Unbounded Recursion Stack Overflow in markdown

ADACORE SECURITY ADVISORY

Document Id: SEC.MARKDOWN-0009

Version 2 - Jul 10, 2026

AdaCore

Title	Unbounded Recursion Stack Overflow in markdown	
Status	Final	
Author	Thomas Serabian	
Reviewed by	Frédéric Léger	

Revision History

Version	Date	Comments
1	Jun 30, 2026	Initial description
2	July 10, 2026	Add fix build date

Contents

1. Preface

1.1. Scope

This document is an advisory describing the security impact of `MARKDOWN-0009`. The issue is tracked under the ticket number `MARKDOWN-0009` in AdaCore's issue tracking database.

This document also presents possible workarounds and mitigation for the issue.

1.2. Distribution

This advisory is made available in confidence to AdaCore customers under embargo until 2026-09-08 so that they can address the issue it describes before public availability.

Thereafter, it will be available to the general public under the terms of the `CC BY-ND 4.0` license.

1.3. Contact

For questions on this document, please contact AdaCore support at product-security@adacore.com or using the standard reporting procedures if you are an AdaCore customer.

2. Vulnerability

2.1. Affected Products

The vulnerability described in this document was reported for the following product versions:

- *Markdown* - `markdown` *26.0*

The vulnerability described in this document applies to the following product versions:

- `markdown` < 27.0

2.2. Severity and Impact

CVSS v3.1 score: 5.5 (CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:N/I:N/A:H)

This vulnerability allows a Denial of Service attack via CPU exhaustion or process crash on Markdown if the attacker submits a crafted Markdown document containing deeply nested block containers to an application.

A CVE (*Common Vulnerabilities and Exposures*) has been created for that case, and will be referred as `CAN-2026-2032496` (until MITRE delivers a real CVE ID).

2.3. Detailed Description

Thanks to the investigation work done by Jon Hood, a vulnerability on *Markdown* has been discovered.

The MarkDown block parser places no limit on how deeply block containers (blockquotes >, list items -) may nest. Attacker-controlled Markdown therefore drives two distinct failures from one input:

- Quadratic-time CPU exhaustion (CWE-407). In `Markdown.Parsers.Parse_Line` (source/parser/markdown-parsers.adb:211-217), each nesting marker re-runs the full block-detector battery at an ever-deeper cursor, so a single line of n nested markers parses in O(n2).
- Unbounded recursion -> stack overflow (CWE-674). Document -> Complete_Parsing (source/parser/implementation/markdown-implementation.adb:26) and the controlled teardown Unreference (:139) recurse one stack frame per nesting level (~200 bytes/frame). On a typical worker-thread stack this crashes the process.

3. Solution

3.1. Workarounds

No workaround is available

3.2. Correction

The vulnerability described in this document is fixed in the following product versions:

- *Markdown* - `markdown` >= 26.2

Starting from build date `20260707`, the *Markdown* component is fixed on `wavefront`.

4. Appendix

4.1. CVSS Score Justification

Metric	Justification
AV:L	The vulnerable Markdown parser processes local files or requires the malicious file to be present locally on the system.
AC:L	The attacker only needs to submit a crafted Markdown payload to an application utilizing the parser.
PR:N	There is no privilege needed.
UI:R	User interaction is required, as a user must open or initiate the processing of the malicious Markdown file.
S:U	The exploited vulnerability only impacts the application processing the Markdown file, meaning the vulnerable and impacted components remain within the same security authority.
C:N	The vulnerability does not allow access to any unauthorized information, resulting in no impact on confidentiality.
I:N	The vulnerability does not allow modification of application data or system files, resulting in no impact on integrity.
A:H	Deeply nested block containers cause quadratic-time CPU exhaustion and a stack overflow process crash, leading to a complete denial of service.