



APPLICATION RECOGNITION

Reference Guide

Table of Contents

- Chapter 1. Introduction..... 1
- Chapter 2. System Design..... 2
- Chapter 3. Application Architecture..... 5
- Chapter 4. Configuration Flow..... 6
- Chapter 5. Running Application..... 8
- Chapter 6. References..... 10

Chapter 1. Introduction

Application Recognition (AR) allows identifying applications that are in use on a monitored networking node.

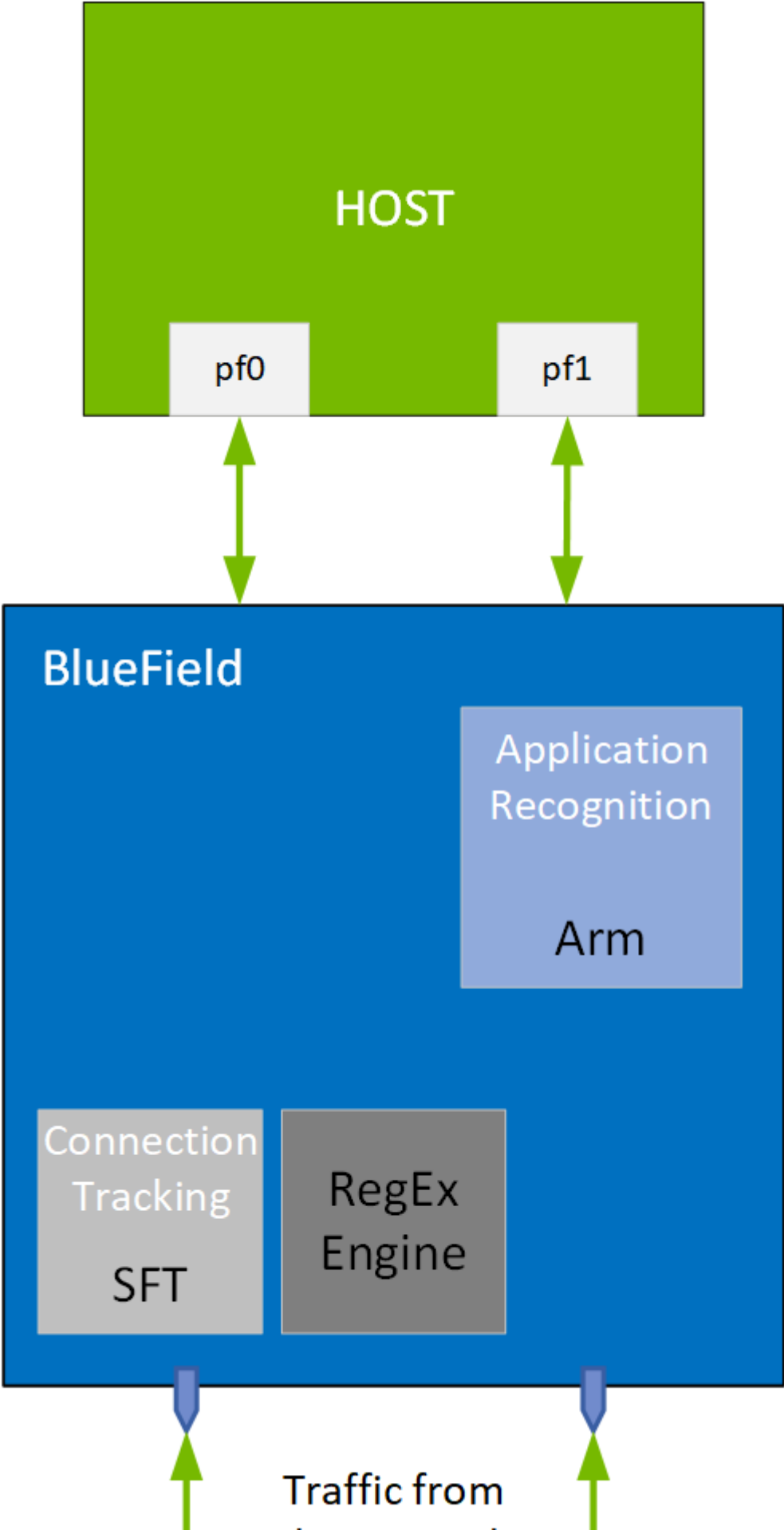
AR enables the security administrator to generate consolidated reports that show usage patterns from the application perspective. AR is also used as a corner stone of many security applications such as L7-based firewalls.

Due to the massive growth in the number of applications that communicate over Layer 7 (HTTP), effective monitoring of network activity requires looking deeper into Layer 7 traffic so individual applications can be identified. Different applications may require different levels of security and service.

This document describes how to build AR using the deep packet inspection (DPI) engine, which leverages NVIDIA® BlueField®-2 DPU capabilities such as regular expression (RXP) acceleration engine, hardware-based connection tracking, and more.

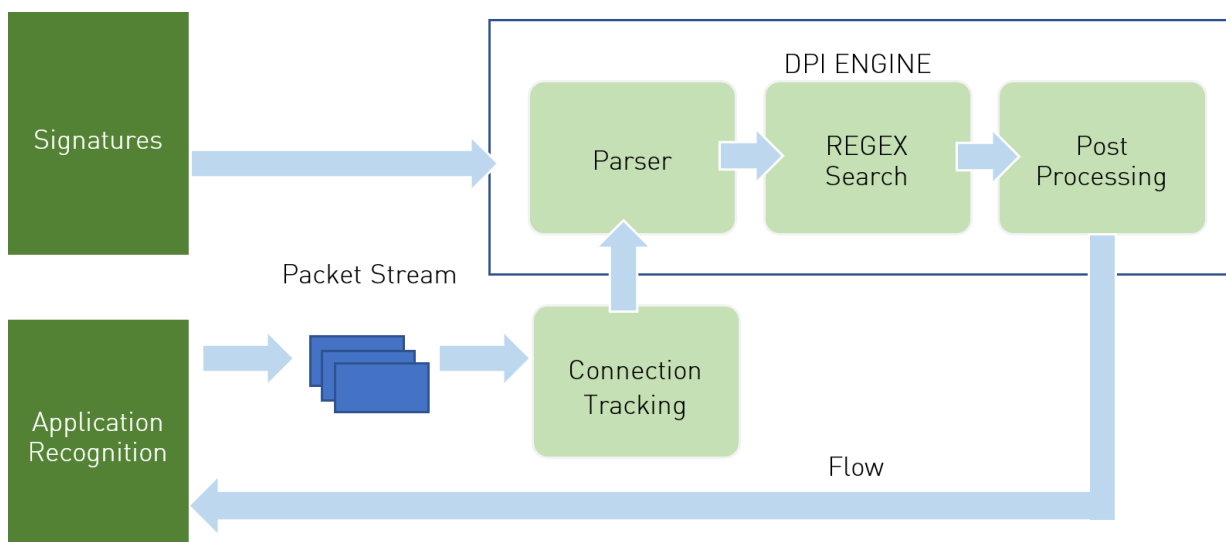
Chapter 2. System Design

The AR application is designed to run as "bump-on-the-wire" on the BlueField-2 instance, it intercepts the traffic coming from the wire, and passes it to the Physical Function (PF) representor connected to the host.



Chapter 3. Application Architecture

AR runs on top of Data Plan Development Kit (DPDK) based Stateful Flow Tracking (SFT) to identify the flow that each packet belongs to, then uses DPI to process L7 classification.



1. Signatures are compiled by DPI compiler and then loaded to DPI engine
2. Ingress traffic is identified using Connection Tracking module
3. Traffic is scanned against DPI engine compiled signature DB
4. Post processing is performed for match decision
5. Matched flows are identified and actions can be executed (Allow/Deny)

Chapter 4. Configuration Flow

1. DPDK initialization

```
dpdk_init(&argc, &argv, &nb_queues);
```

2. AR initialization

```
ar_init(argc, argv, cdo_filename, csv_filename);
```

- a). Initialize NetFlow using default configuration `/etc/doca_netflow.conf`.
- b). Initialize signature database.

3. Stateful Flow Table (SFT) and Port initialization

```
flow_offload_init(nb_queues);
```

- a). SFT initialization.
- b). Mempool allocation.
- c). Port initialization.

4. DPI initialization

```
dpi_ctx = doca_dpi_init(&doca_dpi_config, &err);
```

- a). Configure RegEx engine.
- b). Configure DPI queues.

5. Load compiled signatures to RegEx engine.

```
doca_dpi_load_signatures(dpi_ctx, ar_config.cdo_filename);
```

6. Configure DPI packet processing.

```
dpi_worker_lcores_run(nb_queues, CLIENT_ID, ar_worker_attr);
```

- a). Configure DPI enqueue packets.
- b). Send jobs to RegEx engine.
- c). Configure DPI dequeue packets.

7. Send statistics and write database.

```
sig_database_write_to_csv(ar_config.csv_filename);  
send_netflow();
```

- a). Send statistics to the collector.
- b). Write CSV file with signatures statistics.

8. AR destroy

```
ar_destroy(cmdline_thread, ar_config);
```

- ▶ Clear thread

9. DPI destroy

```
doca_dpi_destroy(dpi_ctx);
```


- ▶ Free DPI resources

Chapter 5. Running Application

- ▶ Prerequisites

Please refer to the [DOCA Installation Guide](#) for details on how to install BlueField related software.

- ▶ BlueField software image 3.6
- ▶ DOCA software package 1.0

- ▶ To build the application:

1. Prepare the environment variables (DPDK is installed under `/opt/mellanox`). Run:

- ▶ For Ubuntu:

```
export LD_LIBRARY_PATH=/opt/mellanox/dpdk/lib/aarch64-linux-gnu/
```

- ▶ For CentOS:

```
export LD_LIBRARY_PATH=/opt/mellanox/dpdk/lib64
```

2. The application recognition example is installed as part of the `doca-dpi-lib` package, the binary is located under `/opt/mellanox/doca/examples/ar/bin/doca_app_rec`. To re-build the application recognition sample, run:

- ▶ For Ubuntu:

```
export PKG_CONFIG_PATH=/opt/mellanox/dpdk/lib/aarch64-linux-gnu/pkgconfig/  
cd /opt/mellanox/doca/examples/ar/src  
meson /tmp/build  
ninja -C /tmp/build
```

- ▶ For CentOS:

```
export PKG_CONFIG_PATH=/opt/mellanox/dpdk/lib64/pkgconfig/  
cd /opt/mellanox/doca/examples/ar/src  
meson /tmp/build  
ninja -C /tmp/build
```

`doca_app_rec` will be created under `build/app`.

3. The application recognition example is a DPDK application. Therefore, the user is required to provide DPDK flags, and allocate huge pages. Run:

```
echo 1024 > /sys/kernel/mm/hugepages/hugepages-2048kB/nr_hugepages
```

- ▶ To run the application:

```
doca_app_rec [flags] -- --cdo [cdo_file] -output_csv [output_csv_file] -  
print_match
```

For example:

```
/opt/mellanox/doca/examples/ar/bin/doca_app_rec -a
0000:03:00.0,class=regex:eth,representor=[65535],sft_en=0 -- --c /usr/etc/doca/
dpi/ar/ar.cdo -o ar_stats.csv -p
```



Note: The flag `-a 0000:03:00.0,class=regex:eth,representor=[65535],sft_en=0` is a must for proper usage of the application. Modifying this flag will result unexpected behavior as only 2 ports (an uplink and a representor) are supported.

To print the output when the DPI engine finds a match, use `--print_match`.

For additional information about the available flags for DPDK use `-h` before the `-` separator. For the application, use `-h` after the `--`.

The application will periodically dump a `.csv` file with the recognition results containing statistics about the recognized apps in the format `SIG_ID, APP_NAME, MATCHING_FIDS, and DROP`.

As per the example above, a file called `ar_stats.csv` will be created.

Additional features can be triggered by using the shell interaction. This allows blocking and unblocking specific signature IDs using the following commands:

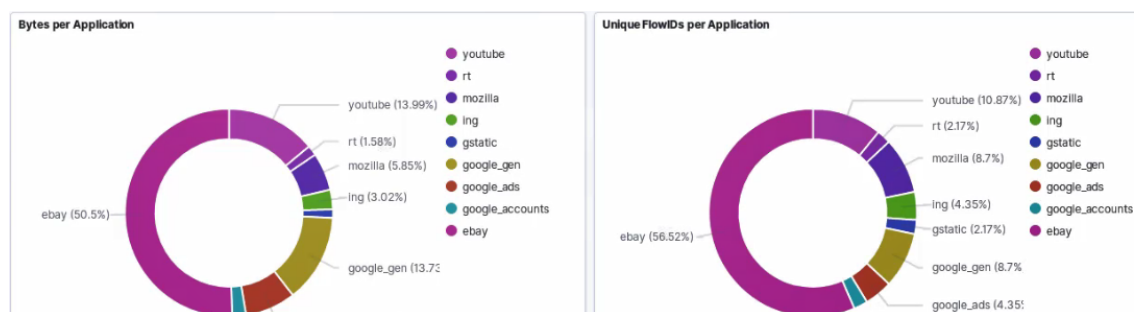
- ▶ `block <sig_id>`
- ▶ `unblock <sig_id>`

The `TAB` key allows autocompletion while the `quit` command terminates the application.

Application flags:

- ▶ `-c` or `--cdo <path>` – path to CDO file compiled from a valid PDD
- ▶ `-o` or `--output_csv <path>` – path to the output of the CSV file
- ▶ `-p` or `--print_match` – prints FID when matched in DPI engine
- ▶ `-i` or `--interactive` – adds interactive mode for blocking signatures
- ▶ `-n` or `-netflow` – exports data from BlueField to remote NetFlow collector

NetFlow collector UI example:



Chapter 6. References

▶ `/opt/mellanox/doca/examples/ar/src/ar.c`

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation nor any of its direct or indirect subsidiaries (collectively: "NVIDIA") make no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assume no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer ("Terms of Sale"). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer's own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer's sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer's product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

Trademarks

NVIDIA, the NVIDIA logo, and Mellanox are trademarks and/or registered trademarks of Mellanox Technologies Ltd. and/or NVIDIA Corporation in the U.S. and in other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

© 2021 NVIDIA Corporation. All rights reserved.