16 Reflected XSS on /www/delivery/afr.php (bypass of report #775693)

Share:

TIMELINE

axfla submitted a report to Revive Adserver. Sep 19th (2 years ago)

It is possible to bypass the first fix of this XSS by closing the script tag, and then opening a new one. cURL PoC is trivial :

```
curl "https://revive-instance/www/delivery/afr.php?refresh=10000&</script><script>alert(1)</script>"
```

The response will be :

Code 893 Bytes Wrap lines Copy Download

```
1  <!DOCTYPE html PUBLIC '-//W3C//DTD XHTML 1.0 Transitional//EN' 'http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd'>
2  <html xmlns='http://www.w3.org/1999/xhtml' xml:lang='en' lang='en'>
3  <head>
4  <title>Advertisement</title>
5
6      <script type='text/javascript'><!--// <![CDATA[
7          setTimeout('window.location.replace("https://revive-instance/www/delivery/afr.php?refresh=10000&</script><script>alert(1)</script>&loc=")', 10000000
8      // ]]> --></script><noscript><meta http-equiv='refresh' content='10000;url=https://revive-instance/www/delivery/afr.php?refresh=10000&amp;&lt;/script&gt
9      <style type='text/css'>
10 body {margin:0; height:100%; background-color:transparent; width:100%; text-align:center;}
11 </style>
12 </head>
13 <body>
14
15 </body>
16 </html>
17
18 ## Impact
19
20 An attacker can perform arbitrary actions on behalf of the victim.
```

mbeccati Revive Adserver staff posted a comment. Sep 20th (2 years ago)

Hi axfla, thanks for the report.

I've tried the same URL on Firefox and Chrome and the result was correct as they urlencode the "<>"s. No script execution of any kind. Aside from sending the curl output to a browser manually, how would you think this can be exploitable?

Cheers

mbeccati Revive Adserver staff changed the status to **Needs more info**. Sep 20th (2 years ago)

axfla changed the status to **New**. Sep 20th (2 years ago)

Hi, thank you for your response.

Indeed, < is URL-encoded by the browser, but same goes for " and '. Reading report #775693, I think it was the same (that is the reason why the other researcher provided a cURL PoC as well, payloads were not decoded by Revive).
So yes, to be exploited, it would require a browser that does not encode the query. Inter Explorer versions up to 10 don't URL encode the URL (despite the RFC). IE11 will not encode query parameter's names and values. So those could be used. In IE11, there are built-in XSS filters, but you can't rely on them.

https://trustfoundry.net/browser-url-encoding-decoding-and-xss/ is an interesting read about XSS and URL-encoding.

Cheers

axfla posted a comment. Sep 20th (2 years ago)

Here is an additional resource, a blog post by Troy Hunt in which he illustrates that such vulnerabilities can be exploited in IE9 : https://www.troyhunt.com/browser-url-encoding-and-website/

mbeccati Revive Adserver staff posted a comment. Sep 20th (2 years ago)

It's 2020 and those browsers are hardly relevant. I'm not saying we couldn't do better, but as it is I don't think we can consider this a vulnerability in Revive Adserver.

axfla posted a comment. Updated Sep 20th (2 years ago)

But then why was report #775693 accepted and fixed ? They are the exact same vuln, and it was 2020 too. IE11 is provided with windows 10 as of today.

axfla posted a comment. Sep 20th (2 years ago)

This is a fix bypass, but what you're telling me is you fixed nothing. Accepting the risk could be an understandable decision - despite the fact that I strongly believe it shoud be fixed, since it is an incorrect handling of user input, and you don't know who your users will be, and if they will need compatibility with older browsers -, but the problem is this decision isn't coherent with the previous one.

axfla posted a comment. Updated Sep 20th (2 years ago)

Here it is, chained with an IE XSS filter bypass. If you paste this URL in the latest version of IE, XSS will be triggered. Screenshot attached.

mbeccati Revive Adserver staff changed the status to **Triaged**. Sep 21st (2 years ago)

Fair enough. It's a one character change and the following should do:

```
$jsDest = addcslashes(addcslashes($dest, "\0..\37/\"\\"), "'\\");
```

Could you please double check?

Due to the expected extremely low percentage of the applicable platforms, it is not an high priority fix and it will be included in the next release a few months from now.

xfla posted a comment. Sep 21st (2 years ago)

Port Swigger suggests Unicode-escaping of user inputs in a JS context (https://portswigger.net/web-security/cross-site-scripting/preventing). In my opinion it would make sense to at least escape "<" and ">" as well, I think it would be a good idea in the long run, to make sure it's never possible to open a self-closing HTML tag inside a `<script>`. That way you would be sure to be safe.

That being said, I can't think of a way of exploiting it without "/" right now. Since the next release is scheduled a few months later, I will think about this and let you know if I can find a payload without "/".

mbeccati Revive Adserver staff closed the report and changed the status to **Resolved**. Sep 21st (2 years ago)

I will mark this as resolved for now. When we get closer to the release date we will review this and prepare the disclosure. How would you like to be referenced (username, real name, etc)?

xfla posted a comment. Updated Sep 21st (2 years ago)

My real name, Axel Flamcourt, would be perfect, thanks. Before disclosure, could you please remove/redact the screenshot ? I just realised I didn't redact the host in the above screenshot. I came across this issue in Revive while hunting on another bug bounty program.

mbeccati Revive Adserver staff requested to disclose this report. Jan 19th (2 years ago)

Thanks again for the report. Revive Adserver v5.1.0 has been just released.
The Security Advisory https://www.revive-adserver.com/security/revive-sa-2021-001/ has been published and a CVE-ID will be requested.

xfla agreed to disclose this report. Jan 19th (2 years ago)

Thank you, one more thing I would like to mention about the advisory is that the rXSS worked in the current version of IE as well, ie. IE11. You might want to change this part.

This report has been disclosed. Jan 19th (2 years ago)