New issue Jump to bottom

CVE-2020-8444: analysisd: OS_ReadMSG heap use-after-free with ossec-alert msgs. #1817

Open **cpu** opened this issue on Jan 15, 2020 · 4 comments

cpu commented on Jan 15, 2020 Contributor

The `ossec-analysisd`'s `OS_ReadMSG` function calls `OS_CleanMSG` at the start of processing a received message from the ossec queue UNIX domain socket.

In `src/analysisd/cleanevent.c` the `OS_CleanMSG` function populates the `lf` struct, setting fields like `log`, `hostname` and `program_name` to substrings of the `lf->full_log` buffer.

After cleaning any messages that meet the ossec-alert decoder's criteria are given to that decoder for further processing.

After processing an ossec alert msg from a client the ossec alert decoder will free the `lf->full_log` pointer at the end of its processing, replacing it with a new pointer and populating `lf->generated_rule`:

ossec-hids/src/analysisd/decoders/plugins/ossecalert_decoder.c
Lines 184 to 191 in abb36d4

```
    free(lf->full_log);
    lf->full_log = NULL;
    os_strdup(tmpstr_buffer, lf->full_log);
    lf->log = lf->full_log;


    /* Rule that generated. */
    lf->generated_rule = rule_pointer;
```

Though the `OSSECAlert_Decoder_Exec` function returns `NULL` and not `1` further rule processing of the `lf` struct occurs during `OS_ReadMSG` because of the `lf->generated_rule` set by the decoder before freeing `lf->full_log`.

ossec-hids/src/analysisd/analysisd.c
Lines 872 to 879 in abb36d4

```
    if (lf->decoder_info->type == OSSEC_ALERT) {
        if (!lf->generated_rule) {
            goto CLMEM;
        }

        /* Process the alert */
        currently_rule = lf->generated_rule;
    }
```

If any subsequent processing associated with the generated rule accesses the `lf->hostname` or `lf->program_name` fields set by `OS_CleanMSG` they will be accessing memory of a freed heap chunk previously containing the `lf->full_log`.

I believe the bug was introduced in `fcca013` on July 23, 2008 and affects OSSEC v2.7+.

This is triggerable via an authenticated client through the `ossec-remoted`. The client needs only write a ossecalert message that will have the `program_name` or `hostname` set during `OS_CleanMSG`.

I don't have a strong sense for the possibility of exploitation. I suspect this may be turned into an out of bounds read of heap memory accessing `program_name` or `hostname` during rule processing if the area pointed to after the syscheck decoder free isn't null terminated.

One possible fix would be for the ossecalert decoder to `os_strdup` the `lf->hostname` and `lf->program_name` before freeing `full_log`.

cpu mentioned this issue on Jan 15, 2020

OSSEC-HIDS Security Audit Findings #1821
Closed

cpu commented on Jan 16, 2020 Contributor Author

> One possible fix would be for the ossecalert decoder to os_strdup the lf->hostname and lf->program_name before freeing full_log.

thinking about this more: I think this proposed fix would introduce a memory leak. Using `os_strdup` will mean that the `hostname` and `program_name` pointers no longer point into the `lf->log` or `lf->full_log` buffer and will instead point to newly allocated memory.

The `Free_Eventinfo` function seems to be written with the assumption that freeing `lf->full_log` will free the `program_name` and `hostname`. Unlike other fields of the `Eventinfo` struct these two fields are not explicitly freed in the body of `Free_Eventinfo`.

This is likely a case where someone more familiar with this codebase will have to suggest a better fix.

cpu mentioned this issue on Jan 16, 2020

CVE-2020-8447: analysisd: OS_ReadMSG heap use-after-free decoding syscheck msgs. #1818
Open

cpu changed the title ~~analysisd: OS_ReadMSG heap use-after-free with ossec-alert msgs.~~ CVE-2020-8444: analysisd: OS_ReadMSG heap use-after-free with ossec-alert msgs. on Jan 30, 2020

cpu commented on Jan 30, 2020 Contributor Author

This was assigned CVE-2020-8444

NicoleG25 commented on Nov 30, 2020

Is there any progress on this or some sort of ETA ?
Thanks in advance :)
@atomicturtle

attritionorg commented on Mar 5, 2021

Bump as well. I don't see a fixing commit for this one but not sure if I overlooked something.

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

3 participants