Suricata - Feature #1489
Log a message when memcap limit is reached
06/17/2015 08:33 AM - Alexander Gozman

| Status:   | New          |
| Priority: | Normal       |
| Assignee: | Community Ticket |
| Category: |             |
| Target version: | TBD         |
| Effort:   | low          |
| Difficulty: | low        |

**Description**
Probably, suricata should log a message when a memcap limit is reached. This can be done in ...CheckMemcap() functions. For instance, DNSCheckMemcap() sets STATE_MEMCAP_REACHED - close enough. Such log messages can be useful when testing and/or tuning suricata under high load.

**Related issues:**
Related to Feature #614: Rate limiting messages

**History**

#1 - 06/17/2015 12:25 PM - Victor Julien
I'm not sure Suricata itself should output this, or it could be a done in a lua script based on the stats.

#2 - 06/17/2015 02:10 PM - Alexander Gozman
Victor Julien wrote:

I'm not sure Suricata itself should output this, or it could be a done in a lua script based on the stats.

Yes, it could be done somewhere else. However snort emits debug messages when memcap is exceeded, so it provides a good debugging tool right out of the box. Anyway it's not a priority.

#3 - 06/28/2015 05:24 AM - Peter Manev
I think it is very useful/helpful in terms of debug/tuning to have those (optionally enabled maybe) verbose/dbg msg in suricata.log with regards to memcaps being reached.

#4 - 07/23/2015 04:48 AM - Victor Julien
I think if we do such a thing we need a logging method that would allow us to say "log this once" or "log this no more than once a second". Any log message based on traffic is a risk wrt log flooding etc.

#5 - 07/23/2015 04:48 AM - Victor Julien
- Target version changed from 3.0RC1 to TBD

#6 - 08/05/2015 09:10 AM - Alexander Gozman
Victor Julien wrote:

I think if we do such a thing we need a logging method that would allow us to say "log this once" or "log this no more than once a second". Any log message based on traffic is a risk wrt log flooding etc.

Well, I think this can be either hardcoded or have a setting in a configuration file (like "memcap-limit-warn-count: 5", value of -1 will log it without any limit). And we can implement a macro like this one:

```c
#define DO_FIRST_N(max, stmt) \
  do { \
    static volatile int logLimiter = 0; \
    if (++logLimiter > (max)) { \
      break; \
    } \
  } \
  stmt;
```
And use it like:

```c
DO_FIRST_N(1, SC_LOG_WARNING(...));
```

Maybe there's more neat and tidy solution :)

---

**#7 - 08/17/2015 10:07 AM - Victor Julien**

This variable logLimiter will only exist in the scope of the DO_FIRST_N(1, SC_LOG_WARNING(...)); statement, right? How will it be shared between threads or multiple invocations of a code block?

**#8 - 08/17/2015 10:34 AM - Alexander Gozman**

Victor Julien wrote:

> This variable logLimiter will only exist in the scope of the DO_FIRST_N(1, SC_LOG_WARNING(...)); statement, right? How will it be shared between threads or multiple invocations of a code block?

If I remember correctly, static variables have a local scope but a global lifetime. So this one should be shared between threads and work correctly after multiple calls (however, there may be a non-critical race condition with this simple counter). Some time ago I did a quick test, with and without threads, and it seemed to work.

Maybe there's another solution for this, but I've tried to implement something like LOG_FIRST_N macro from google logging library (glog).

---

**#9 - 01/01/2016 05:46 PM - Andreas Herz**

- Assignee set to OISF Dev

**#10 - 08/09/2018 10:30 AM - Victor Julien**

- Assignee changed from OISF Dev to Anonymous
- Priority changed from Low to Normal
- Effort set to low
- Difficulty set to low

**#11 - 02/23/2019 10:16 PM - Andreas Herz**

- Assignee set to Community Ticket

**#12 - 09/26/2019 10:35 AM - Victor Julien**

- Related to Feature #614: Rate limiting messages added