Code | Issues 182 | Pull requests 54 | Actions | Projects | Security 2 ···

New issue

Jump to bottom

# [bugfix] Fix redos in preprocessRFC2822 regex #6015

Merged

**ichernev** merged 3 commits into `moment:develop` from `vovikhangcdv:fix-redos-in-preprocessRFC2822` on Jul 6

Conversation 15 | Commits 3 | Checks 0 | Files changed 1

**vovikhangcdv** commented on Jun 7 • edited — Contributor

Fixes: #6012

Using the "look-ahead" mechanism regex in `preprocessRFC2822()` to resolve the problem Regular Expression Denial of Service (ReDoS)#6012

**vovikhangcdv** added 2 commits 6 months ago

- `fix redos in preprocessRFC2822 regex` ··· `dc0d180`
- `fix redos using local backtracking regex` ··· ✓ `bfd4f23`

**vovikhangcdv** mentioned this pull request on Jun 7

**[bugfix] fix redos in preprocessRFC2822 regex** #6013

Closed

**ichernev** commented on Jun 10 — Contributor

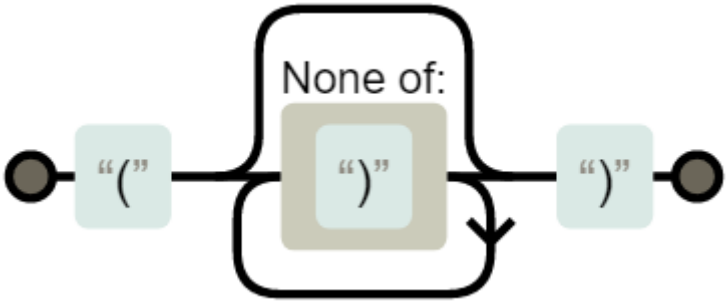
**@vovikhangcdv** can you explain a bit how this is being fixed?

From the linked issue it looks like any open-close relationship that we have to track (not sure if there are more), has the same issue. Also is this compatible all the way back to Roman Empire JavaScript?

**vovikhangcdv** commented on Jun 11 — Contributor Author

> 1. How this is being fixed?

**Firstly, look back to the original regex: `/\([^)]*\)|[\n\t]/g`.**
The hotspot is `\([^)]*\)`. The process would be, sequentially:

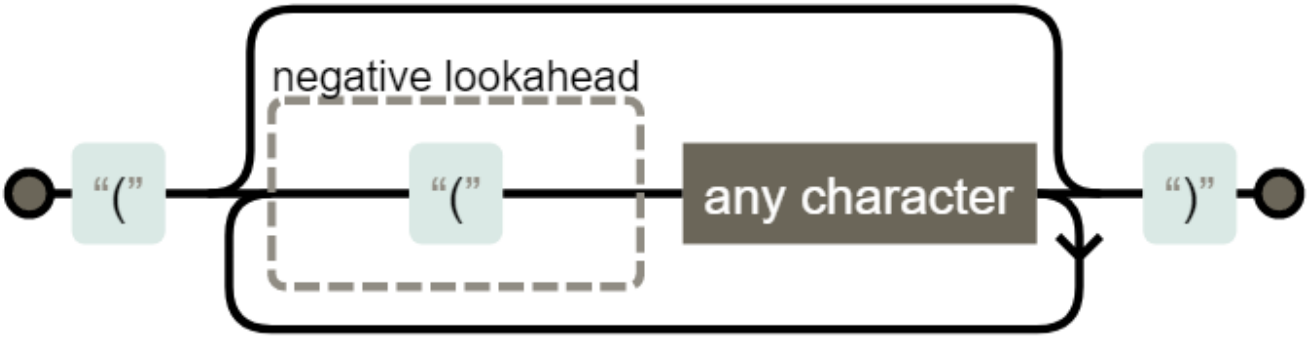



1. Try to match the `(` character.
2. Try to match all those characters that are not `)`, ***as many times as possible (greedy)***.
3. Try to match the `)` character.

This process will be executed for the full input string and repeated with substring (remove the start character) until *match all the three steps* **or** *all characters are checked*.

Consider the evil payload format: `"(".repeat(X)` (the worst-case). This payload will exploit the greedy mechanism of step 2 and the backtracking mechanism of the failure matching process.
For each `(`, step 2 will check all the rest of the substring characters (it costs `X - num_of_checked_substring` substeps) which certainly fails and the process has to backtrack and repeat for total X substrings. **This is a polynomial complexity problem.**

**Now, look to the suggested fix: `/\((?:(?!\().)*\)|[\n\t]/gs`.**
The hotspot (after remove a non-capture group `(?:)`) is: `\(((?!\().)*\)`



**only costs linear steps based on the length of the input string.**

> 2. There are a lot of regular expressions used in this project. Are there any similar issues in this project? How we can find and fix it?

The mission of `moment` is a third-party package that helps other projects and applications achieve their goals. I believe that eliminating all possible security vulnerabilities is crucial work for the community. Give a huge thank you and all the awesome contributors of this project. I see a lot of effort to reduce the risk of ReDos attack vector in the past (#2936, #4163, and other relative commits). After carefully looking at the codebase, this is the only ReDoS vulnerability left I could find. To prevent this kind of vulnerability in the future, we just need to carefully use efficient regexes. The recheck (see the online version) is a useful tool to check the complexity of regex, just make sure that only the safe or linear regexes should be used.

> 3. Is this compatible all the way back to Roman Empire JavaScript?

I am not sure what you asking about. But if you are concerned about the compatibility of the fix, I have tested the fix and confirm that it passed *160,555* test cases of unit tests. Or if I missing something, please let me know, and we will figure it out together.

References:

- Regular expression Denial of Service - ReDoS (OWASP)
- Protect Against Regex Denial Of Service - ReDoS
- Staicu, Cristian-Alexandru, and Michael Pradel. "Freezing the Web: A Study of {ReDoS} Vulnerabilities in {JavaScript-based} Web Servers." 27th USENIX Security Symposium (USENIX Security 18). 2018.

👍 3 ❤️ 5

**emer7** commented on Jun 22

Hi is there update to this fix? Thank you

**vovikhangcdv** commented on Jun 22 Contributor Author

Hi **@ichernev**, could you review the PR?

**ichernev** commented on Jul 5 Contributor

Hey, sorry for the delay, I'll release a build in the coming days with the fix, I was a bit worried about the lookahead (if it was supported), but the current implementation uses whitelist, which is better.

**vovikhangcdv** commented on Jul 5 Contributor Author

> Hey, sorry for the delay, I'll release a build in the coming days with the fix, I was a bit worried about the lookahead (if it was supported), but the current implementation uses whitelist, which is better.

Good to see you back. If you can confirm the issue, could you validate my report on Hunter?. It will help my work too. Thank you, **@ichernev**.

**ichernev** commented on Jul 5 Contributor

How about we change `(/\([^)]*\)|[\n\t]/g` to `(/\([^()]*\)|[\n\t]/g` ? As far as I get it you should avoid matching more open brackets, so we can just prevent that.

**ichernev** commented on Jul 5 Contributor

**@vovikhangcdv**

I can merge the fix with `[^()]` , I can give further attribution to you if necessary (i.e make a security advisory in github), for reporting it. I'll use your excellent description from a few comments above.

`update regex by avoid matching more open brackets` ··· ✓ `4bbb9f3`

**vovikhangcdv** commented on Jul 5 Contributor Author

Hey **@ichernev**,
I can confirm your solution solves the security issue and work well. I have added a commit change from your suggestion.

**vovikhangcdv** commented on Jul 5 • edited Contributor Author

> advisory in github), for reporting it. I'll use your excellent description from a few comments above.

As a researcher, being credited on `moment` security advisory or assigned CVE is my pleasure. I would appreciate that. Thank you **@ichernev**.

**ichernev** commented on Jul 6 Contributor

Reading through the specification of the date format, it looks like formally a comment (stuff in brackets) can have a nested comment inside:

```
FWS             =       ([*WSP CRLF] 1*WSP) /   ; Folding white space
                        obs-FWS

ctext           =       NO-WS-CTL /     ; Non white space controls

                        %d33-39 /       ; The rest of the US-ASCII
                        %d42-91 /       ;  characters not including "(",
                        %d93-126        ;  ")", or "\"

ccontent        =       ctext / quoted-pair / comment

comment         =       "(" *([FWS] ccontent) [FWS] ")"

CFWS            =       *([FWS] comment) (([FWS] comment) / FWS)
```

note how `comment` includes `ccontent` which might include `comment`. Also note that CFWS is part of the `obs-` tokens, short for *obsolete*, so if this was obsolete in 2001 it might be safe to move on :)

To be fair, the Ruby standard library fails to parse dates with comments, so I'm not 100% sure we should deal with this crap. But if we do, there should be code to track open/closed parenthesis, if there is anything unbalanced the parsing should fail (because there are no brackets allowed in the actual string). This code is linear, so won't introduce bottlenecks.

If we keep the current comment approach (disallow open and close paren inside comment), it will "capture" (and remove) only the inner most comment, and the parsing will fail if there are nested comments (which, given the legacy-ness of all of this might be fine :)).

👍 1

**vovikhangcdv** commented on Jul 6 Contributor Author

**ichernev** merged commit `7aebb16` into `moment:develop` on Jul 6

4 checks passed

View details

**ichernev** added a commit that referenced this pull request on Jul 6

`Revert "[bugfix] Fix redos in preprocessRFC2822 regex (#6015)"` ··· `b4e6153`

**ichernev** pushed a commit that referenced this pull request on Jul 6

`[bugfix] Fix redos in preprocessRFC2822 regex (#6015)` ··· `9a3b589`

**vovikhangcdv** deleted the `fix-redos-in-preprocessRFC2822` branch 5 months ago

**ichernev** commented on Jul 6 Contributor

Advisory link: GHSA-wc69-rhjr-hc9g

❤️ 2

**melloware** mentioned this pull request on Jul 7

**Moment.js 2.29.4 CVE-2022-31129** primefaces/primefaces#8968

Closed

**JamieSlome** commented on Jul 8

@ichernev - would it be possible to add the following report URL to the advisory?

**sync-by-unito** bot mentioned this pull request on Jul 8

**Bump moment from 2.29.1 to 2.29.4** apollographql/vscode-graphql#77

Open

Open

**eexit** mentioned this pull request on Jul 12

**Upgrade moment to 2.29.3** TryGhost/Ghost-Storage-Base#66

Open

**adleong** mentioned this pull request on Jul 12

**web: Update moment for CVE-2022-31129** linkerd/linkerd2#8856

Merged

**unlikelyzero** mentioned this pull request on Jul 14

**Update moment** nasa/openmct#5509

Merged

15 tasks

**debricked** bot mentioned this pull request on Jul 18

**Fix CVE–2022–31129** suculent/thinx-device-api#400

Merged

**jhuckaby** added a commit to jhuckaby/Cronicle that referenced this pull request on Jul 29

`Bumped moment to v2.29.4 for vuln` ··· `b939bf4`

**github-actions** bot mentioned this pull request on Aug 1

**CVE-2022-31129 - high detected in moment['2.19.3']** rhicksiii91/goof#279

Open

**stondino00** mentioned this pull request on Aug 7

**Security Dependency Update Moment.js to 2.29.4 in Nextcloud 24.0.4** nextcloud/server#33478

Open

**debricked** bot mentioned this pull request on Aug 8

**Fix CVE–2022–31129** OS2iot/OS2IoT-frontend#113

Closed

**nicoloboschi** mentioned this pull request on Aug 30

**[security] Upgrade moment.js to 2.29.4** datastax/pulsar-admin-console#81

Merged

**michaeljmarshall** pushed a commit to datastax/pulsar-admin-console that referenced this pull request on Aug 30

`[security] Upgrade moment.js to 2.29.4 (#81)` ··· 4e6b3f4

**craigfay** mentioned this pull request on Sep 6

**[CF] (Security) Upgrade "rollup-plugin-license" to resolve vulnerability in MomentJS** Econify/moonshine-css#29

Merged

**jorritfolmer** mentioned this pull request on Oct 21

**Splunkbase blocking issues for cloud vetting v4.1.0** jorritfolmer/TA-dmarc#45

Open

Reviewers

No reviews

Assignees

No one assigned

Labels

None yet

Projects

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

Regular Expression Denial of Service (ReDoS)

4 participants