Suricata - Feature #4150
Proiling mode: Ticks used to generate an alert available?
11/14/2020 02:44 PM - Jeff Lucovsky

<table>
<thead>
<tr>
<th>Status:</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>Community Ticket</td>
</tr>
<tr>
<td>Category:</td>
<td></td>
</tr>
<tr>
<td>Target version:</td>
<td>TBD</td>
</tr>
<tr>
<td>Effort:</td>
<td>medium</td>
</tr>
<tr>
<td>Difficulty:</td>
<td>low</td>
</tr>
<tr>
<td>Label:</td>
<td>Beginner</td>
</tr>
</tbody>
</table>

**Description**
StianB Nov 12th at 11:24 AM
When running Suricata in rule profiling mode, could we get the information about the number of ticks the rule used to create an Alert in the EVE? In a metadata field maybe?

4 replies

Andreas Herz  1 day ago
I'm not sure if I understand that correctly. You don't want the overall ticks but just those that were necessary for the alert trigger?

StianB  1 day ago
Yes, as an addition to the current rule-profiling output.

Andreas Herz  1 day ago
I see, I would recommend creating a redmine ticket for that feature request

Andreas Herz  1 day ago

**Related issues:**
Related to Task #4097: Suricon 2020 brainstorm  Assigned

**History**

#1 - 11/14/2020 02:44 PM - Jeff Lucovsky
- Related to Task #4097: Suricon 2020 brainstorm added

#2 - 11/19/2020 09:19 PM - Victor Julien
- Assignee set to Community Ticket
- Target version set to TBD
- Effort set to medium
- Difficulty set to low
- Label Beginner added

I think this should be fairly trivial. At the end of the individual rule inspection we can know the ticks, so we should be able to store than in the PacketAlert struct and output it in EVE.

#3 - 01/24/2022 06:12 PM - Matthew Davis
Should this only be done when "event_type"="alert" in the eve.json?

#4 - 01/24/2022 07:06 PM - Jeff Lucovsky
Yes -- logs with an alert event_type are produced starting with a call to JsonAlertLogger (src/output-json-alert.c)

#5 - 02/01/2022 02:46 PM - Matthew Davis
Thanks Jeff. I'm spending extra time on this one trying to read and understand the project better.

It looks like it makes sense to me to read the value of the ticks in detect-engine-alert.c:PacketAlertAppend()
But there are several spots in scope that look like potential candidates for getting the value.

In DetectEngineThreadCtx det_ctx, it has SCProfileData and SCProfileKeywordData each with ticks_match and ticks_no_match.

Then in Packet p, it has PktProfiling p. When I take a look at that:

typedef struct PktProfiling{
    uint64_t ticks_start;
    uint64_t ticks_end;

    PktProfilingTmmData tmm[TMM_SIZE];
    PktProfilingData flowworker[PROFILE_FLOWWORKER_SIZE];
    PktProfilingAppData app[ALPROTO_MAX];
    PktProfilingDetectData detect[PROF_DETECT_SIZE];
    PktProfilingLoggerData logger[LOGGER_SIZE];
    uint64_t proto_detect;
} PktProfiling;

it not only has ticks_start and ticks_end but also all of the PktProfiling* structs contain various ticks attributes. Which one should I be using to set the value of the ticks displayed in the output?