Suricata - Documentation #2699

document all eve record types and fields

11/21/2018 02:45 PM - Victor Julien

Status: Assigned  
Priority: Normal  
Assignee: Sascha Steinbiss  
Category:  
Target version: TBD  
Affected Versions:  
Effort: medium  
Difficulty:  
Label:  

Description

For each document type, document fields and their types. Add examples.

It's probably best to add specific tickets for each of the record types.

Related issues:

Related to Task #2685: SuriCon 2018 brainstorm  
Related to Documentation #2620: Documentation: tagged_packets / event_type packet added

History

#1 - 11/21/2018 02:46 PM - Victor Julien
- Assignee set to Robert Haist
- Target version set to TBD

#2 - 11/21/2018 02:46 PM - Victor Julien
- Related to Task #2685: SuriCon 2018 brainstorm added

#3 - 11/21/2018 02:46 PM - Victor Julien
- Related to Documentation #2620: Documentation: tagged_packets / event_type packet added

#4 - 12/03/2018 09:43 PM - Robert Haist
- Effort set to medium

Working on it over at Github: https://github.com/rhaist/suricata-json-schema

Will probably take some time until we have a fully reproducible build but you can expect a preview soon.

#5 - 12/04/2018 08:53 AM - Victor Julien

Maybe I'm misunderstanding the purpose of the schema, but the goal of this ticket is to get the userguide updated so that all missing EVE fields are documented.

The JSON schema ticket is #1369

#6 - 06/14/2019 09:44 PM - Jason Ish

Victor Julien wrote:

Maybe I'm misunderstanding the purpose of the schema, but the goal of this ticket is to get the userguide updated so that all missing EVE fields are documented.

The JSON schema ticket is #1369

I think the 2 are tightly related.

I had started to look at this again, more about how it should look to the end user. I played with using tables in Sphinx, but I don't find that scales well, especially if you want to reformat. When I jumped back to my JSON schema stuff, it is kind of ugly and I'm not sure if it can be used to generate suitable doc for the userguide. So my last attempt is just some custom YAML that I thought I might generate into Sphinx tables. Still not sure if that is a good idea though, given that JSON schema exists.
Ideally there should be one source of truth. If we still feel that JSON schema is suitable for QA testing, maybe that should be it. We could probably do some intermediate processing of it, and perhaps adding extra fields to provide context in end-user doc. By context I mean stuff like: "vlan - only present when the alerting packet has a vlan header".

#7 - 09/24/2019 09:02 PM - Andreas Herz
- Assignee changed from Robert Haist to Sascha Steinbiss

#8 - 09/25/2019 07:07 PM - Andreas Herz
- Tracker changed from Feature to Documentation