

[New issue](#)[Jump to bottom](#)

## CVE-2020-8443: analysisd: OS\_CleanMSG off-by-one heap overflow cleaning syslog msgs. #1816

[Closed](#) cpu opened this issue on Jan 15, 2020 · 1 comment · Fixed by [#1824](#)

cpu commented on Jan 15, 2020 • edited

[Contributor](#)

In `src/analysisd/cleanevent.c` the `OS_CleanMSG` function performs pattern matching and if applicable, tries to decode syslog messages to populate some `lf` structure fields according to the syslog data.

[ossec-hids/src/analysisd/cleanevent.c](#)  
Lines 88 to 169 in `abb36d4`

```
88      /* Check for the syslog date format
89      * ( ex: Dec 29 10:00:01
90      *   or 2015-04-16 21:51:02,805 for proftpd 1.3.5
91      *   or 2007-06-14T15:48:55-04:00 for syslog-ng isodate
92      *   or 2007-06-14T15:48:55.3352-04:00 for syslog-ng isodate with up to 6 optional fraction of a second
93      *   or 2009-05-22T09:36:46.214994-07:00 for rsyslog
94      *   or 2015 Dec 29 10:00:01 )
95      */
96      if (
97          ( /* ex: Dec 29 10:00:01 */
98            (loglen > 17) &&
99            (pieces[3] == ' ') &&
```

When a message contains leading text matching the patterns expected for syslog, and contains a substring like `"[ID xx facility.severity]"` in the correct location `OS_CleanMSG` will attempt to remove it by advancing the `lf->log` pointer beyond the end of the substring:

[ossec-hids/src/analysisd/cleanevent.c](#)  
Lines 327 to 345 in `abb36d4`

```
327      /* Remove [ID xx facility.severity] */
328      if (pieces) {
329          /* Set log after program name */
330          lf->log = pieces;
331
332          if ((pieces[0] == '[') &&
333              (pieces[1] == 'I') &&
334              (pieces[2] == 'D') &&
335              (pieces[3] == ' ')) {
336              pieces += 4;
337
338              /* Going after the ] */
```

The code is careful about checking the result from `strchr` when advancing to the expected closing `]`, however it makes an assumption that there must be a non-null character following the `]` when it subsequently advances the `pieces` pointer by 2:

[ossec-hids/src/analysisd/cleanevent.c](#)  
Line 341 in `abb36d4`

```
341      pieces += 2;
```

If a message like `"Oct 31 00:00:00 0 sshd: [ID 0 auth.notice]"` is processed `OS_CleanMSG` will advance beyond the terminating null byte of `lf->log`, resulting in a heap overflow when operating on the `lf->log` pointer subsequently during decoding.

This code was introduced in [8672fa0](#) on Nov 18, 2006. I believe it affects [OSSEC 2.7+](#).

This is triggerable via an authenticated client through the `ossec-remoted`. The client needs only write a message to the remote server of any queue type that will match the expected syslog format and authority substring, but end immediately after the `]`.

I think the best fix is to change the `pieces` pointer to be incremented by 1 instead of 2, or to update the `strchr` check for `"["` instead of just `"["` (edit: implemented in [#1824](#)).

[This was referenced on Jan 15, 2020](#)

## OSSEC-HIDS Security Audit Findings #1821

[Closed](#)

## analysisd: fix off-by-one in OS\_CleanMSG. #1824


[Merged](#)[ddpbsd](#) closed this as completed in [#1824](#) on Jan 16, 2020

[cpu](#) changed the title ~~analysisd: OS\_CleanMSG off-by-one heap overflow cleaning syslog msgs.~~ CVE-2020-8443: analysisd: OS\_CleanMSG off-by-one heap overflow cleaning syslog msgs. on Jan 30, 2020

cpu commented on Jan 30, 2020

[Contributor](#) [Author](#)

This was assigned [CVE-2020-8443](#)

No one assigned
Labels
None yet
Projects
None yet
Milestone
No milestone
Development
Successfully merging a pull request may close this issue.
 <b>analysisid: fix off-by-one in OS_CleanMSG.</b> cpu/ossec-hids
1 participant
