

Flow Information Storage Assessment Using IPFIXcol

Petr Velan, Radek Krejčí

▶ **To cite this version:**

Petr Velan, Radek Krejčí. Flow Information Storage Assessment Using IPFIXcol. 6th International Conference on Autonomous Infrastructure (AIMS), Jun 2012, Luxembourg, Luxembourg. pp.155-158, 10.1007/978-3-642-30633-4_21 . hal-01529780

HAL Id: hal-01529780
https://inria.hal.science/hal-01529780

Submitted on 31 May 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Flow Information Storage Assessment Using IPFIXcol

Petr Velan and Radek Krejčí

CESNET, z.s.p.o.,
Zikova 4, 160 00 Prague, Czech Republic,
petr.velan@cesnet.cz,rkrejci@cesnet.cz

Abstract. Network monitoring has became a significant part of network management. Each environment and type of network have their specific, different needs. To allow network traffic monitoring in various environments, a necessity of flexible approach thus grows. The current generation of flow collectors provides only a limited flexibility, mainly due to limits of their data storage formats. Moreover, it is quite a challenging task to compare particular storage formats and their suitability for the specific environment and usage. In this paper we present IPFIXcol – a flow collector framework designed for easy data storage formats changing. This way, we plan to evaluate performance and suitability of various data storage formats for specific tasks. Results can be used to build the most appropriate data storage for the specific production environments.

Keywords: collector, flow, IPFIX, NetFlow, network monitoring.

1 Introduction

Flow traffic monitoring becomes widespread and demands for its additional features grow. Sometimes, we significantly miss some specific information in the stored flow records. The currently used flow records relate to IP networks only. However, flow information can be beneficial in non-IP networks too, like building automation and control system (BACS) networks or supervisory control and data acquisition (SCADA) networks. The additional information fields not included in standard flow records are needed for applications such as classification of network traffic based on time characteristics [1]. These needs can be satisfied by metering and collecting processes supporting the IP Flow Information Export (IPFIX) protocol [2].

In this paper, we are introducing IPFIXcol – IPFIX protocol collecting framework for receiving, processing and storing various flexible flow information. We are using this framework in our research to store the same flow information data in multiple formats. This way, the efficiency and suitability of each format can be compared. Based on the results, we plan to optimise current formats or to propose new ones to store network traffic flow information for various scenarios.

The paper is organized as follows. After the Introduction, some of the currently used IPFIX implementations and related works are summarized in Section 2. The IPFIXcol architecture is described in Section 3. Future use of the

framework for comparing and designing flow information storage formats is discussed in Section 4.

2 Related Work

A list of exporters supporting IPFIX protocol includes, for example, the FlowMon probe[1], nProbe[2], Vermont (VERsatile MONitoring Toolkit)[3] or YAF (Yet Another Flowmeter)[4]. IPFIX collector implementations are provided by nTop[5] toolkit, which comes from the same authors as nProbe, and the above-mentioned Vermont toolkit. There is also the python library called ripfix[6] that provides support for building IPFIX collecting and exporting applications. Results of the IPFIX implementations interoperability testing can be found in [3].

Our research is inspired by the comparison of nfdump[7] flat file format and MySQL database [4]. We will take the next step and provide a general comparison of more types of data storage formats along with complex benchmarks.

The research of some new possibilities in storing of streamed network data, such as network traffic flow information, is done by Fusco [5]. IPFIXcol provides framework for testing and results deployment of such a research while using the real network traffic data.

3 IPFIXcol System Architecture

The IPFIXcol is composed of a core program which is further extended by plugins (see Figure 1). Incoming flow data pass through an input plugin associated with a transport protocol. The output of the input plugin is parsed by the core to get some general information for further use by storage plugins. If specified, data can be modified by internal plugins. Storage plugins then store flow records in a specific data format. A more detailed description of each plugin type follows.

Input Plugins get the data from different flow information sources such as local files or network flow record streams from routers or standalone probes. The currently implemented input plugins provide support for an IPFIX file format [6] and IPFIX data transferred over SCTP, UDP, TCP or TLS over TCP protocols.

Internal Plugins extend functionality of the IPFIXcol core. With internal plugins, the IPFIXcol is able to serve as an IPFIX mediator [7] that can modify incoming data. Internal plugins can be used i.e. for data anonymization or flow-based sampling.

[1] `http://www.invea-tech.com/products-and-services/flowmon/flowmon-probes`
[2] `http://www.ntop.org/nprobe/nprobe-complies-with-ipfix-specification/`
[3] `http://vermont.berlios.de/`
[4] `http://tools.netsa.cert.org/yaf/`
[5] `http://www.ntop.org/`
[6] `http://ripfix.rubyforge.org/`
[7] `http://nfdump.sourceforge.net/`

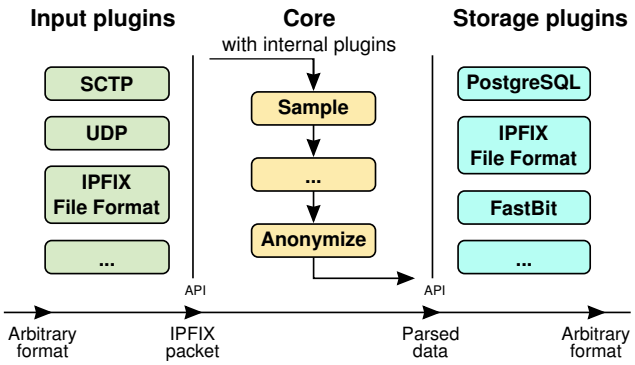


Fig. 1. System Architecture

Storage Plugins are used to process received data. This usually includes storing the flow records into files or databases. Furthermore, storage plugins can also resend the data to other destinations or process them to get overall statistics. Configuration of the plugins allows specifying rather complex cross connections between the flow information sources and the data stores.

Currently, the IPFIXcol storage plugins allow storing data in the IPFIX flat file format, PostgreSQL database and FastBit column-based database [8].

4 Conclusion and Future Work

This paper has introduced the IPFIXcol flow collecting framework. The IPFIXcol is already in a testing phase, including several basic input and output plugins.

The IPFIXcol is not just another IPFIX collector. It provides a framework for testing various procedures of flow information pre-processing or storage data formats. The IPFIXcol with an effective storage data format can be used for monitoring high-speed networks. By changing the storage data format or an internal processing plugin to support specific information fields, the IPFIXcol can be effectively used for monitoring research and non-standard environments such as building management or SCADA networks [9].

4.1 Future Work

To measure and compare characteristics of various flow data storage formats, we need to establish a set of test scenarios. We will take several queries typical of flow information data, such as data selection based on specified properties, data aggregation or combinations of both, so that the stored data are accessed sequentially or randomly. Then we will transform these queries into a specific format used by the data storage management tools.

New IPFIXcol storage plugins will be used to store the same data in different formats, so that our test queries return the same results. This way we can also

measure the throughput of different database solutions, which can be interesting in high-speed networks.

With these tests, we will be able to determine which type of data storage is most suitable for a specific environment and an application. We can also compare widely used nfdump flat file format, which is limited to the NetFlow v9, with more flexible databases that support IPFIX.

The acquired results will allow us to select and eventually improve the primary IPFIXcol storage format and its query tool. Currently, the usage of the FastBit library for storing network traffic flow information is very promising. The library is being developed at Lawrence Berkeley National Laboratory and we can contribute to it with specific demands from our use cases. Based on the test results, we might need to develop our own specific purpose database dedicated to storing flexible flow data.

Acknowledgement This work is supported by the "CESNET Large Infrastructure" project LM2010005 funded by the Ministry of Education, Youth and Sports of the Czech Republic.

References

1. Piskač P., Novotný J.: Using of Time Characteristics in Data Flow for Traffic Classification. In: Proceedings of the AIMS 2011. Nancy: Springer, 2011. p. 173–176, 4 pp. ISBN 978-3-642-21483-7.
2. Claise B.: Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information. RFC 5101, IETF (2008).
3. Trammell B.: DEMONS IPFIX Interoperability Event -- Final Report (2011). `www.ietf.org/proceedings/80/slides/ipfix-4.pdf`.
4. Hofstede R., Sperotto A., Fioreze T., Pras A.: The Network Data Handling War: MySQL vs. NfDump, In: 16th EUNICE/IFIP WG 6.6 Workshop, 2010, Trondheim, Norway. p. 167–176, 10 pp. ISBN 978-3-642-13970-3.
5. Fusco, F., Vlachos, M., Stoecklin, M.: Real-time creation of bitmap indexes on streaming network data, In The VLDB Journal, (2011).
6. Trammell B., Boschi E., Mark L., Zseby T., Wagner A.: Specification of the IP Flow Information Export (IPFIX) File Format. RFC 5655, IETF (2009).
7. Kobayashi A., Claise B., Muenz G., Ishibashi K.: IP Flow Information Export (IPFIX) Mediation: Framework. RFC 6183, IETF (2011).
8. Lawrence Berkeley National Laboratory. FastBit, `http://crd-legacy.lbl.gov/~kewu/fastbit/`.
9. Krejčí, R., Čeleda, P., Dobrovolný, J.: Traffic Measurement and Analysis of Building Automation and Control Networks. Paper to appear in AIMS 2012 (2012)