New issue

Jump to bottom

The Recursive call of "lyxml_parse_elem" leads to crash #1453

Closed **zounathan** opened this issue on Mar 8, 2021 · 13 comments

zounathan commented on Mar 8, 2021

No description provided.

michalvasko commented on Mar 8, 2021 Member

Could you provide the data to reproduce? We cannot fix anything without them.

zounathan commented on Mar 8, 2021 • edited by michalvasko Author

```
<?xml version="1.0" encoding="UTF-8"?>
<module name="all-imp"
        xmlns="urn:ietf:params:xml:ns:yang:yin:1"
        xmlns:all_imp="urn:all-imp">
  <yang-version value="1.1"/>
  <namespace uri="urn:all-imp"/>
  <prefix value="all_imp"/>
  <identity name="ident4"/>
  <identity name="ident5">
          <base name="ident4">
                  <base name="1">
                          <base name="2">
                          <base name="2">
                          <base name="2">
                          <base name="2">
                          <base name="2">
                          <base name="2">
                          <base name="2">
                          <base name="2">
                          <base name="2">
                          <base name="2">
                          <base name="2">
                          <base name="2">
                          <base name="2">
                          <base name="2">
                           .....repeat
                          <base name="2"/>
                  </base>
          </base>
  </identity>
</module>
```

michalvasko commented on Mar 8, 2021 • edited Member

That was not very helpful, please provide the **exact** module to reproduce the crash. You can attach it if it is too big.

zounathan commented on Mar 8, 2021 Author

> That was not very helpful, please provide the **exact** module to reproduce the crash. You can attach it if it is too big.

It's the file to be parsed with the code below:

```
ctx = ly_ctx_new(NULL, 0);
lys_parse_path(ctx, file, LYS_IN_YIN);
```

with the repeat of "<base name="2">", recursive calls of function `lyxml_parse_elem ocurrs` . And finally result in a crash.

michalvasko commented on Mar 8, 2021 Member

I need the module exactly, without any `repeat` , what is it supposed to do? I just do not get it at all, is it an infinite repeat or what? Is the crash because of a stack overflow? In that case there is nothing we can do about that.

zounathan commented on Mar 8, 2021 Author

repeat 135728 times of `<base name="2">` . It's too long, I don't paste the whole poc.
In this case, the stack will grow to the unreachable address, which results in the crash.

michalvasko commented on Mar 8, 2021 Member

Um, yes, that definitely sounds like a stack overflow, what exactly are you trying to achieve? You can increase the stack size on your system if you really need to load such a module but I doubt that.

zounathan commented on Mar 8, 2021 Author

No matter how big I increase the stack size, I can also increase the repeat times of `<base name="2">` , and finally result in the crash.
I'm doing test on a device which using this lib. And I find this bug that make device deny of service.

michalvasko added a commit that referenced this issue on Mar 8, 2021

`common FEATURE add a hard limit for recursion` … `298b30e`

michalvasko commented on Mar 8, 2021 Member

Fair enough, now it should be limited.

fredgan commented on May 25, 2021 Contributor

CVE-2021-28903 was assigned to this issue.

mruprich commented on Jul 14, 2021

@michalvasko Hi, this hard limit does not solve very small stack sizes since if you (for any reason) limit your stacksize to some very small value, you can still easily hit the segfault:

1. ulimit -s 512
2. Run the code from comment above with fixed version of libyang
3. Segmentation fault (core dumped)

Not sure about a better solution though(sigh) :(

michalvasko commented on Jul 14, 2021 Member

The only solution I can think of is removing the recursion from this (and other) function calls. I seriously doubt it is worth it and will certainly not be implemented for the old *libyang* v1.

mruprich commented on Jul 14, 2021

Ok, good to know. I just thought it was good to mention this.

Thanks ;)

michalvasko closed this as completed on May 31

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

4 participants