sungjungk Update README.md History

1 contributor

27 lines (21 sloc) | 1.95 KB

Description

A memory allocation failure was discovered in the parse_report function in src/whoopsie.c in whoopsie 0.2.69. The vulnerability causes a big memory allocation, which may lead to remote denial of service in the g_malloc and g_realloc functions in glib/gmem.c.

Background

What is the whoopsie?

whoopsie is the "Ubuntu Error Reporting" daemon, and is installed by default in both desktop/server installations. When something crashes, whoopsie does two things: collects the crash report generated by Apport and can send them to Ubuntu/Canonical (specifically to https://daisy.ubuntu.com)

Basic operation

When a program has been crashed, Linux system tries to create a '.crash' file on '/var/crash/' directory with python script located in '/usr/share/apport/apport'. The file contains a series of system crash information including core dump, syslog, stack trace, memory map info, etc. After then, whoopsie parses key-value pairs in '.crash' file and encodes it into binary json (bson) format. Lastly, whoopsie forwards the data to a remotely connected Ubuntu error report system.

Details

In whoopsie 0.2.69 and earlier, there is a denial of service vulnerability in the parse_report function. A crafted input, i.e., crash report located in '/var/crash/', will lead to a denial of service attack. During the parsing of the crash report, the data length is not checked. The value of data length can be directly controlled by an input file. In the parse_report() function, the g_malloc or g_realloc is called based on data length. If we set the value of data length close to the amount of system memory, it will cause the daemon process to terminate unexpectedly(i.e., application crash), hang the system, or trigger the OOM killer.

How to run

You require the following modules to run whoopsie_killer.py:

- argparse

Demo video

- Let's check whoopsie_killer2.poc