Suricata - Feature #2127
Rate Limiting by Tracking Both Source & Destination IP Addresses
06/04/2017 01:51 PM - David Lam

Status: Closed
Priority: Normal
Assignee: Ruslan Usmanov
Category:
Target version:
Effort: medium
Difficulty: medium

Description
When using Suricata to monitor inline router traffic, the current rate limiting features are limited in usability when both the source and destinations are unknown identities.
For example, if one was to run a guest network in the 10.0.0.0/8 range and allow these users to access the Internet, it would be very difficult to detect and thwart brute force like activities coming from the guest network because one would have to compromise on how the matching is performed.

Let say someone was running a script to continuously make connections to a popular website in order to try stolen credit card numbers. The script makes 30 new connections per second to the site over port 443 in order to determine what card numbers are potentially valid. To discourage this behavior, one may put in a rule that only allows a client to have 25 SYNs to any particular IP at port 443 in a period of 2 seconds, with a timeout of 30 seconds. However, the current ability is to track either by src or dst, which means if doing it by src I would have to make an educated guess as far as how many SYNs are appropriate for a client to make to the global Internet on port 443 given a particular source regardless of the destination, which is difficult to guess and is much more prone to false positives since a user could be running a lot of applications on the system which is contributing to the SYN load to various IPs on the Internet. On the other hand, tracking by dst is not very helpful either because if the brute force activities were against a popular destination, it will cause the whole network to lose access to that resource (e.g. Gmail).

Hence it would be very helpful if one can track by both, where both src-dst are used as a pair for the rule rate limits in order to accomplish what I mentioned above. The use case can also be applied to other types of activities, such as port scans, SSH brute force, SYN floods, etc., and the implementation could be used as an effective tool to reduce the number of abuse complaints to the network administrator.

History
#1 - 06/06/2017 02:08 PM - Victor Julien
- Target version set to TBD
I think this would be useful. Implementation shouldn't be terrible hard either, as we already have the 'ip pair' table. It's very similar to how we already implemented the per 'host' support.
Any takers?

#2 - 06/27/2017 04:27 PM - Andreas Herz
- Assignee set to Anonymous

#3 - 11/01/2017 12:12 PM - Ruslan Usmanov
I want to implement it, will create pull request soon.

#4 - 11/01/2017 03:42 PM - Ruslan Usmanov
https://github.com/OISF/suricata/pull/2965

#5 - 11/02/2017 12:54 PM - Ruslan Usmanov
https://github.com/OISF/suricata/pull/2970

#6 - 07/13/2018 08:08 PM - Victor Julien
- Effort set to medium
- Difficulty set to medium

07/17/2022 1/2
#7 - 02/23/2019 10:20 PM - Andreas Herz
- Assignee set to Community Ticket

#8 - 09/27/2019 11:16 AM - Victor Julien
- Status changed from New to Closed
- Assignee changed from Community Ticket to Ruslan Usmanov
- Target version deleted (TBD)

https://github.com/OISF/suricata/pull/3056

#9 - 09/27/2019 05:18 PM - David Lam
Is this now implemented somewhere?