The Future of Network Flow Monitoring

Petr Velan
Masaryk University, Institute of Computer Science
Botanická 68a, 60200, Brno

velan@ics.muni.cz

Keywords. Network, Flow Monitoring, Security.

Abstract

Flow monitoring has been used for accounting and security for more than two decades. This paper describes how it was developed, what is its current status, and what challenges can be expected in this field in the following years.

1.1 The Past

The first mention of a flow export can be found in RFC 1272 [1] published in 1991 by IETF Internet Accounting (IA) Working Group (WG). The goal of the document was to provide background information on Internet accounting. The authors describe methods of metering and reporting network utilisation. The goal at the time was to provide a framework for traffic accounting. However, the common belief was that the internet should be free and any form of traffic capture, even for accounting purposes, is undesirable. This, together with the lack of vendor interest, resulted in the conclusion of the working group in 1993. Note that the negative attitude towards the monitoring returns more than 20 years later [2].

In 1995, Claffy, Braun, and Polyzos showed a methodology for internet traffic flow profiling based on packet aggregation [3], which started a revival of flow monitoring efforts. The Realtime Traffic Flow Measurement (RTFM) Working Group was active since 1996 and was concluded in 2000 by publishing several RFCs describing new traffic flow measurement framework with increased flexibility and even provided bi-directional flow support [4]. Since these documents fulfilled the objectives of the RTFM WG, the group was concluded in 2000. However, no flow export standard was developed as the vendors showed no interest in this area.

Meanwhile, Cisco realised that similar kind of flow information is already stored in a flow cache of their packet switching devices. The purpose of this cache is to speed up packet switching by making a forwarding decision only for the first packet of each flow. Unlike the RTFM flow measurement framework, the primary purpose of flow cache is not accounting nor monitoring. Therefore the configuration of the measurement process using a flow cache in a switch is severely limited. Despite the limitations, once Cisco introduced its flow export technology called NetFlow, it achieved widespread adoption. The main reason for the wide adoption was the fact that it was readily available on most Cisco devices with little effort. The NetFlow was patented in 1996, and the first version that became available to the general public around 2002 was NetFlow v5 [5], albeit Cisco newer released any official specification. The NetFlow v5 format simply specified a single set of fields that should be exported from each flow record.

NetFlow v5 was soon obsoleted by NetFlow v9 which remedied some of the deficiencies of the previous version. The state of NetFlow v9 is described in [6]. It allowed defining an arbitrary set of

fields for export using templates. It also introduced support for new protocols, such as IPv6, Virtual Local Area Networks (VLAN), Multiprotocol Label Switching (MPLS), Border Gateway Protocol (BGP) or Multicast.

Other vendors created their own versions of flow exporting protocols, although they retained some level of compatibility with NetFlow. There are JFlow by Juniper, CFlow by Alcatel-Lucent, RFlow by Ericsson, and other protocols. When the potential of flow monitoring for security purposes became realised in 2005 [7], more effort was devoted to extending flow records with information not directly associated with switching. Cisco presented Flexible NetFlow technology [8] in 2006, which allows to dynamically define and export new types of information, such as parts of payloads or traffic identification.

In 2001, it was clear that exporting flow information from switching devices was going to be supported by vendors. However, no standard flow export protocol existed at the time, and NetFlow v5 was not yet released to the general public. For that reason, the IETF started IP Flow Information Export (IPFIX) WG [9]. The charter was updated over the years to match current requirements. Several vendors were engaged in the IPFIX WG's activities, most notably Cisco, which significantly contributed from the start. The WG defined a set of requirements for the IPFIX protocol [10] and evaluated existing candidate protocols [11] to decide the most suitable approach to defining the new protocol. The NetFlow v9 specification (RFC 3954) was designed with IPFIX requirements in mind [12] and was released in order to compete in this evaluation (RFC 3955). After the evaluation, the NetFlow v9 was chosen as a basis of the new IPFIX protocol. For this reason, IPFIX is sometimes called NetFlow v10 and even starts with protocol version 10 in its header. However, the IPFIX protocol supports many new features and is not completely backwards compatible with NetFlow.

The IPFIX WG did more than just design the IPFIX protocol. In the 29 RFCs published before its conclusion, the WG paid attention to, for example bidirectional flow export, architecture for IP flow information export, reducing redundancy in flow and IP flow mediation framework. The IPFIX protocol specification is described by *"Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information"* [13] which became an Internet Standard. The working group was concluded in 2014; however, IPFIX related Internet-Drafts are still being created by involved parties. Further information about IPFIX development is provided by Brownlee in [14].

The importance of flow monitoring for security purposes was recognized by Cisco engineers in 2005 who proposed to use NetFlow for anomaly detection and traffic analysis [7]. Creation of dedicated flow monitoring probes allowed to easily extend the set of collected flow features and add the application information to the flows. Pioneers in this area were Cisco, ntop, Masaryk University, and CESNET. Applications such as HTTP, DNS, and SMTP were being analysed. Cisco published a tool called *joy* [15] in 2016 which allows collecting a rich set of information about network connections.

1.2 The Now

Concern for the privacy of users has been rising in recent years, which led to extensive deployment of encryption of network traffic. It is more and more difficult to monitor network applications as most traffic is protected by TLS of other encryption protocols. HTTP/2 is supported only together with encryption by mainstream browsers. A recent push for the addition of WireGuard VPN to Linux kernel has triggered its increasing adoption. However, despite the use of encryption, the need to monitor the traffic has not decreased. The challenge that we are facing is monitoring analysis of encrypted traffic.

Fortunately, machine learning algorithms are increasingly available as well; therefore, statistical analysis of encrypted data can be performed with relative ease. There is a large body of research on encrypted traffic classification and malware detection in encrypted traffic. The most recent results from Cisco show that information from TLS protocol together with per packet metrics can be used to achieve high accuracy in malware detection. However, flow records need to be extended with additional information to provide enough features for the machine learning algorithms.

1.3 The Future

The level of encryption can be only expected to grow. There is an RFC draft called *Encrypted Server Name Indication for TLS 1.3* which proposes to encrypt even Server Name Indication in TLS protocol. Combined with increasing deployment of DNS over TLS and DNS over HTTPS protocols, most of the current visibility into network traffic will soon be lost. This will result in higher demand for statistical analysis of network traffic.

To obtain accurate results for encrypted traffic classification, an annotated dataset of high quality is needed. There are two approaches to obtain such datasets. The first is to observe and capture normal network traffic and manually or semi-automatically annotate it. The second approach is to generate the traffic manually and label the observed traffic based on the known traffic patterns. However, both approaches are time-consuming and error-prone. Moreover, such datasets become obsolete in time and might not contain the necessary traffic mix that is seen in real networks. Therefore, most of the research should be focused on generating and obtaining datasets that will enable us to perform encrypted traffic classification with high accuracy.

A promising way to obtain such datasets is to combine information from multiple sources, such as DNS resolvers, server logs, and application logs. This will allow us to assign labels to flow data with high accuracy and create datasets that are both real and of high quality. Once the data sets are available, machine learning can be used to find correlations and relations in the data, which can be used to analyse even non-labelled traffic. However, masquerading network traffic as a different category is just the next step that attackers are likely to be examining.

Apart from the encrypted traffic classification, there is also the question of the quality of the generated data. For example, will the machine learning methods work well if flow generation parameters, such as timeouts, are changed? How are the flow exporters behaving under heavy load, are the exported flows incomplete? These and similar questions need to be answered, especially when machine learning is relied upon.

Paper origin

The first part of this paper has been accepted as a dissertation of the author.

Acknowledgment

This research was supported by ERDF "CyberSecurity, CyberCrime and Critical Information Infrastructures Center of Excellence" (No. CZ.02.1.01/0.0/0.0/16 019/0000822).

References

[1] C. Mills, D. Hirsh, and G.R. Ruth. *Internet Accounting: Background*. RFC 1272 (Informational). RFC. Fremont, CA, USA: RFC Editor, November 1991. URL: https://www.rfc-editor.org/rfc/rfc1272.txt (page 1).

[2] S. Farrell and H. Tschofenig. *Pervasive Monitoring Is an Attack*. RFC 7258 (Best Current Practice). RFC. Fremont, CA, USA: RFC Editor, May 2014. URL: https://www.rfc-editor.org/rfc/rfc7258.txt (page 1).

[3] Kimberly C. Claffy, Hans-Werner Braun, and George C. Polyzos. "A Parameterizable Methodology for Internet Traffic Flow Profiling". In: *IEEE Journal on Selected Areas in Communications* 13.8 (October 1995), pp. 1481–1494. ISSN: 0733-8716. DOI: 10.1109/49.464717 (page 1).

[4] N. Brownlee, C. Mills, and G. Ruth. *Traffic Flow Measurement: Architecture*. RFC 2722 (Informational). RFC. Fremont, CA, USA: RFC Editor, October 1999. URL: https://www.rfc-editor.org/rfc/rfc2722.txt (page 1).

[5] Cisco Systems, Inc., San Jose, CA and USA. *NetFlow Services Solutions Guide*. January 2007. URL: http://www.cisco.com/en/US/docs/ios/solutions_docs/netflow/nfwhite.html (Accessed on April 27, 2017) (page 1).

[6] B. Claise. *Cisco Systems NetFlow Services Export Version 9*. RFC 3954 (Informational). RFC. Fremont, CA, USA: RFC Editor, October 2004. URL: https://www.rfc-editor.org/rfc/rfc3954.txt (page 1).

[7] Cisco Systems, Inc., San Jose, CA and USA. *Cisco IOS NetFlow and Security*. February 2005. URL: http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6586/ps6642/prod_presentation0900aecd80311f49.pdf (Accessed on April 27, 2017) (page 2).

[8] Cisco Systems, Inc., San Jose, CA and USA. *Cisco IOS Flexible NetFlow*. December 2008. URL: http://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/flexible-netflow/product_data_sheet0900aecd804b590b.html (Accessed on April 27, 2017) (page 2).

[9] The Internet Engineering Steering Group. *IP Flow Information Export (ipfix) Charter*. URL: http://datatracker.ietf.org/wg/ipfix/charter/ (Accessed on April 27, 2017) (page 2).

[10] J. Quittek et al. *Requirements for IP Flow Information Export (IPFIX)*. RFC 3917 (Informational). RFC. Fremont, CA, USA: RFC Editor, October 2004. URL: https://www.rfc-editor.org/rfc/rfc3917.txt (page 2).

[11] S. Leinen. *Evaluation of Candidate Protocols for IP Flow Information Export (IPFIX)*. RFC 3955 (Informational). RFC. Fremont, CA, USA: RFC Editor, October 2004. URL: https://www.rfc-editor.org/rfc/rfc3955.txt (page 2).

[12] Brian Trammell and Elisa Boschi. "An Introduction to IP Flow Information Export (IPFIX)". In: *IEEE Communications Magazine* 49.4 (April 2011), pp. 89–95. ISSN: 0163-6804. DOI: 10.1109/MCOM.2011.5741152 (page 2).

[13] B. Claise, B. Trammell, and P. Aitken. *Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information*. RFC 7011 (Internet Standard). RFC. Fremont, CA, USA: RFC Editor, September 2013. URL: https://www.rfc-editor.org/rfc/rfc7011.txt (page 2).

[14] Nevil Brownlee. "Flow-Based Measurement: IPFIX Development and Deployment". In: *IEICE Transactions on Communications* E94.B.8 (September 2011), pp. 2190–2198. DOI: 10.1587/transcom.E94.B.2190 (page 2).

[15] Cisco. *Joy*. URL: https://github.com/cisco/joy/ (Accessed on May 25, 2019) (page 2).