Suricata-Update - Feature #3202
classtypes: validate classtypes in use

10/01/2019 05:31 AM - Victor Julien

Status: Rejected
Priority: Normal
Assignee: Shivani Bhardwaj
Category: 
Target version: Soon
Effort: 
Difficulty: 
Label: 

Description
If a ruleset has many rules that use undefined classtypes the test phase of Suricata will output and error for each of the rules.

I think Suricata-Update should validate that the classtype in the rule is defined in the classification.config.

Related issues:
Blocked by Feature #3203: manage classification.config Closed

History
#1 - 10/01/2019 05:31 AM - Victor Julien
- Tracker changed from Support to Feature

#2 - 10/01/2019 05:37 AM - Victor Julien
- Related to Feature #3203: manage classification.config added

#3 - 10/01/2019 06:06 AM - Jason Ish
What should happen if the validation does not pass? If this should abort, and restore the old ruleset, we get that with the pass of Suricata -T to validate the rule load.

#4 - 10/01/2019 06:08 AM - Victor Julien
Validation to me means: it should check that classtypes are defined. Generate a single warning per undefined classtype. Disable rules that contain undefined classtypes. At the end log a count of how many are disabled for this reason.

#5 - 10/01/2019 06:11 AM - Victor Julien
Although maybe this is too much of a 'silent failure'. Many rules will fail to load while SU only generates warnings. Guess erroring out may be better after all.

#6 - 10/01/2019 06:12 AM - Jason Ish
So auto-disable rules that don't have a known classification?

So implementation wise:
- Read in system classification.config (/usr/share/suricata/classification.config or /etc/suricata/classification.config).
- Read in any classification.config from any downloaded rulesets.
- Merge...
- For any rule with unknown classification, log warning, disable rule.

#7 - 10/01/2019 06:16 AM - Jason Ish
Victor Julien wrote:

Although maybe this is too much of a 'silent failure'. Many rules will fail to load while SU only generates warnings. Guess erroring out may be better after all.

Yes, I think I agree.

Suricata-Update does not know anything about classification.config in its current form, so its non-trivial to fix. Pure validation is done by Suricata during the test phase, so already get validation for free. If we want to do something about an invalid/unknown classification, then yes, Suricata-Update should do some pre-validation and fixup.

03/17/2022 1/2
How should we differentiate between the Snort's classification.config and the ruleset's in the merged classification.config (only for readability and clarity purposes)?

Also, just to be clear, are we looking at two levels of validation for classification.config? First in suricata-update then the usual in suricata?

I wouldn't worry about. One output file. I'd start with the engine included classification.config, then append new ones found while loading rules to the end of it. Would be nice to include a

# From ruleset ...

But we currently throw that information pretty early on, so it could really change the effort on this one.

Also, just to be clear, are we looking at two levels of validation for classification.config? First in suricata-update then the usual in suricata?

I'm not too keen on validation in Suricata-Update? What do we do if its invalid?

Outputting an up to date classification.config gives everything a higher chance of being valid tho. So I'd be in favour of just having Suricata do the validation.

Closing. I'm not sure if this is an issue that needs fixing anymore. We still need to make more use of the Suricata-Update managed classification file which is almost an alternative to this issue.