Matching a long list of tls.fingerprint fields is extremely CPU intensive

10/17/2019 12:01 AM - Michal Purzynski

Status: Closed
Priority: Normal
Assignee:
Category:
Affected Versions: 4.1.5
Label:

Description
The suricata-update script comes with some useful rule sources, like the sslbl.abuse.ch.

This list alone, when combined with the ET Pro ruleset makes the CPU usage on my sensor go from 6% to 60% and even 100% on as little as 400Mbit/sec of traffic.

This sensor is way oversized - Intel(R) Xeon(R) Gold 6126 CPU @ 2.60GHz x 2 so 24 cores tuned to perfection.

Removing just those 3000 rules dropped the CPU usage to as little as 6%. I repeated the test thrice (just so I could say thrice, because I never get to say thrice).

Feel free to tell me the engine works like designed and it's just abused here, and the intelligence framework / datasets should be used instead. In this case let's move it to suricata-update so people don't experience problems by default.

This is Suricata version 4.1.5 RELEASE

07/15/2022
History

#1 - 10/20/2019 08:50 PM - Andreas Herz
- Tracker changed from Bug to Support
- Assignee set to Community Ticket
- Target version set to 70

#2 - 08/07/2020 02:11 PM - Victor Julien
- Status changed from New to Closed
- Assignee deleted (Community Ticket)

Use tls.cert_fingerprint; dataset:isset,tls-fingerprints; you should be able to get much better perf.

But even w/o datasets tls.cert_fingerprint will outperform the legacy tls.fingerprint.