Heap overflow in jerry-core #3976

Closed **Changochen** opened this issue on Jul 5, 2020 · 14 comments

Assignees

Labels bug

Changochen commented on Jul 5, 2020

JerryScript revision

git hash: `392ee71`

Test case

```
( function ( { a =  arguments   }) {
    const arguments
}
)
```

In debug build, it triggers an assertion 'scope_stack_p > context_p->scope_stack_p' failed

Execution steps

./jerry poc.js

Build cmd

python tools/build.py --compile-flag="-fsanitize=address"

Stack dump:

```
=================================================================
==59856==ERROR: AddressSanitizer: global-buffer-overflow on address 0x5581992ba27c at pc 0x558198ef79cd bp 0x7ffffcf5f490 sp 0x7ffffcf5f480
READ of size 2 at 0x5581992ba27c thread T0
    #0 0x558198ef79cc  (/home/yongheng/jerry_clean/build/bin/jerry+0x2d9cc)
    #1 0x558198f263d7  (/home/yongheng/jerry_clean/build/bin/jerry+0x5c3d7)
    #2 0x558198f81326  (/home/yongheng/jerry_clean/build/bin/jerry+0xb7326)
    #3 0x558198f34f6c  (/home/yongheng/jerry_clean/build/bin/jerry+0x6af6c)
    #4 0x558198eff1a3  (/home/yongheng/jerry_clean/build/bin/jerry+0x351a3)
    #5 0x558198f030ca  (/home/yongheng/jerry_clean/build/bin/jerry+0x390ca)
    #6 0x558198f3dbcb  (/home/yongheng/jerry_clean/build/bin/jerry+0x73bcb)
    #7 0x558198f3f19c  (/home/yongheng/jerry_clean/build/bin/jerry+0x7519c)
    #8 0x558198f2f238  (/home/yongheng/jerry_clean/build/bin/jerry+0x65238)
    #9 0x558198f33ac8  (/home/yongheng/jerry_clean/build/bin/jerry+0x69ac8)
    #10 0x558198f3c101  (/home/yongheng/jerry_clean/build/bin/jerry+0x72101)
    #11 0x558198f0484d  (/home/yongheng/jerry_clean/build/bin/jerry+0x3a84d)
    #12 0x558198f0515a  (/home/yongheng/jerry_clean/build/bin/jerry+0x3b15a)
    #13 0x558198f745c0  (/home/yongheng/jerry_clean/build/bin/jerry+0xaa5c0)
    #14 0x558198ee044e  (/home/yongheng/jerry_clean/build/bin/jerry+0x1644e)
    #15 0x7f2eec31db96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
    #16 0x558198ee1219  (/home/yongheng/jerry_clean/build/bin/jerry+0x17219)
```

LaszloLango added the **bug** label on Jul 6, 2020

rerobika assigned **zherczeg** on Jul 9, 2020

Changochen commented on Aug 3, 2020 Author

Hi, seems this issue has been opened for a month. Is there any plan to fix this issue?

zherczeg commented on Aug 4, 2020 Member

Jerry is an open project where anybody can open PRs not just contributors, so feel free to fix it.

rerobika unassigned **zherczeg** on Aug 4, 2020

ossy-szeged commented on Aug 4, 2020 Contributor

> Hi, seems this issue has been opened for a month. Is there any plan to fix this issue?

Thanks for reporting this issue. Of course it would be great to fix this real bug, but it doesn't have the highest priority now. I don't know when it wil be fixed, but I'm sure that the fix should be included in the following release. There is no fixed release date, but we try to release in every 3-4 months and the latest was on 12th June .

dbatyai self-assigned this on Aug 13, 2020

ossy-szeged commented on Sep 29, 2020 Contributor

just a notice, still valid issue today on latest master (`0ffe166`)

NicoleG25 commented on Dec 1, 2020

Hi,
Is there a plan to address this any time soon? :)
Be aware that CVE-2020-24344 was assigned to this issue.

zherczeg commented on Dec 1, 2020 Member

It looks like it is not valid anymore. I think we can close this.

NicoleG25 commented on Dec 1, 2020

Do you happen to know where the fix was applied ? **@zherczeg**

Thanks in advance !

ossy-szeged commented on Dec 1, 2020 Contributor

I bisected, `841d536` was the hash fixed the assertion. Now we get syntaxerror for this poc.js:

```
$ build/bin/jerry poc.js
}
^

SyntaxError: Value assignment is expected after a const declaration. [poc.js:3:1]
```

Otherwise it would be great to add this poc.js to jerry test case.

zherczeg commented on Dec 1, 2020 Member

I am curious about something. I searched the cve entry, and it provided very little information. It says something this bug is a vulnerability, but no example is provided. I thought you need to provide a program which does something, e.g. runs a shell script in jerry-main using this bug (that you cannot normally do). How can somebody be sure it is an actual vulnurability without proving it?

attritionorg commented on Dec 2, 2020

@zherczeg When asking for a CVE ID, you do not need to provide much and there are no standards for submitting evidence of the vulnerability. They assign blindly in most cases and will REJECT the ID later if it is disputed by the vendor and the vendor provides evidence. Not exactly fair to the vendor but that is how the process has been for a long time.

zherczeg commented on Dec 2, 2020 Member

That is interesting. So you can open a cve for every bug in every project in practice? So is it a global issue tracker?

attritionorg commented on Dec 2, 2020

In theory yes, but if MITRE noticed a flood of requests they would likely scrutinize them a bit more and start rejecting the request or ignoring them. As long as it works (or works close enough) they are likely to let it run as is.

ossy-szeged commented on Dec 2, 2020 Contributor

@zherczeg I think we shouldn't trivialize heap buffer overflow issues. All of them should be treated as possible security vulnerability until we can't fix the issue or we can prove if it is a harmless or false bug. We can't expect complete exploit to take an issue seriously.

👍 1

rerobika closed this as completed on Jan 8, 2021

rzr commented on Jun 16, 2021 Contributor

Please also review:
jerryscript-project/iotjs#1973

Assignees

dbatyai

Labels

bug

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

9 participants