Suricata - Feature #120
Capture full session on alert
03/12/2010 07:07 PM - Dave Smith

Status: In Review
Priority: Low
Assignee: Scott Jordan
Category: 
Target version: 7.0rc1
Effort: high
Difficulty: medium

Description
On alert, Snort either captures only the individual packet that hits on a rule, or ‘n’ number of packets thereafter that you specify with the tag keyword. Investigation of an alert is usually pretty hard with only that one packet, unless you happen to separately be saving all traffic on the wire to disk, and can go and retrieve the relevant pcap from elsewhere manually. Tagging will only get you packets after the alert - if your rule hits a few packets into the session, the previous packets are lost.

It would be great to have the capability to capture an entire session. I previously worked for a large multinational company that had a proprietary, in-house developed IDS that did this. Its engine held a rolling packet buffer of a couple hundred MB of of traffic from the wire that the engine could reach back into, to collect the beginning of the session, and it seemed to work quite well.

Related issues:
Related to Task #2309: SuriCon 2017 brainstorm
New
Related to Documentation #2219: Save pcap only if alert
Assigned
Related to Task #4097: Suricon 2020 brainstorm
New
Related to Bug #5189: Suricata alerts pcap issue
New
Has duplicate Feature #385: Configuration option to log all known (pcap) data...
Closed

History
#1 - 03/16/2010 11:48 AM - Will Metcalf
I really like this idea. Being multi-threaded it would be nice to have a pcap logging thread that supports file rotation similar to tsharks ring buffer or daemonlogger. Maybe we could even have multiple targets similar to time machine so that you could specify x mb to be written to mem (primary storage) and then have it migrated to disk (secondary) storage once you ran out of allocated space. Using this method may result in performing tasks such session extraction to be minimally invasive performance wise if you stick with the in mem buffer. It would also be nice to have flow stats as well similar to something like argus

#2 - 04/06/2011 08:01 AM - Victor Julien
- Assignee set to Anonymous
- Priority changed from Normal to Low

I like this idea as well, but I'm not convinced it should be part of Suricata. There are many tools that can log traffic (including Suricata in current git) and based on the alerts Suricata emits a 3rd party tool could easily extract that streams. Kind of like what Sguil does manually, but then in an automated way. My vote is to leave this to post-processing outside of Suricata.

#3 - 07/04/2012 03:40 AM - Victor Julien
- Target version set to TBD

#4 - 07/29/2014 05:10 PM - Brian Keefer
Will Metcalf wrote:

I really like this idea. Being multi-threaded it would be nice to have a pcap logging thread that supports file rotation similar to tsharks ring buffer or daemonlogger. Maybe we could even have multiple targets similar to time machine so that you could specify x mb to be written to mem (primary storage) and then have it migrated to disk (secondary) storage once you ran out of allocated space. Using this method may result in performing tasks such session extraction to be minimally invasive performance wise if you stick with the in mem buffer. It would also be nice to have flow stats as well similar to something like argus

Yes to all of this. Critically though, like the original request says it's a must-have to log packets to PCAP from the stream that caused an alert. Working with barnyard2 to get this data is unwieldy. The great thing about Snort/Sourcefire is being able to use tcpdump & Wireshark to analyze

03/17/2022 1/2
payload of a stream that caused alert. Just being able to do that is bare minimum IMO.

#5 - 08/24/2015 10:14 AM - Victor Julien
- Assignee changed from Anonymous to Mathew Oldham

#6 - 12/01/2017 05:02 AM - Victor Julien
- Related to Task #2309: SuriCon 2017 brainstorm added

#7 - 07/17/2018 08:13 AM - Victor Julien
- Assignee changed from Mathew Oldham to Anonymous
- Effort set to high
- Difficulty set to medium

#8 - 07/17/2018 08:23 AM - Victor Julien
- Related to Feature #385: Configuration option to log all known (pcap) data for a stream when an alert fires added

#9 - 02/23/2019 10:07 PM - Andreas Herz
- Assignee set to Community Ticket

#10 - 11/01/2019 02:10 PM - Andreas Herz
- Related to Documentation #2219: Save pcap only if alert added

#11 - 11/01/2019 02:12 PM - Andreas Herz
Add a global rule tag keyword

#12 - 11/02/2019 09:00 AM - Victor Julien
- Related to deleted (Feature #385: Configuration option to log all known (pcap) data for a stream when an alert fires)

#13 - 11/02/2019 09:00 AM - Victor Julien
- Has duplicate Feature #385: Configuration option to log all known (pcap) data for a stream when an alert fires added

#14 - 09/16/2020 01:40 PM - Victor Julien
- Status changed from New to In Review
- Assignee changed from Community Ticket to Scott Jordan
- Target version changed from TBD to 7.0rc1

https://github.com/OISF/suricata/pull/5345

#15 - 11/27/2020 04:52 PM - Jason Ish
- Related to Task #4097: Suricon 2020 brainstorm added

#16 - 03/16/2022 05:31 PM - Jason Ish
- Related to Bug #5189: Suricata alerts pcap issue added

#17 - 03/16/2022 06:00 PM - Chatak Kumar
Tried with latest version also https://github.com/OISF/suricata/pull/6941
But issue in packets formed by suricata alert pcap
I tried with this pcap https://www.malware-traffic-analysis.net/2021/07/14/index.html
Check http only in wireshark and compare with Suricata log.pcap . We can see Info/header is truncated which makes hard to understand.