NVIDIA DOCA DPI

Sample Guide

Table of Contents

Chapter 1. Introduction

Deep packet inspection (DPI) is a method of examining the full content of data packets as they traverse a monitored network checkpoint.

DPI provides a more robust mechanism for enforcing network packet filtering as it can be used to identify and block a range of complex threats hiding in network data streams more accurately. This includes:

▶ Malicious applications

▶ Malware data exfiltration attempts

▶ Content policy violations

▶ Application recognition

▶ Load balancing

For more information about the DOCA DPI library, refer to NVIDIA DOCA DPI Programming Guide.

Chapter 2. Dependencies

N/A

Chapter 3. Prerequisites

DPI-based applications can run either on the host machine or on the DPU target.

As DPI leverages the regular expressions (RegEx) engine, users must make sure it is enabled:

1. Allocate hugepages:
   ```
   sudo echo 2048 > /sys/kernel/mm/hugepages/hugepages-2048kB/nr_hugepages
   ```
2. Make sure the RegEx engine is active:
   ```
   systemctl status mlx-regex
   ```

   If the status is inactive (`Active: failed`), run:
   ```
   systemctl start mlx-regex
   ```
3. DPI compiler must be used since the RegEx engine accepts only `.cdo` files. The CDO files are constructed by compiling a signature file written in Cta open-source format.

   To compile the signature file, run the following:
   ```
   doca_dpi_compiler -i <path to signature file> -o /tmp/signatures.cdo -f suricata
   ```

   The `.cdo` file is created in the output path flagged as the `-o` input path of the compiler.
4. For gRPC related samples, assign the IP 192.168.102.2 to the server.

   Make sure to deploy the DPI gRPC server and needed setups according to the NVIDIA DOCA gRPC Infrastructure User Guide.

Chapter 4. Running the Sample

1. Refer to the following documents:
   ▶ NVIDIA DOCA Installation Guide for Linux for details on how to install BlueField-related software.
   ▶ NVIDIA DOCA Troubleshooting Guide for any issue you may encounter with the installation, compilation, or execution of DOCA samples.

2. To build a given sample:
```
cd /opt/mellanox/doca/samples/doca_dpi/<sample_name>
meson build
ninja -C build
```

> 💬 **Note:** The binary `doca_<sample_name>` will be created under `./build/`.

3. Sample (e.g., `dpi_scan`) usage:
```
Usage: doca_dpi_scan [DPDK Flags] -- [DOCA Flags] [Program Flags]
DOCA Flags:
  -h, --help                  Print a help synopsis
  -v, --version               Print program version information
  -l, --log-level             Set the log level for the program <CRITICAL=20,
 ERROR=30, WARNING=40, INFO=50, DEBUG=60>

Program Flags:
  -s, --sig-file              Signatures file path
```

4. For additional information per sample, use the –h option after the -- separator:
```
./build/doca_<sample_name> -- -h
```

5. DOCA DPI samples are based on DPDK libraries. Therefore, the user is required to provide DPDK flags. The following is an example from an execution on the DPU:
```
./build/doca_dpi_scan -a 0000:03:00.0,class=regex -a
 auxiliary:mlx5_core.sf.4,sft_en=1 -- -s /tmp/signatures.cdo
```

In contrast to DOCA DPI-based applications, the samples require only one network port. This is because they parse the packets they received without forwarding them.

> 💬 **Note:** When running on the DPU using the command above sub-functions must be enabled according to the Scalable Function Setup Guide.

> 💬 **Note:** When running on the host, virtual functions must be used according to the instructions in the NVIDIA DOCA Virtual Functions User Guide.

Chapter 5. Samples

5.1. DPI Scan

This sample illustrates how to use DPI programming interface on a single packet.

The sample logic includes:

1. Initializing DOCA DPI.
2. Loading the signature `.cdo` file into the RegEx engine.
3. Creating DPI flow for the incoming packet.
4. Inserting packet into DPI work queue.
5. Polling packet from work queue when processing ends.
6. Retrieving matched signature information if there is a match.
7. Retrieving statistics information.
8. Destroying DPI structures.

Reference:

▶ `/opt/mellanox/doca/samples/doca_dpi/dpi_scan/dpi_scan.c`

5.2. gRPC DPI Scan

This sample illustrates how to use gRPC DPI programming interface on a single packet.

The sample logic includes:

1. Initializing DOCA gRPC DPI client.
2. Loading the signature `.cdo` file into the RegEx engine.
3. Creating DPI flow for the incoming packet.
4. Inserting packet into DPI work queue.
5. Polling packet from work queue when processing ends.
6. Retrieving matched signature information if there is a match.
7. Retrieving statistics information.
8. Destroy DPI structures.

Reference:

- ▶ `/opt/mellanox/doca/samples/doca_dpi/grpc_dpi_scan/grpc_dpi_scan.c`

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
http://www.nvidia.com