Suricata - Feature #3311
Add better default suricata configuration for different traffic sizes and cpu/system architectures
11/03/2019 10:33 AM - Peter Manev

Status: New
Priority: Normal
Assignee: OISF Dev
Category: 
Target version: TBD
Effort: 
Difficulty: 

Label: 

Description
Related to improving Out of the Box Experience.

Often enough users struggle with coming up with a decent 1Gbps suricata.yaml config for example. It will be useful in terms of user experience to ship/install Suricata with some recommendations/examples for the following scenarios:
- 1 Gbps
- 10 Gbps
- 2-3 (sniffing) port
- IPS set up for AFP

etc..
The above should be based on certain assumptions (mainly available CPU/RAM etc).

History
#1 - 11/05/2019 11:32 AM - Victor Julien
- Assignee set to OISF Dev
- Target version set to TBD

I think this ticket contains 2 separate tasks: 1) create an easy system to produce these configs based on a single 'master' yaml. 2) define the various settings for the various profiles.

#2 - 11/07/2019 09:54 PM - Andreas Herz
Would you then ship different suricata.yaml files or is it just a documentation thingy?

#3 - 11/09/2019 09:32 AM - Victor Julien
The idea is to ship multiple yamls for those different performance profiles.

#4 - 11/11/2019 05:07 PM - Jason Ish
A few thoughts here.

We should identify all the fields in the suricata.yaml that would need to be customized and put place_holder type values in them. Then using YAML, we could create a file with a list of named profiles to provide these values. A Python script (or even sed) could output a config with the place_holder values replaced. Idea for profiles could be AWS instance types, or suggested settings for certain requirements.

Would it make sense for a script to profile the system (memory size, etc) and auto-generate a profile?

Script could be part of suricatactl.

#5 - 12/30/2019 08:50 AM - Peter Manev
A script would make sense indeed - though the expectation would be that Suri only would be running on the system.
Long time ago I started this - https://github.com/pevma/AAIS as part of similar effort.

We could also set up couple of "hardcoded" configs that aim at covering 1Gbps setups - those should be pretty easy I think. A 10Gbps setup would be a bit more complex as it would depend actually also on NUMA/Intel/AMD architecture.