Stream4Flow: Real-time IP Flow Host Monitoring using Apache Spark

Tomas Jirsik*†
*Institute of Computer Science, †Faculty of Informatics
Masaryk University, Botanická 68a, 602 00 Brno, Czech Republic
E-mail: jirsik@ics.muni.cz

***Abstract*—In this paper, we present Stream4Flow, a framework for cyber situational awareness based on Apache Spark Streaming. We demonstrate utilization of Stream4Flow for real-time IP flow host monitoring in a large campus network. Contemporary IP flow analysis systems are not designed for the continuous host monitoring. Gaining the detailed overview of each host is not straightforward with these systems due to connection-based paradigm and performance challenges. We show that distributed stream processing is a natural solution for detailed IP flow host monitoring. Moreover, we describe a real-time host monitoring workflow in data streams in detail and present advantages of flow-based host monitoring in Apache Spark including real-time host profiling, dynamic level of detail and granularity.**

I. Introduction

Cyber situational awareness (CSA) aims to provide a complex understanding of a cyber system. IP flow monitoring is essential for achieving of CSA in computer networks. IP flows and associated monitoring infrastructure are designed to provide a holistic *macro view* over a network, which enables us to comprehend a network as a whole. Latest trends in CSA, however, show the necessity of providing not only macro view but also detailed *micro view* over a network [1], e.g. due to uprise of IoT. A security manager requires data on both a whole network and individual hosts in a network. The micro view, i.e. information on all hosts in a network, can be provided by IP flow monitoring to some extent, but it is not straightforward and faces several issues resulting from original purpose - holistic network monitoring.

Major issues of IP flow monitoring hindering a creation micro view over each host are:

- *Connection-based data paradigm* – data are provided per connection and stored into five-minute bins. A transformation to host-based view is time-consuming and includes aggregation and filtration for each bin of data.
- *Detection oriented analyses* – the goal of the majority of the analyses in IP flow monitoring is to identify a traffic of interest, e.g., malicious one. Considering this goal, we comprehend the traffic of interest, but we have only limited information about the other network traffic, which prevents us from a complex understanding of the network.
- *Analysis delay* – the majority of current IP flow monitoring tools analyses data in five-minute intervals. The information of hosts is then available with a five-minute aggregation. The aggregation level needs to be reduced as important behavior characteristics are lost due to the aggregation [2].

Our previous research [3] proved that distributed stream-based IP flow monitoring is a promising approach to achieve the micro view in CSA. This paper demonstrates the utilization of open-source stream processing Stream4Flow[1] framework for real-time IP flow monitoring in the large campus network. We briefly outline the framework architecture and focus on host monitoring workflow description along with advantages associated with our approach. We show that distributed stream-based approach can overcome the above-mentioned issues and provide the micro (and macro) view straightforward.

II. System Overview

Stream4Flow is a prototype open-source framework based on a general workflow for stream-based IP flow analysis described in [2]. The core of the framework is Apache Spark Streaming, a system for real-time high throughput data processing. Stream4Flow architecture is depicted on Fig. 1.

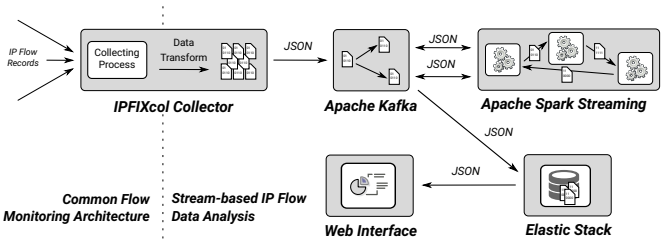

Fig. 1. Stream4Flow architecture [2].

Spark Streaming introduces Resilient Distributed Datasets (RDD), fault-tolerant collections of data elements in D-Streams, to achieve a sub-second latency in computations [4]. Such a solution in combination with its distributed nature, Map-Reduce programming principle, and integration with IP flow monitoring tools makes the detailed host monitoring at large scale possible even in real time.

In our approach, normalized IP flows are collected in message broker Apache Kafka. Kafka serves as a data stream producer for main application in Apache Spark. The host

978-1-5386-3416-5/18/$31.00 © 2018 IEEE

[1]https://stream4flow.ics.muni.cz

monitoring application takes advantage of distributed stream-based data processing and performs real-time transformation of connection-based information in IP flows to host-based view. First, the application reduces incoming data stream by filtering only the connections of interest (i.e. source IP address belonging to a monitored network). Second, data analysis windows for statistics are created. Next, the data stream is cleaned from unnecessary IP flow keys to contain only information relevant for analyses. During an analysis, data streams are multiplied so that several analyses run in parallel. This way of processing reduces the analysis time.

The transformation from the connection- to host-based view is done in each data stream by using map-reduce principle. Source IP address serves as a map key. The map is then reduced by the key and various statistics (e.g., number of flows) are computed for each IP address. The multiplied data streams are united into a single data stream containing host-based records. Next, a reduce operation is applied to the data stream to obtain a collection of statistics for a host. The data stream now contains one record of all computed statistics per host for a given analysis window. The data stream is passed back to Kafka. From there host-based records are stored in Elasticsearch or further analyzed for host profile generation.

III. Real-time IP Flow Host Monitoring

The demonstrated advantages of stream-based IP flow monitoring in Stream4Flow are following: *dynamic level of detail* and *granularity*, ability to create a *host profile*, and a possibility to enrich host profile with *additional types of information*.

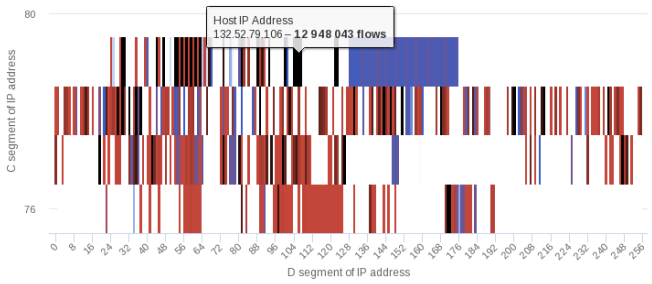

Fig. 2. Example of zoomed heatmap visualization of /16 networks.

To illustrate the dynamic level of view detail, we present a zoomable heat map for IP address space (see Fig. 2). The zoomable map allows us to focus both on a whole network and individual hosts. Axis of a heat map represent coordinates of particular CIDR network block. A zoom function allows us to visualize different CIDR blocks. Color scale of a map represents a characteristic of a given host or network, e.g., the number of flows, assigned administrator, etc. A network topology can be displayed using this map, e.g., by using same colors of hosts connected to the same switch. This map also serves as an access point to information on an individual host.

The real-time data processing enables us to acquire information of all hosts in a sub-second granularity. Less detailed statistics are available through statistics aggregation. Our application provides 10 s, 30 s, 1- and 5-minute data granularity for all host computed statistics. For each host, we compute a number of flows, packet, and bytes transferred, the average duration of a connection, a number of distinct peers and ports, and distribution of TCP flags. This information is instantly available in plots after a host is selected in a heat map (Fig. 3). Based on these statistics, we create a long-term profile of a host and define its expected behavior. A deviation from the expected behavior can be monitored and instantly highlighted via the heat map. Moreover, a host profile can be extended by log information received from central logging via Kafka.

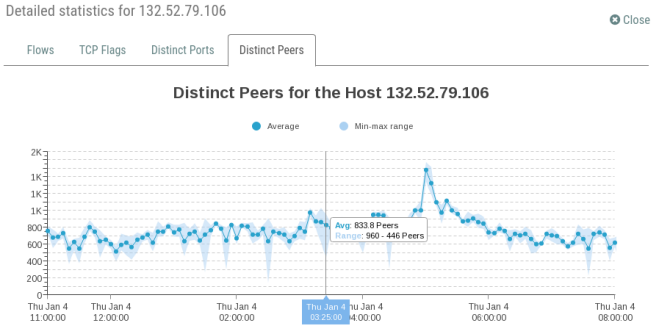

Fig. 3. Distinct peers statistics of a host.

IV. Summary

In this paper, we demonstrate a real-world implementation of real-time IP flow host monitoring. Described application overcomes the issues of traditional IP flow monitoring related to host monitoring and provides real-time cyber situational awareness. Our demonstration[2] displays a dynamic level of detail and granularity of several behavior characteristics computed from IP flows for each host in a monitored network.

Acknowledgment

This research was supported by the Security Research Programme of the Czech Republic 2015-2020 (BV III/1–VS) granted by the Ministry of the Interior of the Czech Republic under No. VI20162019014.

References

[1] A. Kott, C. Wang, and R. F. Erbacher, *Cyber Defense and Situational Awareness*, A. Kott, C. Wang, and R. F. Erbacher, Eds. Springer International Publishing, 2015, vol. 62. [Online]. Available: http://link.springer.com/10.1007/978-3-319-11391-3

[2] T. Jirsik, M. Cermak, D. Tovarnak, and P. Celeda, "Toward stream-based IP flow analysis," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 70–76, 2017.

[3] T. Jirsik and P. Celeda, "Toward Real-time Network-wide Cyber Situational Awareness using Apache Spark," in *Proceedings of the NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, 2018, To Appear.

[4] M. Zaharia, T. Das, H. Li, T. Hunter, S. Shenker, and I. Stoica, "Discretized streams: fault-tolerant streaming computation at scale," in *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles - SOSP '13*, no. 1. New York, New York, USA: ACM Press, 2013, pp. 423–438.

[2]Publicly available https://github.com/CSIRT-MU/Stream4Flow