NVIDIA DOCA App Shield

Sample Guide

MLNX-15-060521 _v1.5.1 | November 2022

Table of Contents

Chapter 1. Introduction

DOCA App Shield is a library for monitoring the host and authenticating the integrity of core processes.

For more information about DOCA App Shield library, refer to NVIDIA DOCA App Shield Programming Guide.

Chapter 2. Dependencies

The library requires a minimum DPU firmware version of 24.32.1010.

Chapter 3. Prerequisites

Make sure to follow the stages of the library prerequisites detailed in NVIDIA DOCA App Shield Programming Guide to make sure the library could be used by the samples. Afterwards, make sure to copy the generated JSON files into the DPU to the following path:

- `/tmp/symbols.json`
- `/tmp/mem_regions.json`

For more information regarding the runtime arguments, including how to get the VUID, please refer to section "Arg Parser DOCA Flags" of the NVIDIA DOCA App Shield Agent Application 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 applications.
2. To build a given sample:

```
cd /opt/mellanox/doca/samples/doca_apsh/<sample_name>
meson build
ninja -C build
```

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

3. Sample (e.g., `apsh_libs_get`) usage:

```
Usage: doca_apsh_libs_get [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:
  -p, --pid                         Process ID of process to be analyzed
  -f, --vuid                        VUID of the System device
  -d, --dma                         DMA device name
  -s, --osty <windows|linux>        System OS type
```

   For additional information per sample, use the `-h` option:

```
./build/doca_<sample_name> -h
```

Chapter 5. Samples

5.1. Apsh Libs Get

This sample illustrates how to properly initialize DOCA App Shield and use its API to get the list of loadable libraries of a specific process.

The sample logic includes:

1. Opening DOCA device with DMA ability.
2. Creating DOCA Apsh context.
3. Setting and starting the Apsh context.
4. Opening DOCA remote PCI device via given vendor unique identifier (VUID).
5. Creating DOCA Apsh system handler.
6. Setting fields and starting Apsh system handler.
7. Getting the list of system process using Apsh API and searching for a specific process with the given PID.
8. Geting the list of process-loadable libraries using `doca_apsh_libs_get` Apsh API call.
9. Querying the libraries for 3 selected fields using `doca_apsh_lib_info_get` Apsh API call.
10. Printing libraries' attributes to the terminal.
11. Cleaning up.

References:

- `/opt/mellanox/doca/samples/doca_apsh/apsh_libs_get/apsh_libs_get_sample.c`
- `/opt/mellanox/doca/samples/doca_apsh/apsh_libs_get/apsh_libs_get_main.c`
- `/opt/mellanox/doca/samples/doca_apsh/apsh_libs_get/meson.build`
- `/opt/mellanox/doca/samples/doca_apsh/apsh_common.c`; `/opt/mellanox/doca/samples/doca_apsh/apsh_common.h`

5.2. Apsh Modules Get

This sample illustrates how to properly initialize DOCA App Shield and use its API to get the list of installed modules on a monitored system.

The sample logic includes:

1. Opening DOCA device with DMA ability.
2. Creating DOCA Apsh context.
3. Setting and starting the Apsh context.
4. Opening DOCA remote PCI device via given VUID.
5. Creating DOCA Apsh system handler.
6. Setting fields and start Apsh system handler.
7. Getting the the list of system-installed modules using `doca_apsh_modules_get` Apsh API call.
8. Querying the names of modules using `doca_apsh_module_info_get` Apsh API call.
9. Printing the attributes of up to 5 moduless attributes to the terminal.
10. Cleaning up.

References:

- `/opt/mellanox/doca/samples/doca_apsh/apsh_libs_get/apsh_libs_get_sample.c`
- `/opt/mellanox/doca/samples/doca_apsh/apsh_libs_get/apsh_libs_get_main.c`
- `/opt/mellanox/doca/samples/doca_apsh/apsh_libs_get/meson.build`
- `/opt/mellanox/doca/samples/doca_apsh/apsh_common.c`; `/opt/mellanox/doca/samples/doca_apsh/apsh_common.h`

5.3. Apsh Pslist

This sample illustrates how to properly initialize DOCA App Shield and use its API to get the list of running processes on a monitored system.

The sample logic includes:

1. Opening DOCA device with DMA ability.
2. Creating DOCA Apsh context.
3. Setting and starting the Apsh context.
4. Opening DOCA remote PCI device via given VUID.
5. Creating DOCA Apsh system handler.
6. Setting fields and starting Apsh system handler.
7. Getting the list of processes running on the system using `doca_apsh_processes_get` Apsh API call.
8. Querying the processes for 4 chosen attributes using `doca_apsh_proc_info_get` Apsh API call.
9. Printing the attributes of up to 5 processes to the terminal.
10. Cleaning up.

References:

- `/opt/mellanox/doca/samples/doca_apsh/apsh_pslist/apsh_pslist_sample.c`

- `/opt/mellanox/doca/samples/doca_apsh/apsh_pslist/apsh_pslist_main.c`
- `/opt/mellanox/doca/samples/doca_apsh/apsh_pslist/meson.build`
- `/opt/mellanox/doca/samples/doca_apsh/apsh_common.c`; `/opt/mellanox/doca/samples/doca_apsh/apsh_common.h`

5.4. Apsh Threads Get

This sample illustrates how to properly initialize DOCA App Shield and use its API to get the list of threads of a specific process.

The sample logic includes:

1. Opening DOCA device with DMA ability.
2. Creating DOCA Apsh context.
3. Setting and starting the Apsh context.
4. Opening DOCA remote PCI device via given VUID.
5. Creating DOCA Apsh system handler.
6. Setting fields and starting Apsh system handler.
7. Getting the list of system processes using Apsh API and searching for a specific process with the given PID.
8. Getting the list of process threads using `doca_apsh_threads_get` Apsh API call.
9. Querying the threads for up to 3 selected fields using `doca_apsh_thread_info_get` Apsh API call.
10. Printing thread attributes to the terminal.
11. Cleaning up.

References:

- `/opt/mellanox/doca/samples/doca_apsh/apsh_threads_get/apsh_threads_get_sample.c`
- `/opt/mellanox/doca/samples/doca_apsh/apsh_threads_get/apsh_threads_get_main.c`
- `/opt/mellanox/doca/samples/doca_apsh/apsh_threads_get/meson.build`
- `/opt/mellanox/doca/samples/doca_apsh/apsh_common.c`; `/opt/mellanox/doca/samples/doca_apsh/apsh_common.h`

5.5. Apsh Vads Get

This sample illustrates how to properly initialize DOCA App Shield and use its API to get the list of virtual address descriptors (VADs) of a specific process.

The sample logic includes:

1. Opening DOCA device with DMA ability.
2. Creating DOCA Apsh context.

3. Setting and start the Apsh context.
4. Opening DOCA remote PCI device via given VUID.
5. Creating DOCA Apsh system handler.
6. Setting fields and starting Apsh system handler.
7. Getting the list of system processes using Apsh API and searching for a specific process with the given PID.
8. Getting the list of process VADs using `doca_apsh_vads_get` Apsh API call.
9. Querying the VADs for 3 selected fields using `doca_apsh_vad_info_get` Apsh API call.
10. Printing the attributes of up to 5 VADs to the terminal.
11. Cleaning up.

References:

- `/opt/mellanox/doca/samples/doca_apsh/apsh_vads_get/apsh_vads_get_sample.c`
- `/opt/mellanox/doca/samples/doca_apsh/apsh_vads_get/apsh_vads_get_main.c`
- `/opt/mellanox/doca/samples/doca_apsh/apsh_vads_get/meson.build`
- `/opt/mellanox/doca/samples/doca_apsh/apsh_common.c`; `/opt/mellanox/doca/samples/doca_apsh/apsh_common.h`

5.6. Apsh Envars Get

This sample illustrates how to properly initialize DOCA App Shield and use its API to get the list of environment variables of a specific process.

Note: This sample works only on target systems with Windows OS.

The sample logic includes:

1. Opening DOCA device with DMA ability.
2. Creating DOCA Apsh context.
3. Setting and starting the Apsh context.
4. Opening DOCA remote PCIe device via given VUID.
5. Creating DOCA Apsh system handler.
6. Setting fields and starting Apsh system handler.
7. Getting the list of system processes using Apsh API and searching for a specific process with the given PID.
8. Getting the list of process envars using `doca_apsh_envars_get` Apsh API call.
9. Querying the envars for 2 selected fields using `doca_apsh_envar_info_get` Apsh API call.
10. Printing the envars attributes to the terminal.
11. Cleaning up.

References:

- `/opt/mellanox/doca/samples/doca_apsh/apsh_envars_get/apsh_envars_get_sample.c`
- `/opt/mellanox/doca/samples/doca_apsh/apsh_envars_get/apsh_envars_get_main.c`
- `/opt/mellanox/doca/samples/doca_apsh/apsh_envars_get/meson.build`
- `/opt/mellanox/doca/samples/doca_apsh/apsh_common.c`; `/opt/mellanox/doca/samples/doca_apsh/apsh_common.h`

5.7. Apsh Privileges Get

This sample illustrates how to properly initialize DOCA App Shield and use its API to get the list of privileges of a specific process.

Note: This sample works only on target systems with Windows OS.

The sample logic includes:

1. Opening DOCA device with DMA ability.
2. Creating DOCA Apsh context.
3. Setting and starting the Apsh context.
4. Opening DOCA remote PCIe device via given VUID.
5. Creating DOCA Apsh system handler.
6. Setting fields and starting Apsh system handler.
7. Getting the list of system processes using Apsh API and searching for a specific process with the given PID.
8. Getting the list of process privileges using the `doca_apsh_privileges_get` Apsh API call.
9. Querying the privileges for 5 selected fields using the `doca_apsh_privilege_info_get` Apsh API call.
10. Printing the privileges attributes to the terminal.
11. Cleaning up.

References:

- `/opt/mellanox/doca/samples/doca_apsh/apsh_privileges_get/apsh_privileges_get_sample.c`
- `/opt/mellanox/doca/samples/doca_apsh/apsh_privileges_get/apsh_privileges_get_main.c`
- `/opt/mellanox/doca/samples/doca_apsh/apsh_privileges_get/meson.build`
- `/opt/mellanox/doca/samples/doca_apsh/apsh_common.c`; `/opt/mellanox/doca/samples/doca_apsh/apsh_common.h`