Suricata - Feature #3688
Re-implement fast_pattern:only; in some way
05/02/2020 05:43 PM - Jason Williams

Status: Feedback
Priority: Normal
Assignee: Jason Williams
Category:
Target version: TBD
Effort:
Difficulty:

Description
I've been working on updating the Suricata 5 ET/ETPRO set over the past few weeks to proper notation and it is very apparent to me that we need some kind of way to do the below.

If i have a rule that is looking for the string "test" at the beginning of http.user_agent, i can write a rule snippet such as:

http.user_agent; content:"test"; startswith;

Which is fine, unless it is the fast pattern which would cause many unnecessary checks on things that are not in the buffer i want.

So to get a good fast_pattern on this i have to revert to using http.header

http.header; content:"User-Agent|3a 20|test"; fast_pattern;

So now i get a better fast_pattern match as its a more unique string in the flow, but i lost the ability to look in the smaller faster buffer of http.user_agent.

I would like to be able to do the below in suri rule language:

content:"User-Agent|3a 20|test"; fast_pattern:only; http.user_agent; content:"test"; startswith;

History
#1 - 05/02/2020 06:01 PM - Victor Julien
http.user_agent; content:"test"; startswith; is essentially the same as http.user_agent; content:"test"; depth:4; which will be used by AC/HS to only evaluate it against the first 4 bytes. So this should be efficient, much more so than http.header; content:"User-Agent|3a 20|test"; fast_pattern;.

I don't understand the last example. What is it supposed to do?

#2 - 05/03/2020 04:10 PM - Jason Williams
Victor Julien wrote in #note-1:

http.user_agent; content:"test"; startswith; is essentially the same as http.user_agent; content:"test"; depth:4; which will be used by AC/HS to only evaluate it against the first 4 bytes. So this should be efficient, much more so than http.header; content:"User-Agent|3a 20|test"; fast_pattern;.

I don't understand the last example. What is it supposed to do?

Ok, I will do some more testing and see what we see in terms of perf. The last example was something we used to do in rules, but I did it wrong anyways, there was no reason to double match on the same content, bad example on my part.

#3 - 05/12/2020 09:02 AM - Victor Julien
- Status changed from New to Feedback
- Assignee set to Jason Williams
- Target version set to TBD