


# Unbounded parameter parsing in `Rack::QueryParser` can lead to memory exhaustion

High ioquatix published GHSA-gjh7-p2fx-99vx 18 hours ago

Package	Affected versions	Patched versions
 rack (RubyGems)	< 2.2.14	2.2.14
	>= 3.0, < 3.0.16	3.0.16
	>= 3.1, < 3.1.14	3.1.14

Severity

High 7.5 / 10

CVSS v3 base metrics

Attack vector	Network
Attack complexity	Low
Privileges required	None
User interaction	None
Scope	Unchanged
Confidentiality	None
Integrity	None
Availability	High
<a href="#">Learn more about base metrics</a>	

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

CVE ID

CVE-2025-46727

Weaknesses

- CWE-400
- CWE-770

Credits

-  TaiPhung217 Reporter
-  jeremyevans Remediation developer
-  ioquatix Coordinator

Description

## Summary

Rack::QueryParser parses query strings and application/x-www-form-urlencoded bodies into Ruby data structures without imposing any limit on the number of parameters, allowing attackers to send requests with extremely large numbers of parameters.

## Details

The vulnerability arises because Rack::QueryParser iterates over each &-separated key-value pair and adds it to a Hash without enforcing an upper bound on the total number of parameters. This allows an attacker to send a single request containing hundreds of thousands (or more) of parameters, which consumes excessive memory and CPU during parsing.

## Impact

An attacker can trigger denial of service by sending specifically crafted HTTP requests, which can cause memory exhaustion or pin CPU resources, stalling or crashing the Rack server. This results in full service disruption until the affected worker is restarted.

## Mitigation

- Update to a version of Rack that limits the number of parameters parsed, or

- Use middleware to enforce a maximum query string size or parameter count, or
- Employ a reverse proxy (such as Nginx) to limit request sizes and reject oversized query strings or bodies.

Limiting request body sizes and query string lengths at the web server or CDN level is an effective mitigation.