Race condition in WordPress plugin allows php remote code execution

A. Cervoise
antoine.cervoise@devoteam.com

07/09/2014

Who am I?

- French
- IT Security Consultant at Devoteam
- Cigar smoker
- Music lover

About a race condition in a WordPress plugin

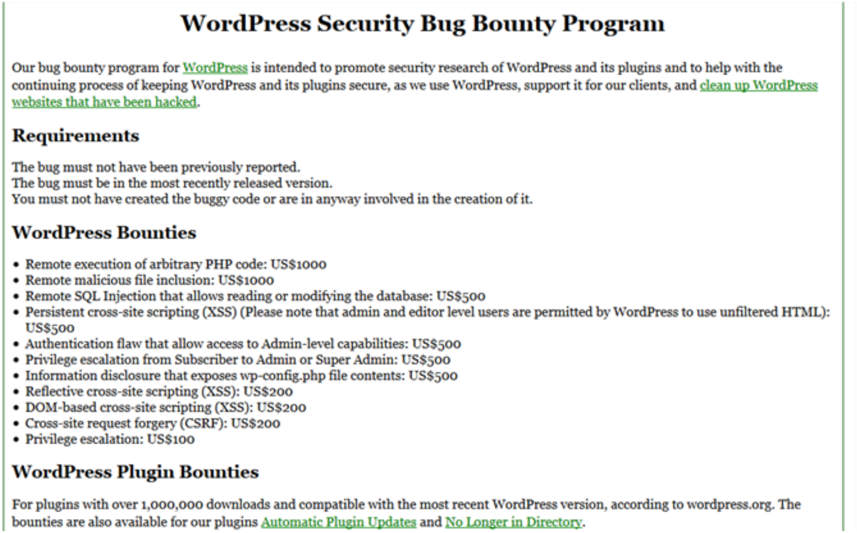
- A funny vulnerability
- Link to a story to tell

Plan

- A short story
- A funny vulnerability
  - WordPress and WP Photo Album Plus
  - Let's find it!
  - Let's patch it!

Why am I looking for vulnerabilities?

- Trying new stuff
- Learning
- etc.
- But not this time!



WordPress Security Bug Bounty Program

Our bug bounty program for WordPress is intended to promote security research of WordPress and its plugins and to help with the continuing process of keeping WordPress and its plugins secure, as we use WordPress, support it for our clients, and clean up WordPress websites that have been hacked.

Requirements

The bug must not have been previously reported.
The bug must be in the most recently released version.
You must not have created the buggy code or are in anyway involved in the creation of it.

WordPress Bounties

- Remote execution of arbitrary PHP code: US$1000
- Remote malicious file inclusion: US$1000
- Remote SQL Injection that allows reading or modifying the database: US$500
- Persistent cross-site scripting (XSS) (Please note that admin and editor level users are permitted by WordPress to use unfiltered HTML): US$500
- Authentication flaw that allow access to Admin-level capabilities: US$500
- Privilege escalation from Subscriber to Admin or Super Admin: US$500
- Information disclosure that exposes wp-config.php file contents: US$500
- Reflective cross-site scripting (XSS): US$200
- DOM-based cross-site scripting (XSS): US$200
- Cross-site request forgery (CSRF): US$200
- Privilege escalation: US$100

WordPress Plugin Bounties

For plugins with over 1,000,000 downloads and compatible with the most recent WordPress version, according to wordpress.org. The bounties are also available for our plugins Automatic Plugin Updates and No Longer in Directory.

Why am I looked for vulnerabilities?

- Trying this weird bug bounty program

My goals were:

- Found vulnerabilies (at least two) in WordPress plugins with over 500k download!

Cross Site Scripting

- Event Manager, a plugin for organize event
- On 07/07/2014 more than 900k downloads
- Simple XSS in phone field in subscription form
- Reported to:
    - to plugins@wordpress.org
    - to the bug bounty program

Code execution

- WP Photo Album Plus
- On 07/07/2014 more than 1000k downloads
- PHP Code Execution
- Reported to:
  - to plugins@wordpress.org

What did WordPress did?

- Report the XSS to the developper
- About the code execution, they told me that they do not handle with vulnerability in plugin.

What did the bug bounty program did?

- Replied to my email in a first time
- Then no more email from them

WordPress

- Content Management System
- PHP and MySQL
- GNU GPL

WordPress - Users Profiles

- Administrator
- Editor
- Author
- Contributor
- Subscriber

WorPress - About Code execution

Admin can exucte PHP code because it can:

- Upload plugins or themes
- Install plugins or themes from wordpress.org
- Edit plugins or themes code

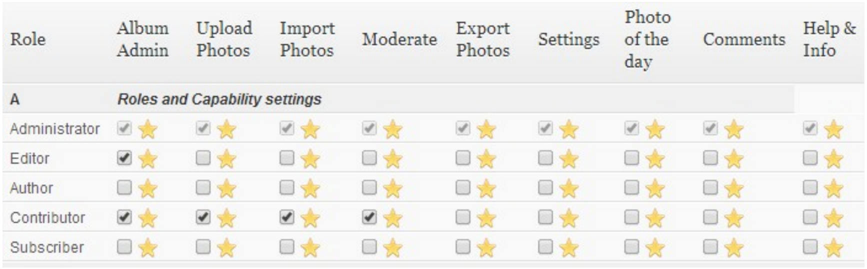
WP Photo Album Plus (WPPA)

- Add photos albums to WordPress
- WordPress plugin
- Gnu GPL v2

WPPA - Users Profiles

- Album Admin
- Upload Photos
- Import Photos
- Moderate
- Export Photos
- Settings
- Photo of the day
- Comments
- Help and Info

About WordPress and WP Photo Album Plus



Role	Album Admin	Upload Photos	Import Photos	Moderate	Export Photos	Settings	Photo of the day	Comments	Help & Info
A	*Roles and Capability settings*								
Administrator	☑ ★	☑ ★	☑ ★	☑ ★	☑ ★	☑ ★	☑ ★	☑ ★	☑ ★
Editor	☑ ★	☐ ★	☐ ★	☐ ★	☐ ★	☐ ★	☐ ★	☐ ★	☐ ★
Author	☐ ★	☐ ★	☐ ★	☐ ★	☐ ★	☐ ★	☐ ★	☐ ★	☐ ★
Contributor	☑ ★	☑ ★	☑ ★	☑ ★	☐ ★	☐ ★	☐ ★	☐ ★	☐ ★
Subscriber	☐ ★	☐ ★	☐ ★	☐ ★	☐ ★	☐ ★	☐ ★	☐ ★	☐ ★

Let's find it!

Let's find it!

Functionality abuse: ZIP Upload

- Upload a zip with photos
- Import photo in an album

How it works? - Demo time (1/4)

First test

- Upload a zip with photos
- Import photo: unzip your file
- Your files are in /wp-content/wppa-depot/username/filename.ext

Let's upload a ZIP file with PHP - Demo time (2/4)

Normal usage

- Upload a zip
- Unzip it
- WPPA deteles non pictures files

Wait?!

- Upload a zip
- Unzip it: here your PHP file is online
- WPPA deletes non pictures files: here your PHP file is delete)

I tawt I taw a puddy cat!

If you PHP file is online, even a few seconds, you can execute it!

Scenario

- Upload a zip with a PHP file
- Run a script trying to access the PHP file
- Unzip it
- The script call the PHP file
- WPPA delete your PHP file

Let's upload a backdoor! - Demo time (3/4)

Let's find it!

I did! I did taw a puddy tat!

Let's find it!

Grep gives the file used for upload zip: wwpa-upload.php. Part used for unzip file can be easily found:

```
if ( isset( $_POST['wppa-upload-zip'] ) )
```

Code begins with a call to wppa_sanitize_files().

```
function wppa_sanitize_files() {
    // Get this users depot directory
    $depot = WPPA_DEPOT_PATH;
    __wppa_sanitize_files($depot);
}
```

Let's find it!

```
function __wppa_sanitize_files($root) {
    // See what's in there
    $allowed_types = array('zip', 'jpg', 'png',
    'gif', 'amf', 'pmf', 'bak', 'log');
    [..]
    if ($files) foreach ($files as $file) {
        if (is_file($file)) {
            $ext = strtolower(substr(strrchr($file, "."), 1));
            if (!in_array($ext, $allowed_types)) {
                unlink($file);
                wppa_error_message(sprintf(__('File %s is of an unsupported filetype and has been
                removed.', 'wppa'), basename($file)));
                $count++;
            }
        }
        elseif (is_dir($file)) {
            $entry = basename($file);
            if ( $entry != '.' && $entry != '..' )
                __wppa_sanitize_files($file);
        }
    }
    return $count;
}
```

Conclusion

- A function called each time the page is reload
- It cleans the user folder

Let's find it!

Let's upload a "persistant" backdoor! - Demo time (4/4)

Let's find it!

Race condition?!

Exploitability depends on network latency

Increase time

- Upload your PHP with a lots of big pictures
- PHP file is available since the server is unziping files

Let's patch it!

Let's patch it!

Let's patch it!

Some correction

- Do not unzip the whole file, just the image
- User an .htaccess file to prevent execution of PHP code
- Unzip in a folder with a random name

Developper correction

- First one: restrict ZIP upload to admin (5.1.13)
- Next one: Do not unzip the whole file, just the image (5.1.14)

Conclusion

- Two vulnerabilities discovered in half a day
- A fake bug bounty program discovered
- For vulnerabilities discovered in WordPress plugin, forget the plugins [at] wordpress.org team

More information

XSS in Event Manager

- Found in version 5.3.3
- Corrected in version 5.3.6.1
- Mention in changelog

Code execution in WP Photo Album Plus

- Found in version 5.0.10
- Corrected in version 5.1.14
- No mention in changelog

Technical security auditor?

- A PhD Student is looking for you
- An interview for its PhD about managing hackers
- Contact me for more details

More information

Questions or lunch time?