Snort 3 Multiple Packet Processing Threads

Generated: 2018-08-29

This guide introduces Snort 3 capabilities for running multiple packet processing threads. Using the new option `--max-packet-threads` or `-z` Snort will start N packet processing threads, where N is the number of threads specified after the `--max-packet-threads` or `-z` option with a maximum of 8 threads.

1. Processing Multiple PCAP Files

Running Snort against a single pcap file is achieved via the `-r` option. Snort can process multiple pcap files at a run via the `--pcap-dir` and `--pcap-filter` options. The `--pcap-dir` option allows specifying the directory from which Snort will recursively read pcap files. The `--pcap-filter` option filters the pcap files to read from the specified directory.

To employ multiple packet process threads, Snort 3 includes the option `--max-packet-threads` or `-z`. This option allows specifying the number of Snort threads to process network traffic.

Example – employ 4 threads to process pcap file ending with the pattern '*.pcap' from a directory called 'pcaps'

```
# snort -c snort.lua --pcap-dir ./pcaps --pcap-filter '*.pcap' -l /var/log/snort --
plugin-path /extra -k none -z 4
```

Reviewing Snort threads with the top program displays the 2 threads specified in the example above, plus an additional thread for logging as a result of using the `-l` option.

```
PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
17079 root      20   0 1297372   1.0g   8560 R 98.0 18.0   0:04.43 snort
17094 root      20   0 1297372   1.0g   8560 R 35.3 18.0   0:01.06 snort
17095 root      20   0 1297372   1.0g   8560 R 34.0 18.0   0:01.02 snort
17097 root      20   0 1297372   1.0g   8560 R  8.0 18.0   0:00.24 snort
17028 root      20   0 1297372   1.0g   8560 S  1.7 18.0   0:15.40 snort
```

Note that when using multiple threads while logging to files, each thread will generate its own set of log files, depending on the logging configured in `snort.lua` file.

```
# ls -l /var/log/snort/
```

```
-rw-------. 1 root root  49237 Aug 24 05:44 0_alert_fast.txt
-rw-------. 1 root root   3216 Aug 24 05:44 0_appid_stats.log
-rw-------. 1 root root  19240 Aug 24 05:44 0_data_log
-rw-------. 1 root root      0 Aug 24 04:39 0_file.log
-rw-------. 1 root root   7137 Aug 24 05:44 1_alert_fast.txt
-rw-------. 1 root root   7509 Aug 24 05:44 1_appid_stats.log
-rw-------. 1 root root  40982 Aug 24 05:44 1_data_log
-rw-------. 1 root root      0 Aug 24 04:39 1_file.log
-rw-------. 1 root root  14896 Aug 24 05:44 2_alert_fast.txt
-rw-------. 1 root root   2835 Aug 24 05:44 2_appid_stats.log
-rw-------. 1 root root 214707 Aug 24 05:44 2_data_log
-rw-------. 1 root root      0 Aug 24 05:44 2_file.log
-rw-------. 1 root root  13259 Aug 24 05:44 3_alert_fast.txt
-rw-------. 1 root root   3965 Aug 24 05:44 3_appid_stats.log
-rw-------. 1 root root  34574 Aug 24 05:44 3_data_log
-rw-------. 1 root root      0 Aug 24 05:44 3_file.log
```

If the `--id-subdir` option is used, then each thread will create a directory named after the thread's ID under the specified log directory or the default log directory `/var/log/snort`.

```
# ls -l /var/log/snort/
```

```
drwx------. 2 root root 83 Aug 24 05:45 0
drwx------. 2 root root 83 Aug 24 05:45 1
drwx------. 2 root root 83 Aug 24 05:45 2
drwx------. 2 root root 83 Aug 24 05:45 3
```

2. Processing Live Traffic from Network Interfaces

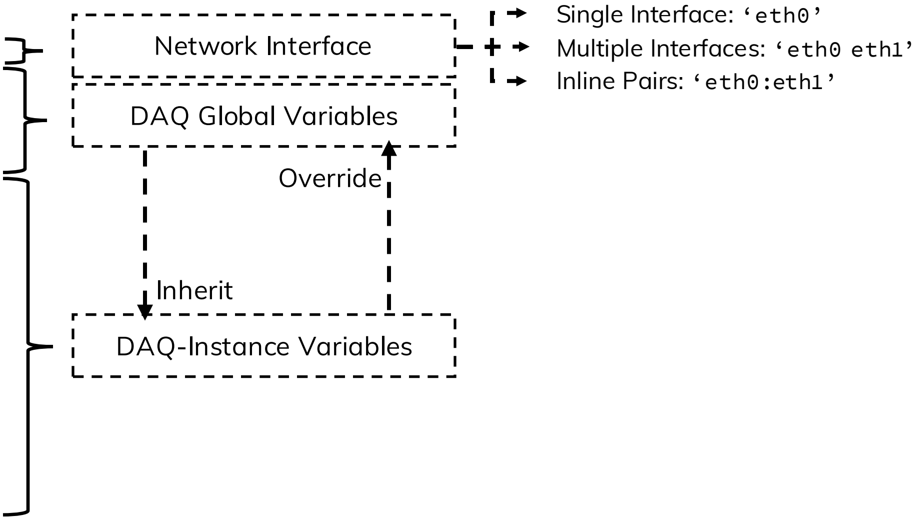
Running multiple packet processing threads involves:

1. Configuring DAQ by specifying its global variables and instance-specific variables. These configurations can be implemented via the configuration file `snort.lua` or via the command line.
2. Instructing Snort to run multiple threads via the option `--max-packet-threads` or `-z`.

The below DAQ example configured to `afpacket` module of DAQ against (`input_spec`) a single interface `ens192`. The global DAQ configuration (`variables`) section is setup to load balance incoming traffic against 2 instances (`lb_total`) using the kernel FANOUT capability.

The instance-specific variables are set per-instance. To ensure load balancing, each instance is given an ID (`lb_id`) within the total number of instances (`lb_total`). Note that instance-specific DAQ variables inherit configurations from the global variable and can override them as well.

```
daq = {
    module_dirs = {
        '/usr/local/lib/daq'
    },
    module = 'afpacket',
    input_spec = 'ens192',
    variables = {
        'lb_total=2',
        'fanout_type=hash'
    },
    instances = {
        {
            id = 0,
            variables = {
                'lb_id=1'
            }
        },
        {
            id = 1,
            variables = {
                'lb_id=2'
            }
        }
    }
}
```



The equivalent command line for running Snort with the above configurations looks like:

```
# snort -c snort.lua --daq-dir /usr/local/lib/daq --daq afpacket --daq-var
lb_total=4 --daq-var fanout_type=hash -i ens192 --daq-var lb_id=1 -i ens129 --daq-
var lb_id=2 -z 2
```

In other words, specifying DAQ global variable are set ahead of instance-specific variables, and for each instance, the same interface specifications must be specified.

3. References

- https://www.snort.org/downloads/snortplus/snort_manual.html
- https://github.com/snortadmin/snort3/tree/master/doc
- http://seclists.org/snort/2016/q3/383
- http://seclists.org/snort/2018/q3/151