

New issue

Jump to bottom

stack-overflow in ecma_regexp_match #3753

Closed owl337 opened this issue on May 18, 2020 · 6 comments

Assignees



Labels

bug

owl337 commented on May 18, 2020 · edited by dbatyai

JerryScript revision

bd1c4df

Build platform

Ubuntu 16.04.6 LTS (Linux 4.15.0-99-generic x86_64)

Build steps

```
python ./tools/build.py --clean --debug --compile-flag=-fsanitize=address --compile-flag=-m32 --compile-flag=-fno-omit-frame-pointer --compile-flag=-fno-common --lto=off --error-message=on --system-allocator=on
```

Test case

```
r = new RegExp ("((?)?)*a");
assert (r.exec("ba")[0] == "a");
```

Output

```
ASAN:SIGSEGV
=====
==103435==ERROR: AddressSanitizer: stack-overflow on address 0xff318fcc (pc 0x0808fc96 bp 0xff319158 sp 0xff318fb0 T0)
#0 0x0808fc95 in ecma_regexp_match /home/jerryscript/jerry-core/ecma/operations/ecma-regexp-object.c:535
#1 0x80915ee in ecma_regexp_match /home/jerryscript/jerry-core/ecma/operations/ecma-regexp-object.c:1060
#2 0x80915ee in ecma_regexp_match /home/jerryscript/jerry-core/ecma/operations/ecma-regexp-object.c:1060
#3 0x8091111 in ecma_regexp_match /home/jerryscript/jerry-core/ecma/operations/ecma-regexp-object.c:995
#4 0x8091b54 in ecma_regexp_match /home/jerryscript/jerry-core/ecma/operations/ecma-regexp-object.c:1122
#5 0x80915ee in ecma_regexp_match /home/jerryscript/jerry-core/ecma/operations/ecma-regexp-object.c:1060
#6 0x8091b54 in ecma_regexp_match /home/jerryscript/jerry-core/ecma/operations/ecma-regexp-object.c:1122
...
#249 0x8091b54 in ecma_regexp_match /home/jerryscript/jerry-core/ecma/operations/ecma-regexp-object.c:1122
#250 0x8091b54 in ecma_regexp_match /home/jerryscript/jerry-core/ecma/operations/ecma-regexp-object.c:1122

SUMMARY: AddressSanitizer: stack-overflow /home/jerryscript/jerry-core/ecma/operations/ecma-regexp-object.c:535 ecma_regexp_match
==103435==ABORTING
```

Credits: This vulnerability is detected by chong from OWL337.

zhczeg commented on May 18, 2020

Member

I think these reports need to be improved. Backtrace needs to be shorter, since it takes hours to scroll down to the bottom. If I want a backtrace I can produce it myself, so it can even be omitted from the report.

In my experiences 99% of these reports are belongs to the following categories:

- infinite JS loops, recursions (fixing them is outside the scope of the engine)
- obsolete asserts (just improve or move the assert)
- trivial unhandled cases

Creating patches for these should take less than 3 minutes, so it would be good to replace the backtrace with an initial patch proposal. Thank you for your efforts.

akosthekiss commented on May 18, 2020

Member

@zhczeg You (we) cannot and must not expect those who are not active developers of the project to understand the code base and submit a PR to fix the issues they find. That's not their task.

zhczeg commented on May 18, 2020

Member

When real people submit issues, they try to be helpful, and often analyze their own issue quite well, which shows their respect to the people who is working on the project. This feels missing from these automated reports. If the reporter would check the source code he sent, the problem would be obvious to him, and we could discuss it. There is no "fix" to infinite recursion in any language, only various ways to abort the execution.

dbatyai commented on May 18, 2020

Member

This pattern should not even result in infinite recursion, it is actually a problem with the current implementation. #3746 will resolve this however.

akosthekiss commented on May 18, 2020

Member

@zhczeg Please, don't do this. This is not even passive-aggressive, but you are actively accusing our community members of being disrespectful and not being helpful. None of which they are.

Some members of the community have time, skills, and/or resources to develop the code base of the project. Some others may not be developers of the project though, but their feedback is still valuable. You simply MUST NOT alienate those who report issues in JerryScript. We have to be thankful for them for their time spent on helping us make JerryScript better by spotting faults we missed.

And the above is also true for automated reports (even if I don't think that the above report was made by an automated system -- only found by one, perhaps). Someone has spent their time to create that automated system and ran it on JerryScript, for our benefit. (BTW, automation is not evil. It helps. CI is also our friend. They all help to focus human resources on the creative part of the development by lifting the burden of repetitive tasks from our shoulders.)


Having said all that, it may still happen that an issue report (be it "hand-made" or automatic) is invalid. Then, you have several options:

- You MAY ignore those issues that irritate you for other maintainers to handle.
- You may explain why you think an issue report is invalid. And you may also *suggest* ways how to avoid them. E.g., how to build JerryScript with stack limit checks enabled.

The bottom line is, please, keep this project a welcoming and safe place for all current and potential members of our community.

@owl337 / chong: Thanks for your report. As @dbatyai mentioned, it is being fixed.

 dbatyai added the **bug** label on May 21, 2020

 rerobika assigned rerobika and dbatyai and unassigned rerobika on May 25, 2020

rerobika commented on May 26, 2020

Member

Resolved via [#3746](#).

 rerobika closed this as completed on May 26, 2020

Assignees

 dbatyai

Labels

bug

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

5 participants

