

# Some dereference failures found during software vulnerabilities investigation on Redis

Edit advisory

⊗ Closed

Moderate ) janislley opened GHSA-rw4g-qm64-8mc6 on Jul 10 · 4 comments

Package

No package listed

Affected versions

Patched versions

7.0.11

None

janislley opened on Jul 10 • edited -

## Description

Hello.

I am performing some tests on redis as a security investigative report.

During the tests, potential software vulnerabilities were found.

To identify this kind of vulnerabilities it was used the tool ESBMC-WR: https://github.com/janislley/esbmc-wr

More about the tool: https://arxiv.org/pdf/2102.02368.pdf

Tests were performed in the latest redis version.

Please let me know if you need more reports or information regarding the tests.

Check the logs of the verification:

# Issue 01: dereference failure: array bounds violated

[FILE] utils/corrupt\_rdb.c

[ARGS] ['--unwind', '1', '--no-unwinding-assertions']

[FUNCTION] main

State 3 file corrupt\_rdb.c line 24 function main thread 0

Violated property:

file corrupt\_rdb.c line 24 function main dereference failure: array bounds violated

**VERIFICATION FAILED** 

## Issue 02: dereference failure: invalid pointer

[FILE] src/localtime.c

[ARGS] ['--unwind', '1', '--no-unwinding-assertions']

[FUNCTION] nolocks\_localtime

State 1 file localtime.c line 64 function nolocks localtime thread 0

State 2 file localtime.c line 65 function nolocks\_localtime thread 0

State 3 file localtime.c line 66 function nolocks\_localtime thread 0 days = -4024454547492 (11111111 11111111 11111100 01010110 11111011 11010000 11110111 11011100)

State 4 file localtime.c line 69 function nolocks localtime thread 0

Violated property:

file localtime.c line 69 function nolocks\_localtime

dereference failure: invalid pointer

**VERIFICATION FAILED** 

# Issue 03: dereference failure: invalid pointer

[FILE] src/setproctitle.c

 $[\mathsf{ARGS}] \ [\text{'--unwind'}, \ '1', \ '\text{--no-unwinding-assertions'}]$ 

[FUNCTION] spt\_copyenv

State 1 file setproctitle.c line 118 function spt\_copyenv thread 0 envsize = 32768 (00000000 00000000 10000000 00000000)

State 2 file setproctitle.c line 119 function spt\_copyenv thread 0 envcopy = &dynamic\_1\_array[0]

State 6 file string.c line 264 function memcpy thread 0

Violated property:

file string.c line 264 function memcpy dereference failure: invalid pointer

**VERIFICATION FAILED** 

- A significant and a significan
- 🔀 🚳 janislley was credited as a reporter on Jul 10
- 🔀 🤹 janislley accepted credit on Jul 10

Decline credit

janislley changed the title Software vulnerabilities investigation on Redis Some dereference failures found during software vulnerabilities investigation on Redis on Jul 11

yossigo commented on Jul 11

<u>@janislley</u> Thanks for approaching us. We've looked at the code this points to but failed to spot any issues. We may also not fully understand the structure of this report. Did you manually verify this report is accurate, and can you provide more specific details about the issues you believe you've found?



janislley commented on Jul 11 • edited -

#### Hi @yossigo

This is an automated verification to exploit potential vulnerabilities;

I'll go ahead and explain some points about the issues found.

 $\label{lssue 02 - in line 69, look at tmp->tm_isdst = dst} \ \ \text{it seems to be an invalid pointer reference for mp->tm_isdst} \ \ \text{value}.$ 

The static verification tool is suggesting that the pointer tmp could potentially be invalid. There are several reasons why a pointer could be invalid:

• If tmp has been declared but not initialized (i.e., it does not point to a valid struct tm), then dereferencing it would also lead to undefined behavior.

• If tmp points to an array of struct tm objects and you're trying to access an element outside of the array bounds, this is also undefined behavior.

Other ones, seem to be a false positive.



#### oranagra commented on Jul 11

@janislley look at the code, there are two places calling that function and they both pass a reference to a struct on the stack. there's no way for this variable (argument) to be uninitialized or pointing to invalid memory.







#### janislley commented on Jul 11

#### Got it @oranagra .

I will close this investigation case with your explanation.

Thank you!



#### Severity

Moderate 5.7 / 10

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

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

## CVE ID

No known CVE

#### Weaknesses

CWE-119 CWE-822

# Credits

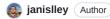




# Collaborators

Only the following users and teams can see and collaborate on this advisory:







# **Publishers**

Only the following users and teams can publish this advisory:

