Sec Bug #79971 special character is breaking the path in xml function

Submitted: 2020-08-13 13:09 UTC **Modified:** 2021-11-15 07:30 UTC
From: rawataman6525 at gmail dot com **Assigned:** stas (profile)
Status: Closed **Package:** *XML functions
PHP Version: 7.2 **OS:** linux
Private report: No **CVE-ID:** 2021-21707

View | **Add Comment** | **Developer** | **Edit**

[2020-08-13 13:09 UTC] rawataman6525 at gmail dot com

```
Description:
------------
Hi,

I was just playing around with php simplexml function and found an unexpected behaviour in this function.

so to reproduce this
1. create a folder and put php file with this content

<?php

$FILE = "test/";
$path = "/home/aman/".$FILE."poc.xml";
echo simplexml_load_file($path);

?>

2. change /home/aman to your home directory and then change test/ to your folder that you've created in first step

3. Create two file poc.xml and poc-2.xml

now run the above script `php poc.php`

output will be the content of poc.xml file as expected

now change the add `poc-1.xml%00` after test/ in the above script and now as we should get error like this

`simplexml_load_file(): I/O warning : failed to load external entity`

but due to special character we simplexml function only read path befor %00 and after it whatever it is does not
matter

and poc-1.xml file will be loaded successfully

Test script:
---------------
$FILE = "test/";
$path = "/home/aman/".$FILE."poc.xml";
echo simplexml_load_file($path);

Expected result:
----------------
PHP Warning:  simplexml_load_file(): I/O warning : failed to load external entity

Actual result:
--------------
file opened successfully without expected error
```

Patches

Add a Patch

Pull Requests

Add a Pull Request

History

All | Comments | Changes | Git/SVN commits | Related reports

[2020-08-13 14:34 UTC] cmb@php.net

```
-Status: Open
+Status: Verified
-Package: SimpleXML related
+Package: *XML functions
```

[2020-08-13 14:34 UTC] cmb@php.net

```
Thanks for reporting this issue!

new SimpleXMLElement(…, …, true), DOMDocument::load(),
DOMDocument::loadHTMLFile(), XMLReader::open() and maybe others
are affected by this as well.
```

[2020-08-14 05:41 UTC] rawataman6525 at gmail dot com

```
Yes other functions are also seems to be affected
```

[2020-08-21 12:26 UTC] rawataman6525 at gmail dot com

```
any update are you guys working on the fix or not ? please let me know
```

[2020-08-31 21:24 UTC] cmb@php.net

```
-Assigned To:
+Assigned To: stas
```

[2020-08-31 21:24 UTC] cmb@php.net

Actually, the path is silently truncated in xmlParseURI() which may be regarded as bug in libxml2. We could catch that early, though, and bail out:

```
 ext/libxml/libxml.c | 3 +++
 1 file changed, 3 insertions(+)

diff --git a/ext/libxml/libxml.c b/ext/libxml/libxml.c
index c871cb89bd..6e450377f5 100644
--- a/ext/libxml/libxml.c
+++ b/ext/libxml/libxml.c
@@ -308,6 +308,9 @@ static void *php_libxml_streams_IO_open_wrapper(const char *filename, const char
        int isescaped=0;
        xmlURI *uri;

+       if (strstr(filename, "%00")) {
+               return NULL;
+       }

        uri = xmlParseURI(filename);
        if (uri && (uri->scheme == NULL ||
```

Stas, what do you think?

[2020-09-01 05:45 UTC] stas@php.net

I am not entirely convinced it's either PHP problem or security problem, but I guess we could fix that. However, xmlParseURI() decodes URL-encoding, wouldn't that create other issues if you pass unsecured strings to it?

[2020-09-01 05:57 UTC] stas@php.net

Also there's php_libxml_input_buffer_create_filename and php_libxml_output_buffer_create_filename - does it need to be changed too?

[2020-09-01 08:27 UTC] cmb@php.net

I think this is exactly the same issue like non percent-encoded NUL bytes in paths, and so should be treated the same way. The fact that these functions generally accept percent-encoded characters needs to be documented, but I don't see that we can change that (in stable release branches), because existing code may deliberately rely on that.

Anyhow, a full-fledged patch (guarding all xmlParseURI() uses): <https://gist.github.com/cmb69/da6a17dc8604a331f3aa7f2860abaa49>.

[2020-09-01 08:27 UTC] cmb@php.net

```
-PHP Version: 7.3Git-2020-08-13 (snap)
+PHP Version: 7.2
```

[2020-09-02 03:39 UTC] rawataman6525 at gmail dot com

If this patch has been implemented then I would like to request CVE for this please. It would be helpful for my resume/CV

Thank you so much

<https://gist.github.com/cmb69/da6a17dc8604a331f3aa7f2860abaa49>

[2020-09-02 03:42 UTC] stas@php.net

I'm still not sure how decoding %00 is different in principle from decoding %2F, for example. Both can be used to circumvent some user-space security mechanisms, if we worry about the former, shouldn't we also worry about the latter?

[2020-09-02 03:47 UTC] rawataman6525 at gmail dot com

Basically %00 is NULL character and whenever we use NULL character then only the string before %00 will be considered as input.

I've also tried several URL encoding but only %00 was breaking the path of xml file.

[2020-09-02 07:38 UTC] cmb@php.net

Well, we should also worry about the undocumented percent-encoding decoding in general (e.g. URI encoded directory traversal), but I don't think we can do anything against it, but document the issue.

[2020-09-02 07:44 UTC] stas@php.net

It just looks a bit weird. Maybe we should fix the percent-decoding part instead? I don't think anybody really relies on XML functions percent-decoding filenames. Percent in general is very rare in filenames but I think it's a bug if XML functions can't read any filenames that have %XX in them. I'd rather make it not percent-decode filenames instead.

[2020-09-02 09:45 UTC] rawataman6525 at gmail dot com

If their is any other problem due to percentage encoding then You can raise an exception whenever someone uses percentage encoding in XML functions.

What do you think guys?

[2020-09-03 16:27 UTC] rawataman6525 at gmail dot com

so what's the final decision? will you implement the fix of what please let me know.

Thank you

[2020-09-07 12:40 UTC] rawataman6525 at gmail dot com

Hi team,

If you have any update then please let me know.

Thank you

[2020-09-12 17:32 UTC] rawataman6525 at gmail dot com

```
<?php

$EXT = ".xml";
$FILE = "file"; //This may be controlled by user.

$FinalPath = $FILE+EXT /*File name can be controlled by user but extension is defined in php so their is no way to
change extension*/

$path = "/home/aman/".$FILE;
echo simplexml_load_file($path);

?>
```

save this file as test.php and run this script.

```
>> php test.php
```

now you will get the file.xml content as expected.

lets see why I marked it as security issue.

change the $FILE variable in above script from 'file' to '../../etc/passwd%00file'

now lets understand what's going on here so basically we are using directory transversal + null character to successfully reading the files on the server.

I think it's a valid issue and it should be fixed and also assign a CVE.

Thank you

[2020-09-19 02:44 UTC] rawataman6525 at gmail dot com

No update?

[2020-10-15 07:25 UTC] rawataman6525 at gmail dot com

Hi what about fix?

[2020-10-30 07:01 UTC] rawataman6525 at gmail dot com

Hi team, Its my humble request to fix this issue If You find this a valid issue so that I can claim bounty for this issue from hackerone : https://hackerone.com/ibb-php.

I hope you'll understand

Thank you

[2021-09-20 05:52 UTC] stas@php.net

```
-Assigned To: stas
+Assigned To: cmb
```

[2021-09-20 11:31 UTC] cmb@php.net

```
-Assigned To: cmb
+Assigned To: stas
```

[2021-09-20 11:31 UTC] cmb@php.net

Whenever we open a file, we feed the the result of xmlParseURI() into xmlURIUnescapeString() and use this. This way, the URI is already percent decoded. The only issue are %00, because these are converted to NUL bytes, and since libxml2 deals with zero terminated strings, that leads to truncation, so we need to catch this (and only this).

DOMImplementation::createDocumentType() does not percent decode, but I see no issue there, because if that URI is ever used to retrieve the DTD, is would go through are loaders, which do the percent decoding anyway. Still, I think that catching %00 early here is a good thing.

Thus, I propose to merge <https://gist.github.com/cmb69/da6a17dc8604a331f3aa7f2860abaa49>.

[2021-09-20 11:38 UTC] rawataman6525 at gmail dot com

If this issue will be fixed then. Will you assign CVE for this?

[2021-09-21 04:46 UTC] stas@php.net

```
Yeah, that what is stopping me - why we are decoding it at all? simplexml_load_file() isn't supposed to url-decode
filenames. I feel like we're paining over the real bug here.
```

[2021-09-21 04:47 UTC] stas@php.net

```
I mean painting over.
```

[2021-09-21 10:17 UTC] cmb@php.net

```
> simplexml_load_file() isn't supposed to url-decode filenames.

So you are suggesting to not call xmlParseURI() in the first
place, but use the URL as is?  That might be the proper solution,
but I wouldn't want to make this change for any of the stable
branches (at least not PHP 7.4, let alone 7.3).  Maybe just apply
and release the patch, and do a PR for the proper behavior
afterwards?  This could then be discussed publicly.
```

[2021-11-15 07:25 UTC] stas@php.net

```
-CVE-ID:
+CVE-ID: 2021-21707
```

[2021-11-15 07:28 UTC] stas@php.net

```
I guess there's not much harm in applying it, even though it's not the proper solution.
```

[2021-11-15 07:31 UTC] git@php.net

```
Automatic comment on behalf of cmb69 (author) and smalyshev (committer)
Revision: https://github.com/php/php-src/commit/f15f8fc573eb38c3c73e23e0930063a6f6409ed4
Log: Fix #79971: special character is breaking the path in xml function
```

[2021-11-15 07:31 UTC] git@php.net

```
-Status: Verified
+Status: Closed
```

Last updated: Fri Dec 16 13:05:56 2022 UTC