Suricata - Feature #2756
Task # 5074 (New): rules: structured rule input

rules: input in json format
12/21/2018 11:49 AM - Victor Julien

Status: New
Priority: Normal
Assignee:
Category:
Target version: TBD
Effort: medium
Difficulty: high

Description
The idea here is that the new rule analyzer can output rules in json. It's quite incomplete at this stage, but when it is complete it would make sense to accept this output as input as well.

Related issues:
Related to Task #2685: SuriCon 2018 brainstorm New
Related to Task #4762: Suricon 2021 brainstorm New

History
#1 - 12/21/2018 11:49 AM - Victor Julien
- Related to Task #2685: SuriCon 2018 brainstorm added

#2 - 03/11/2019 01:52 PM - Victor Julien
- Assignee set to Community Ticket

#3 - 03/12/2019 11:39 AM - Victor Julien
- Target version set to TBD
  - Effort set to medium
  - Difficulty set to high

#4 - 10/22/2021 05:52 PM - Victor Julien
- Related to Task #4762: Suricon 2021 brainstorm added

#5 - 10/26/2021 09:52 AM - Victor Julien
- Subject changed from rules input in json format to rules: input in json format
- Assignee deleted (Community Ticket)

There was quite a bit of interest at suricon 2021. I think the first step should be to define a JSON schema / definition document as a rst doc in a PR to the suricata github repo.

Wrt implementation, I could imagine we start experimenting in suricata-update or a different tool outside of suricata proper, and have that "compile" the JSON into the existing rule format at first.

#6 - 10/26/2021 12:34 PM - Jason Ish
Victor Julien wrote in #note-5:

There was quite a bit of interest at suricon 2021. I think the first step should be to define a JSON schema / definition document as a rst doc in a PR to the suricata github repo.

Wrt implementation, I could imagine we start experimenting in suricata-update or a different tool outside of suricata proper, and have that "compile" the JSON into the existing rule format at first.

The 2 pass parser experimenting with in [https://redmine.openinfosafoundation.org/issues/3317](https://redmine.openinfosafoundation.org/issues/3317) could be extended. This tokenizer/lexer could be extended to be much more fine-grained to the point of breaking down each field in a byte_jump and others to their own struct elements. Essentially
creating an AST which is the result of a first pass.

As a result, the rules could be dumped in JSON in mass (with serde) as a conversion tool, and as the data structures are all created now, JSON rules could be consumed with serde as well.

My hesitation with experimenting with this in suricata-update is I think it's actually easier to work with this sort of stuff in Rust, and the end result would likely be Rust.

#7 - 02/14/2022 10:02 AM - Victor Julien

- Parent task set to #5074