Potential memory corruption issue with GNAT.Regexp

ADACORE SECURITY ADVISORY

Document Id: SEC.UA16-001

Version 1 - Feb 02, 2023

AdaCore

Title	Potential memory corruption issue with GNAT.Regexp	
Status	Final	
Author	Paul Butcher	
Reviewed by	Alexander Senier, Olivier Ramonat	

Revision History

Version	**Date**	**Comments**
1	Feb 02, 2023	initial version

Contents

1. Preface

1.1. Scope

This document is an advisory describing a potential memory corruption issue when using the GNAT.Regexp library and the compilation of specific patterns. The issue could be potentially triggered by a malicious entity by supplying specially crafted regular expressions into an application that utilizes the GNAT.Regexp library to dynamically compile the expression.

The issue is tracked under the ticket numbers UB02-010 and UA16-001 in AdaCore's issue tracking database. This document also presents possible workarounds and mitigations for the issue.

1.2. Distribution

This advisory is made available in confidence to AdaCore customers under embargo until 2023-05-01 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:

- GNAT Pro 21.1

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

- GNAT Pro < 22.0

2.2. Severity and Impact

CVSS v3.1 score: 6.1 (medium) (AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L/E:U/RL:O/RC:R)

Without mitigation this issue could be used to corrupt memory within an application and invoke unknown behavior. The actual fault within the implementation of the library results in a buffer being accessed with an out of range index value. This behavior may result in the termination of the application due to a condition such as a segmentation fault. In addition, the application could continue to execute but would now be operating within an unknown state. The impact of this vulnerability will therefore be application dependent and determined by the actual instruction calls within the application code following the out of range buffer read.

An exploitation of this vulnerability could result in a denial of service attack (DoS attack) or the potential circumvention of security controls by transitioning the application into an unknown operating state. This is application specific and depends on the usage of the GNAT.Regex library by the application code.

2.3. Detailed Description

GNAT.Regexp can be used to search for regular expressions. The regular expression String first needs to be compiled into a finite state automaton. This ensures an efficient pattern matching algorithm through the GNAT.Regexp.Match function.

Prior to GNAT Pro 22.0 the library was built without the runtime constraint check “index check”. If an invalid regular expression pattern, that met the following criteria was passed into the GNAT.Regexp.Compile function it will result in an out of range index being used to read a value from a buffer:

- a missing closing bracket at the end of a pattern like “[a-b”, “[a”, or “[]”
- an extra ‘|’ character as the final character

These faults are covered under the following recorded Known Problems (KPs) in GNAT version 21:

- KP-UB02-010 Missing regular expression syntax-error check
- KP-UA16-001 Missing regular expression syntax-error check

3. Solution

3.1. Workarounds

This issue can be mitigated in the following way:

- Ensuring that any regular expression pattern passed into GNAT.Regexp.Compile does not include the invalid format described above.

3.2. Correction

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

- GNAT Pro >= 22.0

The solution implemented within GNAT Pro 22.0 is two fold:

- The invalid regular expression pattern strings will now be rejected by GNAT.Regex.Compile
- The library is now compiled with the runtime constraint check: “index check”

4. Appendix

4.1. CVSS Score Justification

Metric	Justification
AV:N	Depends on how the library is being used. Assuming that regexp can be provided over the network.
AC:H	While causing a crash only requires a specially crafted regular expression, impacting the integrity and confidentiality of an application without causing a crash requires good knowledge of the application memory layout.
PR:N	Assuming a network application, the only privilege required is to pass a regular expression to GNAT.Regexp for complication.
UI:N	When passed a regular expression, GNAT.Regexp will compile it and trigger the issue without user interaction.
S:U	Assuming that only the address space of the application using GNAT.Regexp is affected.
C:L	Impact depends on use case.
I:L	Impact depends on use case.
A:L	Impact depends on use case.
E:U	No exploit has been created.
RL:O	Official fix available.
RC:R	Availability impact can be reproduced, but confidentiality / integrity depends on use case