. Based on a PayPal transaction dated May 9, 2016 from pp@linkrocket.net using Soiza's name, Soiza also appears to be the perpetrator in this case.

404 to 301 included Soiza's code to distribute spam to websites from June 1, 2016 until July 12, 2016. The affected versions are 2.2.0 to 2.2.8.

The domain used to fetch spam and inject it into the plugin was **wpcdn.io**. You can find the code here if you search for the domain.

The plugin injected spam content onto sites, including a domain for a UK-based escort service which we verified belongs to Soiza in the Display Widgets follow-up post.

The plugin asked users to agree to terms before injecting their content with spam, but the terms were lengthy and the ‘spam’ clause appeared right at the end.

Joel James now controls the code to this plugin and has confirmed that the plugin is now clean. As a reminder, this happened over a year ago. At this point I have no hesitation in recommending that you use the 404 to 301 plugin.

WP Slimstat Plugin

We also received data that Soiza was involved with WP Slimstat. I chatted with the author of WP Slimstat on the phone. He goes by the alias Jason Crouse, and would prefer to remain anonymous. He was kind enough to provide all of the data we requested.

He received a PayPal transaction dated November 22, 2015 from email jj@linkrocket.net and name “Garri Kiekbusch.” The email uses the same domain as two other emails that we have established belong to Soiza, but the transaction used a different name. This email was also displayed on an archived linkrocket.net home page along with the other two. The name may be a business partner or an alias.

From October 22, 2013 until August 10, 2016 (2 years 10 months), WP Slimstat had code in it that is very similar to Soiza’s spam code. That period includes versions 3.4 to version 4.3.7.

The domains used for Slimstat were **wordpress.cloudapp.net, then wpcdn.io and later api.wp-stats.io.** You can find a code sample on this page for 4.3.7.

It’s important to note that **wordpress.cloudapp.net and wpcdn.io share the same IP address at the time of writing. api.wp-stats.io**

historically also shared that IP address. This links the three domains and we will refer to this later in the post.

Early versions of Slimstat that included Soiza's code included a checkbox that you had to opt into, but the checkbox did not mention spam or ads. Later versions did mention that ads would be displayed.

I'd like to repeat that I have chatted with the author of this plugin on a fairly long phone call and he shared his real identity with me. He prefers to remain anonymous, but is in control of the plugin once again and has been for a year at this point. WP Slimstat no longer contains any spam code, and has not for some time.

WP Maintenance Mode Plugin

Once we identified several plugins linked with Soiza, we started looking through the repository to find plugins using the same domains. We were surprised to learn that WP Maintenance Mode was involved. It is a very popular plugin with around 500,000 active installs. From June 20, 2013 to September 1, 2014, WP Maintenance Mode had code that allowed content to be remotely injected into a website without permission.

The versions that were affected are 1.8.9 up until the code was removed in version 2.0.0. At that point, ownership of the plugin changed hands to the new owners who appeared to remove the code.

I chatted to the new owners via email and they don't have any information that can help us link this to other affected plugins. I also reached out to the previous owner and managed to connect with him via email but was not able to get any helpful data.

The one link we do have is the domain that WP Maintenance Mode was loading ads from. It was **wordpress.cloudapp.net, one of the domains used by WP Slimstat** which we confirmed was paid by jj@linkrocket.net.

So this appears to, at the very least, be linked to the same advertiser that used Slimstat.

The difference with this plugin is that there was no opt-in required by the user. The advertiser could simply inject content into the website at will. You can find an example of the code in version 1.8.11 on this page.

This page on a hosting company website shows a user in 2013 asking why WP Maintenance Mode is slow in loading. It turns out it was contacting wordpress.cloudapp.net to load content in the background.

NewStatPress Plugin

We found this plugin was involved when searching for the domain associated with the plugins above. We found this post from 2013 where a user was reporting the plugin slowing down when loading something from wordpress.cloudapp.net.

NewStatPress would randomly select a WordPress hook like get_header, get_sidebar, wp_head or wp_footer and print out a decoded JSON blob which it fetched from the **wordpress.cloudapp.net** domain.

This plugin, WP Maintenance Mode and WP Slimstat all use the same domain, which is linked to jj@linkrocket.net.

This plugin ran this code from May 5, 2013 until June 29, 2013. The versions affected were 0.6.2 to version 0.6.7. You can see a sample of what the version 0.6.2 code looked like on this page.

I posted a contact request on this plugin's support forum but did not get a response in time for publication.

Menu Image Plugin

In our research, we found that Menu Image was also loading ads and injecting them into site content. This post, where users are discussing spam being injected into the plugin, is what initially caught our attention.

The Menu Image plugin had opt-in ads that displayed if the user agreed to a long license that was vague in describing what would happen to your website. I chatted to the author via email and he was very helpful in helping us identify whether these ads are linked to the other plugins.

It appears that the author was paid by someone with a different name, tied to a different company. The domain used to load the content was **apistats.net**.

There is one tentative link: **apistats.net at one point shared an IP address with the domain api.wp-stats.io** which was a domain used by WP Slimstat which we linked to jj@linkrocket.net (that Soiza domain again). The api.wp-stats.io domain also shared an IP at one point with wordpress.cloudapp.net and wpcdn.io, which as we know are other domains used to serve content to these affected plugins.

Menu Image contained this code from July 7, 2016 until November 20, 2016 – slightly less than a year ago. The versions affected were version 2.6.4 to versions 2.7.0.

You can find a diff that shows when the code was added, in the file notice.php, on this page on github.

Weptile Image Plugin

This plugin caught our attention when we did a Google search for functions used by other plugins to inject content. This plugin also uses the notice.php file with the same function names as the Menu Image plugin. It also loads content from the apistats.net domain.

Weptile included code to remotely load content on July 2, 2016. It appears to have been removed from the repository on the same day.

According to this article, the plugin later changed ownership. The date the post was published is September 25, 2015, about 1 year before the injection code appeared.

The domain used to inject content was **apistats.net** which, as we pointed out above, does have a tentative link to Soiza and linkrocket.net.

I was not able to contact any authors associated with this plugin.

No Comments Plugin

This plugin caught our attention because it used the wpl.io domain to inject content into websites. That domain was hosted on the same IP address as wpcdn.io which we linked directly to Soiza.

The plugin contained code that injected content from April 25, 2014 until it was removed from the repository in October 2016. From version 1.1.5 the plugin contained code that could deliver spam.

The code in this plugin checked to see if a user was signed in. If the user was not signed in, the plugin would fetch a blob of content from the wpl.io domain, decode it and inject it. The plugin would print meta-data in the affected site's header and replace keywords in the content with links. The links were hidden from view using CSS. There was no option to disable this feature.

I was not able to contact the authors of this plugin because it has been removed from the WordPress repository.

Financial Calculator Plugin

This plugin gets an honorable mention. It has never contained any spam code. However, the author, Ciprian Popescu, was kind enough to chat with me. It turns out that Soiza (pp@linkrocket.net) paid $600 to get access to this plugin but never used his access to add any code.

The date on the transaction was April 24, 2017. After we chatted, Ciprian reverified that Soiza no longer has access, and assured us that the plugin is safe.

A Coordinated Effort to Spam Via WordPress

What this research reveals is a coordinated effort by a malicious actor to use WordPress plugins to serve spam that started in 2013 and lasted until this month.

In some cases, the site owner had to opt in, but the opt-in message was either vague, misleading or obfuscated. In other cases, WordPress sites would serve ad content without the site owner being aware.

The content from these ad servers includes ads for payday loans and ads for escort services based in the UK.

How Plugin Authors Can Avoid Falling Victim

WordPress is now the most popular content management system in the world by a significant margin. It is important that we, as plugin authors, recognize that we are attractive targets for bad people. If they can infect just a handful of plugins with malicious code, it can be worth millions.

Stay secure and keep your code and customers secure. If you consider selling your plugin or a partnership, make sure you understand who you are selling to and what their history is. Also get a clear understanding of what their intentions are. Get references. You might also consider reaching out to the community, including Wordfence, to find out if anyone has any data on your buyer.

I would also strongly recommend to plugin authors to never give ‘commit’ access to a partner. Have them send you a patch or feature request and review and commit the code yourself.

Make sure your buyer or partner understands the plugin directory guidelines.

How WordPress Site Owners Can Avoid Spam in Plugins

As a WordPress site owner it is important that you keep track of which plugins are removed from the WordPress repository. In several cases, the plugins affected in this post were removed multiple times from the WordPress repository. If you are using Wordfence, it will notify you that a plugin you are using has been removed from the repository.

When you see a plugin removed, go to the support forum and ask the developer what happened. If they aren't clear and transparent in their answer, you might considering moving on and using a different plugin.

Stay abreast of security developments by subscribing to our mailing list. When we discover that a plugin has been spamming, we immediately notify the community and take action.

If a plugin does include "terms" that you need to agree to, **read the entire T&C document included, especially the end**. In several cases the plugins affected did have an 'opt-in' checkbox with terms that said ads would be injected into your website. The problem was that the language describing the 'ads' was at the very end of a very long document.

If a plugin has an 'opt-in' checkbox for any kind of data sharing, make sure that it clearly describes *what* is being shared with or fetched from a remote server.

A Final Note

As a final note, I would like suggest that the WordPress.org maintainers enable two-factor authentication and enforce a strong password policy on the plugin repository. I'd like to suggest TOTP which integrates with password managers like 1Password.

These supply chain attacks will only get worse. This will probably be a large and complex project because it involves enabling two-factor on the code repository and other systems. So this will take time, because it risks breaking things.

Mark Maunder – Wordfence Founder/CEO.

Credits: Thank you to all the plugin authors who we interviewed and who shared data with us that helped us compile the connections described in this post. They are listed at the top of this post. Thanks to Brad Haas who compiled the data we gathered into a Maltego investigative graph (pictured at the top of this post) which was invaluable when understanding the complex connections in this story. Special thanks to Åsa Rosenberg, Ryan Britton, Matt Barry, Colette Chamberland, Dan Moen and other Wordfence team members who provided valuable assistance. Thanks to Andie La-Rosa for editing.