Suricata - Bug #1524
Potential Thread Name issues due to RHEL7 Interface Naming Contentions

08/05/2015 04:48 PM - Zach Rasmor

Status: Closed
Priority: Normal
Assignee: Zach Rasmor
Category: Target version: 3.1rc1
Affected Versions: Difficulty:
Effort: Label:

Description
I have begun testing Suricata on RHEL7, on an interface named "enp132s0." The length of this interface name resulted in the following thread names for my 16 threads:

RxPcapenp132s01
RxPcapenp132s02
RxPcapenp132s03
RxPcapenp132s04
RxPcapenp132s05
RxPcapenp132s06
RxPcapenp132s07
RxPcapenp132s08
RxPcapenp132s09
RxPcapenp132s01 <--
RxPcapenp132s01 <--
RxPcapenp132s01 <--
RxPcapenp132s01 <--
RxPcapenp132s01 <--
RxPcapenp132s01 <--
RxPcapenp132s01 <--

Furthermore, the stats log only displayed threads 1-9.

$ grep flow /var/log/suricata/stats.log | tail -20
flow.emerg_mode_entered | FlowManagerThread | 0
flow.emerg_mode_over   | FlowManagerThread | 0
flow.tcp_reuse         | FlowManagerThread | 0
tcp.no_flow            | RxPcapenp132s01  | 0
tcp.no_flow            | RxPcapenp132s02  | 0
tcp.no_flow            | RxPcapenp132s03  | 0
tcp.no_flow            | RxPcapenp132s04  | 0
tcp.no_flow            | RxPcapenp132s05  | 0
tcp.no_flow            | RxPcapenp132s06  | 0
tcp.no_flow            | RxPcapenp132s07  | 0
tcp.no_flow            | RxPcapenp132s08  | 0
tcp.no_flow            | RxPcapenp132s09  | 0
flow_mgr.closed_pruned | FlowManagerThread | 0
flow_mgr.new_pruned    | FlowManagerThread | 31431
flow_mgr.est_pruned    | FlowManagerThread | 0
flow.memuse            | FlowManagerThread | 464960416
flow.spare             | FlowManagerThread | 1050507
flow.emerg_mode_entered| FlowManagerThread | 0
flow.emerg_mode_over   | FlowManagerThread | 0
flow.tcp_reuse         | FlowManagerThread | 6

This truncation in the thread names can be traced back to the following:

From tm-threads.h:
#define TM_QUEUE_NAME_MAX 16
#define TM_THREAD_NAME_MAX 16
And from util-runmodes.c, RunModeSetLiveCaptureWorkersForDevice:

```c
for (thread = 0; thread < threads_count; thread++) {
    char tname[TM_THREAD_NAME_MAX];
    ...
    snprintf(tname, sizeof(tname), "%s%s"PRIu16, thread_name, live_dev, thread+1);
```

I can certainly look into changing the name of my interface to something shorter, but I figured this was worth mentioning since RHEL7 has adopted a new interface naming convention, and this seems to cause issues with stats reporting.

Perhaps the fix is as simple as raising the #define MAX's to 32, but I wasn't sure if that would have unwanted consequences.

History

#1 - 08/17/2015 08:24 AM - Victor Julien
- Target version set to 3.0RC1

Some works has been done to address this, but it's not yet complete. Latest iteration is at https://github.com/inliniac/suricata/pull/1466. As Andreas doesn't seem to be working on it, perhaps you can pick it up?

#2 - 08/17/2015 02:24 PM - Zach Rasmor
Sure, I can take a look at what is done and move it forward.

Two questions:
1. Though I haven't looked at the code in detail yet, I noticed that the PR you referenced is closed without comments. Do you recall what the remaining issues were?
2. What is the process for augmenting another contributor's work - fork his branch and open my own pull request with my commits on top of his?

#3 - 08/18/2015 03:03 AM - Victor Julien
1) The feedback was given in IRC, here are my logs:

maxmo, I think I see the issue: the thread name is directly assigned to 'tv->name'. So tv->name now points to a variable that goes out of scope
maxmo, previously it pointed to a string constant
maxmo, quick and dirty would be passing the result of strdup(name) to TmThreadCreateMgmtThreadByName
maxmo, if you're interesting in doing a bit more work, you could change the tv->name ptr into a small fixed size string

2) You can take the branch, rebase it to my master and then make additional changes in new commits. So yeah, fork his branch and add your own commits.

#4 - 08/18/2015 02:37 PM - Zach Rasmor
- Assignee set to Zach Rasmor

#5 - 08/23/2015 10:57 PM - Zach Rasmor
Hello Victor, I wanted to confirm a couple of things:

In the discussion on https://github.com/inliniac/suricata/pull/1355, you suggested the format "W#12 => worker thread 12 (workers and non-capture threads in autofp)"

Regarding Autofp, the current name for non-capture threads is "Detect"...just wanted to see if you still preferred it shortened to "W#-iface" (as you stated above) or if you'd rather have it as "D#-iface" to closer resemble the current name?

Also, for single mode, do you have a preference between "C#-iface", "W#-iface", or something else? I was thinking "W" might make more sense here.

#6 - 08/25/2015 02:44 AM - Victor Julien
- Status changed from New to Assigned

For autofp I prefer "W" as well, as it's not just doing detect, but also stream tracking/reassembly and the outputs.

ACK on single: single is really the same as 'workers', except there is just one thread and not multiple.

On the IPS side we haven't figured things out yet. There we have a verdict/transmit thread as well. Maybe capture should be RX and verdict/transmit should be TX.

Something like:
RX#1-eth0
TX#1-eth0
#7 - 11/25/2015 02:18 AM - Victor Julien
- Target version changed from 3.0RC1 to 70

#8 - 05/21/2016 02:25 AM - Victor Julien
- Status changed from Assigned to Closed
- Target version changed from 70 to 3.1rc1

https://github.com/inliniac/suricata/pull/2037