How to compile Suricata with DAG support

Overview

Endace DAG™ (Data Acquisition and Generation) Data Capture Cards are the ultimate in dedicated network packet capture interface cards. They guarantee 100% packet capture on any network regardless of packet size, interface type, or network load.

Endace DAG cards are designed for capturing packets and delivering them to host memory with minimal CPU overhead and fidelity that is unobtainable with repurposed generic ASIC NIC hardware. Offloading packet capture from the host CPU means these resources are available to Suricata to perform analysis; ensuring the highest compute per rack unit available.

Requirements

The latest stable versions of the DAG software tools and Suricata should be used and currently these are:

- dag-5.5.0.tar.gz software archive
- suricata-3.1.2 .tar.gz software archive

For DAG customers the latest version is available at https://www.endace.com/endace-support.html

DAG Drivers and Libraries

Pre-compiled DAG driver library packages for a number of distributions are available on the Endace Support site. If your distribution does not have packages available, the source code can be compiled and installed.

The DAG drivers and libraries require a number of system libraries to be installed before compilation.

CentOS 7.x

```
yum install gcc gcc-c++ libxml2-devel kernel-devel-$(uname -r)
make
```

Ubuntu 16.04.1 LTS

```
apt-get install build-essential
```

Build and Install DAG Drivers and Libraries

Once these dependencies have been installed, and the DAG libraries downloaded, installation is straightforward:

```
tar xvf dag-5.5.0.tar.gz
cd dag-5.5.0
./configure
make
make install
ldconfig /usr/local/lib
```

Building Suricata with DAG Support

Once the DAG software is installed, it is time to prepare the system for installing Suricata. Although Suricata may already be available as a package on your distribution of choice, it is unlikely to have been built with DAG support, therefore compiling from source is required.

The dependencies required for Suricata can be installed as follows:

CentOS 7

```
yum install libtool libyaml-devel libpcap-devel pcre-devel
file-devel zlib-devel jansson-devel nss-devel libcap-ng-devel
libnet-devel lua-devel which
```

Ubuntu 16.04.1 LTS

```
apt-get install libpcre3 libpcre3-dbg libpcre3-dev libtool
libpcap-dev libnet1-dev libyaml-0-2 libyaml-dev zlib1g zlib1g-
dev libcap-ng-dev libcap-ng0 make libmagic-dev libjansson-dev
libjansson4 pkg-config
```

Build and Install Suricata

With the dependencies installed, download the latest version of Suricata and build as listed below:

```
tar xvf suricata-3.1.2.tar.gz
cd suricata-3.1.2
./configure --prefix=/usr --sysconfdir=/etc --enable-dag \
        --localstatedir=/var
make
make install
make install-full
ldconfig
```

Running Suricata

Once Suricata is installed, connect the DAG card to the traffic source and check for link and lock with:

```
dagconfig -d 0 default
dagconfig -d 0 -s
```

Check that the values underneath Lock and Link show value of 1.

```
dagconfig -d 0 -u
```

If the source is transmitting packets, check that values for Rx_Bytes, Rx_Frames show non-zero values. Values on Rx_FCS should not increment (Non-zero values are possible).

Validation

To quickly test that Suricata can read from the DAG interface, run Suricata from the commandline with:

```
/usr/bin/suricata -v --dag 0:0
```

Provided the DAG interface is configured properly and receiving packets /var/log/suricata/eve.json should start populating with events and other data from the network.

Multiple Streams

One of the unique capabilities of the DAG card is the ability to load balance packets in a flow safe manner across up to 32 receive memory streams. This allows multiple Suricata threads to run in parallel on separate CPU cores reading from separate memory streams.

The following is an example of how to configure the DAG card for 4 way load balancing, although in practice this should be scaled up to either the maximum number of cores available on the platform or the maximum number of receive streams on the DAG card:

```
dagconfig -d0 mem=128:0:128:0:128:0:128:0
dagconfig -d0 -S ipf_enable=on
dagconfig -d0 -S hash_encoding_from_ipf=on
dagconfig -d0 -S n_tuple_select=2
dagconfig -d0 -S hash_width=2
dagconfig -d0 -S hat_range=0-250:250-500:500-750:750-1000
dagcat-setup -d0 -m z4
```

With this four-way hardware load balancing configured on the DAG, Suricata can then be configured to listen on multiple streams, using a thread per stream:

```
/usr/bin/suricata -v --runmode workers --dag 0:0 --dag 0:2 \
        --dag 0:4 --dag 0:6
```

The best results will be found by matching the number of streams to the available number of cores and allocating as much memory to the DAG receive streams as is practical (this example shows 128MB allocated per stream).

For more information on the Endace portfolio of products, visit: endace.com/products

For further information, email: info@endace.com