New issue

Jump to bottom

CVE-2020-8448: analysisd: OS_CleanMSG segfault processing invalid msg location. #1815

Closed **cpu** opened this issue on Jan 15, 2020 · 1 comment · Fixed by #1823

cpu commented on Jan 15, 2020 • edited Contributor

In `src/analysisd/cleanevent.c` the `ossec-analysisd`'s `OS_CleanMSG` function handles the location portion of a message differently when the first character after a 2 digit ID is `(`, indicating it came from a remote agent via `ossec-remoted`.

When remote agent locations are processed `OS_CleanMSG` tries to null terminate the received location substring of the overall log message to store in `lf->location` by advancing past the `->` in the string to the first `:` character:

ossec-hids/src/analysisd/cleanevent.c
Lines 45 to 59 in abb36d4

```
45      /* Is this from an agent? */
46      if ( *msg == '(' )
47      {   /* look past '->' for the first ':' */
48          pieces = strchr(strstr(msg, "->"), ':');
49          if(!pieces)
50          {
51              merror(FORMAT_ERROR, ARGV0);
52              return(-1);
53          }
54      }
55
56      *pieces = '\0';
```

Unfortunately the nesting of `strstr` as the first argument to `strchr` on L48 doesn't account for the possibility of a location string that starts with `(` but doesn't contain a `->`.

E.g. processing the msg `00:(:` will result in the `strstr` returning NULL, meaning the `strchr` call will be `strchr(NULL, ":")` and a segfault will occur.

This code was introduced in `95cd0c9` on Nov 13, 2013. I believe it affects OSSEC 2.7+.

It seems that in all cases `ossec-remoted` will always write a well-formed location into the message it writes to the `ossec` UNIX domain socket queue because it uses `SendMSG` and populates the `locmsg` unconditionally:

ossec-hids/src/remoted/secure.c
Lines 187 to 190 in abb36d4

```
187     /* Generate srcmsg */
188     snprintf(srcmsg, OS_FLSIZE, "(%s) %s",
189              keys.keyentries[agentid]->name,
190              keys.keyentries[agentid]->ip->ip);
```

ossec-hids/src/shared/mq_op.c
Line 85 in abb36d4

```
85      snprintf(tmpstr, OS_MAXSTR, "%c:%s->%s", loc, locmsg, message);
```

I believe that means the only way this is triggerable is with direct write access to the socket, and probably makes the exploitability very low.

One possible fix (edit: implemented in #1823) is to separate the `strstr` and return an error when it returns `NULL` before performing the subsequent `strchr` with the result pointer.

This was referenced on Jan 15, 2020

OSSEC-HIDS Security Audit Findings #1821
Closed

analysisd: fix possible null ptr deref in OS_CleanMSG. #1823
Merged

ddpbsd closed this as completed in #1823 on Jan 16, 2020

cpu changed the title ~~analysisd: OS_CleanMSG segfault processing invalid msg location.~~ CVE-2020-8448: analysisd: OS_CleanMSG segfault processing invalid msg location. on Jan 30, 2020

cpu commented on Jan 30, 2020 Contributor Author

This was assigned CVE-2020-8448

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone

No milestone

Development

Successfully merging a pull request may close this issue.

analysisd: fix possible null ptr deref in OS_CleanMSG.
cpu/ossec-hids

1 participant