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

Open cpu opened this issue on Jan 15, 2020 · 2 comments

cpu commented on Jan 15, 2020 Contributor

The `ossec-analysisd`'s `OS_ReadMSG` function calls `OS_CleanMSG` at the start of processing a message read 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 syscheck messages are given to the syscheck decoder for further processing.

After processing a syscheck msg from a client the syscheck decoder will free the `lf->full_log` pointer in two places. One place is if the message indicated a change in an existing tracked file:

ossec-hids/src/analysisd/decoders/syscheck.c
Lines 572 to 576 in 60cf8d2

```
572     /* Create a new log message */
573     free(lf->full_log);
574     os_strdup(sdb.comment, lf->full_log);
575     lf->log = lf->full_log;
576     lf->data = NULL;
```

Another place is if the message indicated a new file to track:

ossec-hids/src/analysisd/decoders/syscheck.c
Lines 601 to 604 in 60cf8d2

```
601     /* Create a new log message */
602     free(lf->full_log);
603     os_strdup(sdb.comment, lf->full_log);
604     lf->log = lf->full_log;
```

In both cases the syscheck decoder replaces the existing `lf->log` and `lf->full_log` pointers with pointers to new messages after first freeing the old `lf->full_log`. Afterwards the `DB_Search`, and `DecodeSyscheck` functions return 1 to `OS_ReadMSG`.

Since the decoder returned 1, the `OS_ReadMSG` function will continue processing the event, it will not jump to `CLMEM`:

ossec-hids/src/analysisd/analysisd.c
Lines 762 to 765 in 60cf8d2

```
762     if (!DecodeSyscheck(lf)) {
763         /* We don't process syscheck events further */
764         goto CLMEM;
765     }
```

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

I believe the bug was introduced in `8672fa0` on Nov 18, 2006 and affects OSSEC v2.7+.

This code path is triggerable via an authenticated client through the `ossec-remoted`. The client needs only write a valid syscheck 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 syscheck 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 syscheck decoder to os_strdup the lf->hostname and lf->program_name before freeing full_log.

See my comment on the related bug. I believe this fix would cause a memory leak and shouldn't be implemented as described: #1817 (comment)

cpu changed the title ~~analysisd: OS_ReadMSG heap use-after-free decoding syscheck msgs.~~ CVE-2020-8447: analysisd: OS_ReadMSG heap use-after-free decoding syscheck msgs. on Jan 30, 2020

cpu commented on Jan 30, 2020 Contributor Author

This was assigned CVE-2020-8447

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

1 participant