Bug 7975 (CVE-2024-23307) - [ANCK 5.10][CVE-2024-23307] potentially integer overflow in raid5_cache_count

Status: RESOLVED FIXED
Alias: CVE-2024-23307
Product: ANCK 5.10 Dev
Component: block/storage (show other bugs) block/storage
Version: 5.10.y-16
Hardware: All Linux
Importance: P3-Medium S3-normal
Target Milestone: ---
Assignee: Joseph Qi
QA Contact: shuming
URL:
Whiteboard:
Keywords:
Duplicates (1): ~~8098~~ (view as bug list)
Depends on:
Blocks:

Reported: 2024-01-19 10:49 UTC by Shiloong
Modified: 2024-03-15 09:38 UTC (History)
CC List: 4 users (show)
See Also:

Attachments
Add an attachment (proposed patch, testcase, etc.)

Note
You need to log in before you can comment on or make changes to this bug.

Shiloong **2024-01-19 10:49:57 UTC** Description

on behalf of Gui-Dong Han <2045gemini@gmail.com>:

In raid5_cache_count():
```
    if (conf->max_nr_stripes < conf->min_nr_stripes)
        return 0;
    return conf->max_nr_stripes - conf->min_nr_stripes;
```
The current check is ineffective, as the values could change immediately
after being checked.

In raid5_set_cache_size():
```
    ...
    conf->min_nr_stripes = size;
    ...
    while (size > conf->max_nr_stripes)
        conf->min_nr_stripes = conf->max_nr_stripes;
    ...
```

Due to intermediate value updates in raid5_set_cache_size(), concurrent
execution of raid5_cache_count() and raid5_set_cache_size() may lead to
inconsistent reads of conf->max_nr_stripes and conf->min_nr_stripes.

The current checks are ineffective as values could change immediately
after being checked, raising the risk of conf->min_nr_stripes exceeding
conf->max_nr_stripes and potentially causing an integer overflow.

This possible bug is found by an experimental static analysis tool
developed by our team. This tool analyzes the locking APIs to extract
function pairs that can be concurrently executed, and then analyzes the
instructions in the paired functions to identify possible concurrency bugs
including data races and atomicity violations. The above possible bug is
reported when our tool analyzes the source code of Linux 6.2.

To resolve this issue, it is suggested to introduce local variables
'min_stripes' and 'max_stripes' in raid5_cache_count() to ensure the
values remain stable throughout the check. Adding locks in
raid5_cache_count() fails to resolve atomicity violations, as
raid5_set_cache_size() may hold intermediate values of
conf->min_nr_stripes while unlocked. With this patch applied, our tool no
longer reports the bug, with the kernel configuration allyesconfig for
x86_64. Due to the lack of associated hardware, we cannot test the patch
in runtime testing, and just verify it according to the code logic.

https://lore.kernel.org/linux-raid/20240112071017.16313-1-2045gemini@gmail.com/#r

Shiloong **2024-01-19 10:51:12 UTC** Comment 1

This bug has been reported to upstream:
https://lore.kernel.org/linux-raid/20240112071017.16313-1-2045gemini@gmail.com/#r

it has been fixed by following commit:
https://patchwork.kernel.org/project/linux-raid/patch/20240112071017.16313-1-2045gemini@gmail.com/

Joseph Qi **2024-01-31 10:24:45 UTC** Comment 2

(In reply to Shiloong from comment #1)
> This bug has been reported to upstream:
> https://lore.kernel.org/linux-raid/20240112071017.16313-1-2045gemini@gmail.
> com/#r
>
> it has been fixed by following commit:
> https://patchwork.kernel.org/project/linux-raid/patch/20240112071017.16313-1-
> 2045gemini@gmail.com/

Seems it is not merged into mainline yet. So I suggest we wait until it is in
mainline.

Shiloong **2024-01-31 11:34:54 UTC** Comment 3

*** ~~Bug 8098~~ has been marked as a duplicate of this bug. ***

小龙 **2024-03-14 11:55:44 UTC** Comment 4

The PR Link: https://gitee.com/anolis/cloud-kernel/pulls/2884

Comment 5

Joseph Qi **2024-03-15 09:38:15 UTC**

(In reply to 小龙 from comment #4)
> The PR Link: https://gitee.com/anolis/cloud-kernel/pulls/2884

merged

Format For Printing - XML - Clone This Bug - Top of page