Application-Aware Flow Monitoring

Petr Velan, Pavel Čeleda
Institute of Computer Science
Masaryk University
Brno, Czech Republic
{velan,celeda}@ics.muni.cz

***Abstract*—Network flow monitoring has been a part of network security for the last dozen years. It is constantly evolving to keep pace with changes in network operation and innovative network attacks. The thesis contributes to the continuous efforts by exploring the possibilities unlocked by extending the flow data with application-specific information. We show how the construction of flows is affected by processing of application data, present the benefits to traffic analysis, and assess the inevitable performance loss caused by additional data processing. To compensate for the lost performance, several novel optimisation techniques are proposed for the flow monitoring process. Recognising that the increasing deployment of encryption is going to limit the benefits of application flow monitoring, we perform a survey of methods for measurement of encrypted traffic. The thesis is concluded by an outlook towards future possibilities for flow monitoring advancement.**

***Index Terms*—network, monitoring, measurement, flow, application flow, NetFlow, IPFIX, encryption, performance, 100 Gbps**

I. Introduction

The concept of network flow monitoring was proposed in 1991 to facilitate accounting of network usage. Cisco's NetFlow became a de-facto standard for acquiring IP network and operational data in the following years. However, the main potential of flow monitoring became realised much later, in 2005, when Cisco engineers proposed to use NetFlow for anomaly detection and traffic analysis [18]. Flows records together with packet capture are the two main sources of data for intrusion detection systems nowadays. Moreover, the flow records are being used for data retention [19], which is mandatory for internet service providers in many countries.

Due to the growing cybercrime industry and cyber espionage [20] the traffic analysis and security potential of flow monitoring have become more accentuated over time. To support traffic analysis, Cisco enriched its Flexible NetFlow [21] with information from the Network-Based Application Recognition (NBAR) [22] in 2009. NBAR was initially used for QoS management on Cisco appliances. The trend of enriching flow records with information from application layer continued and resulted in the application-aware flow monitoring, which can be seen as a combination of Deep Packet Inspection (DPI) and flow monitoring. A deployment of flow monitoring has become standard practice since almost every enterprise networking equipment is able to export flow records nowadays. Steinberger et al. surveyed ISPs and network operators and found out that a majority of them uses flow monitoring for attack detection in their networks [23].

978-3-903176-15-7

The importance of flow monitoring is growing. As the speed of network links increases, DPI-based intrusion detection systems are becoming incapable of handling the sheer amount of traffic. Moreover, the increasing amount of encryption makes it harder still for DPI to be effective. The goal of the thesis [1] is to advance flow monitoring techniques to achieve better application visibility and performance. These techniques are primarily intented to improve network management and security.

When the flow monitoring is deployed on a network, the measured flow data are sent to a flow data processing system. The system facilitates flow analysis, reporting, and threat detection. All of these functions require high-quality flow data for their operation. For example, when some of the packets are not monitored or actively sampled, the quality of flow data is significantly reduced [24]. Moreover, when the data contains artifacts [25], the data analysis and threat detection can be impaired as well.

Maintaining a high-quality flow monitoring system is a challenging task due to the constant changes in traffic structure and increasing volume [26]. We have identified three main topics that directly affect flow monitoring and flow data analysis. Firstly, the flow monitoring must keep pace with the increasing speed of networks. Therefore, the performance of flow monitoring must be studied and improved to match the speed of the network links. Secondly, as the network attacks are becoming more sophisticated, application layer information must be provided to enable more efficient threat detection. Lastly, the increasing amount of encrypted traffic makes application visibility difficult. Therefore, to maintain any degree of application visibility, novel approaches for monitoring of encrypted traffic must be explored.

A. Application Layer Information

When we started our research in flow monitoring in 2012, application visibility in flow monitoring was a relatively new concept. With the exception of Flexible NetFlow utilising NBAR, the flow records contained only network and transport layer information. However, even the Flexible NetFlow provided only application recognition without extraction of any application-specific data. At the end of 2011 ntop released a version of their nProbe flow exporter capable of utilising OpenDPI library to provide information about application pro-

tocol [27]. Still, the information provided was only application protocol name and identifier, which was very similar to what the Flexible NetFlow provided.

With the increasing number of discovered vulnerabilities in applications [28], we have found it essential to increase the capabilities of flow monitoring to detect more of these vulnerabilities. Therefore, we have decided to create true application-aware flow monitoring that would allow threat detection algorithms to utilise not only network and transport layer information, but also application layer information as well.

B. Growing Network Speeds

The implementation of flow monitoring started as an additional feature of routers and switches. However, as the main purpose of the networking devices is not flow monitoring, the quality of data is compromised under high load, where the networking capabilities are of higher importance than the monitoring itself. This, and the reason that the devices provided only limited configuration options lead to the development of flow monitoring on commodity hardware [29]. It was shown that even monitoring of a Gigabit Ethernet on commodity hardware is a challenging problem. Therefore, even with increasing CPU frequency and the number of cores, flow monitoring of 10 Gbps and faster networks requires the use of special techniques and optimisations. The recent advances in networking technologies lead to standardisation and deployment of 40 Gbps and 100 Gbps Ethernet links. Moreover, standards for 200 Gpbs and 400 Gbps Ethernet were approved at the end of 2017.

We have decided to research the possibilities of high-speed flow monitoring at these speeds. Moreover, since application-aware flow monitoring requires more performance than basic flow monitoring, we study the impact of processing of application payloads on the flow monitoring performance.

C. Traffic Encryption

Our decision to include application layer information in flows to increase network visibility and aid threat detection was based partially on the amount of unencrypted traffic that could be observed in the network. Research showed that only one-third of web pages could be browsed via HTTPS [30] in 2013. However, the amount of encrypted traffic steadily increased, and more than 70 % of web pages are loaded over HTTPS nowadays [31]. This massive change in the use of encryption necessitates the use of novel approaches to monitoring of the encrypted traffic. Therefore, a study of encrypted protocols, as well as an overview statistical methods of traffic classification, are needed to analyse the impact of encryption on the amount of information that can be provided by the flow monitoring.

II. Application Flow Monitoring

We have described the aspects of extending flow monitoring by using information from the application layer. We have argued that application monitoring is necessary to provide

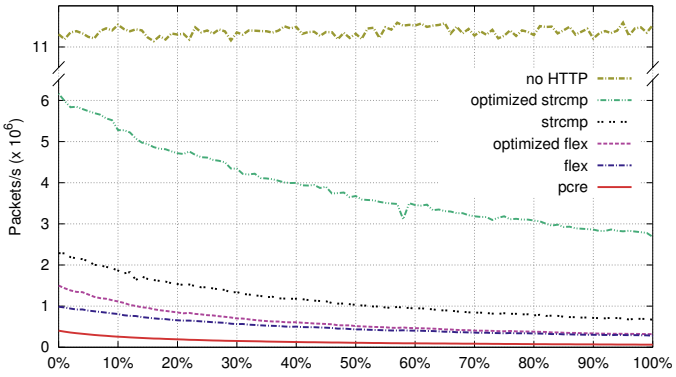

Fig. 1. Parser Performance Comparison with Respect to HTTP Proportion (0 % - No HTTP, 100 % - Only HTTP Headers) in the Traffic - Full Packets 1500 B.

information about current network-based cybersecurity threats. Without the application insight, attackers can perform application layer attacks that have no impact visible using the basic flow monitoring. Therefore the application flow monitoring utilises aspects of DPI to provide more fine-grained information about the observed traffic.

We have surveyed the state-of-the-art of the application parsers creation process. There are several approaches that allow creating application parsers from a higher level description, which allows creating more robust and reliable parsers. However, existing application flow exporters do not often use these approaches and implement their own application parsers. Moreover, only a few flow exporters provide real application visibility, even though most of them support application identification.

The existing sources do not use consistent terminology regarding the application flow monitoring. Therefore, we have proposed a workable terminology that captures the current state of the art in the application flow monitoring. We call the flow monitoring without processing the application layer *basic flow monitoring* and differentiate between *IP flow monitoring* which is a term used for any flow monitoring that uses information from the IP layer. We have provided definitions of both *application flow* and *application flow record*, and explained their relation to the flow and flow record definitions provided in [2].

Once the terminology of application flow has been established, we ventured to describe the application flow monitoring process, particularly the changes that it introduced to the general flow monitoring process, which we described in detail as well. It impacts the flow creation on two main levels. Firstly, a new flow expiration reason has been introduced that can be triggered due to an application event. Secondly, the flow records became application flow records as they contain information from the application layer. These changes have an impact on the number of generated flow records as well as on their sizes.

Another contribution is a discussion of a design of an HTTP protocol parser [3]. We have shown several approaches to the

construction of the parser, implemented them, and evaluated their performance. We have quantified the flow monitoring performance drop caused by each of them and showed that a hand-crafted parser can be highly optimised to provide the best throughput, although it requires an expert programmer to implement it correctly. Performance comparison of the chosen approaches is shown in Figure 1.

III. Flow Monitoring Performance

Performance of flow monitoring systems is another topic covered in the thesis. We have explained different methods of measuring the overall performance and have shown that it is important to choose the one that provides more appropriate results. Different factors that have an impact on the flow monitoring were considered as well. We have shown that the used hardware, system settings, packet capture framework, settings of the flow exporter, and the traffic mix have a considerable impact on the results of the measurement. CPU processing power and memory controller throughput are the most obvious bottlenecks for flow monitoring. We have shortly discussed ways to detect these bottlenecks and decide whether the memory footprint or CPU cycles need to be optimised.

The performance of flow monitoring significantly depends on the performance of the underlying packet capture framework. We have presented the evolution of the state-of-the-art packet capture frameworks. Linux NAPI can be used for packet capture on gigabit networks, however, to achieve full packet capture on 10 Gbps and faster networks, specialised NIC drivers together with kernel network stack bypass must be used. Zero-copy approach and receive side scaling are a necessity to achieve these speeds. The buffers to which the NIC delivers packets through DMA transfers are mapped directly to the user-space. Interrupts are usually disabled at high speeds and extensive polling is performed by the receiving application. When the data needs to be shared between multiple applications, the RX queues are managed by the kernel driver. In this case, the provided API must enable reception of multiple packets at once, a technique called batch or burst packet processing, to avoid unnecessary context switches between kernel and user-space. In case only a single user-space application is processing the packets, such as in the DPDK framework, the RX queues are managed directly by user-space, which increases the performance due to the lesser number of context switches.

Although packet capture is an important part of the flow monitoring process, many optimisation techniques can be applied to the flow monitoring as well. An overview of discussed techniques is provided in Table I. We have discussed the possibility of hardware acceleration using specialised FPGA-based NICs that allows offloading of packet preprocessing to the NIC itself. We have shown that depending on the capabilities of the NIC, different levels of offloading can be achieved. However, the flow exporter must be aware of these capabilities and actively cooperate with the NIC. Multiple optimisations can be used in the design of the flow exporter software as well. We have argued that multithreading and

TABLE I
Flow Monitoring Acceleration Techniques.

Hardware acceleration	Software acceleration
Receive Side Scaling	Multithreading
Packet trimming	NUMA awareness
*Packet header preprocessing**	*Flow state in parsers**
Flow processing offloading	Flow cache design
*Application identification**	Per-flow expiration timeout
	Delayed packet processing
	Bidirectional flow records

**Novel Proposals Are Highlighted*

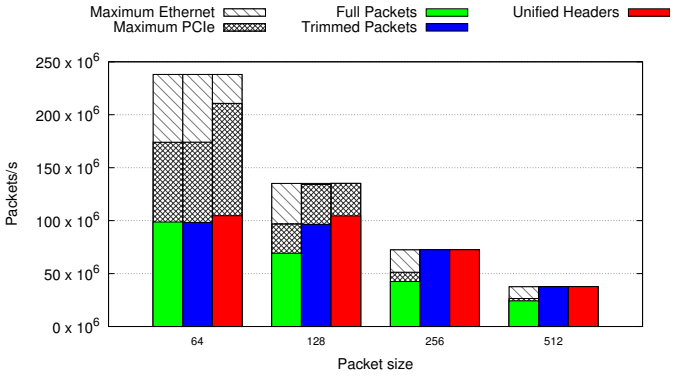

Fig. 2. Packet Processing Performance Comparison in Packets/s for 16,384 Flows per Interface.

NUMA awareness are key to flow monitoring performance, although special care needs to be taken to configure the setup correctly. Other optimisation techniques such as flow cache design, per-flow expiration timeouts, delayed packet parsing, and the use of biflows have been considered as well.

Finally, we have built a high-density flow monitoring system capable of processing 16x10 Gbps [4]. The system utilised two FPGA-based NICs that allowed us to test the offloading of packet processing. Figure 2 compares the performance of processing of full packets, trimmed packets, and pre-processed packet headers. A significant performance improvement achieved using the offloading techniques can be observed. Even 512 B packets cannot be processed without offloading at line rate; however, offloading allows us to process the same amount of traffic for 256 B packet size. Therefore, these techniques are crucial for processing traffic on high-speed networks.

The number of concurrent flows that needed to be kept in a flow cache has had a considerable impact on the measured performance. Moreover, we also demonstrated the importance of having powerful enough CPU by repeating the tests with two different CPUs. A 100 Gbps flow monitoring system with a single NIC was demonstrated in [5]. We believe that by utilising more of the proposed optimisation techniques, we should be able to achieve even higher throughput of the system, possibly reaching even 200 Gbps rate offered by the latest NICs available on the market.

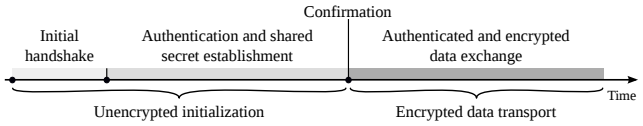

Fig. 3. A General Scheme of Network Security Protocols.

IV. Measurement of Encrypted Traffic

To address the problem of measurement of encrypted traffic, we have presented an overview of current approaches for the classification and analysis of encrypted traffic [6]. First, we selected a number of the most widely used encryption protocols and described their packet structure and standard behaviour in a network. Second, we focused on information which is provided by encryption protocols themselves. Figure 3 shows how the encrypted connection setup is usually performed by most encryption protocols. We have found that the initiation phase often provides information about the protocol version, ciphers used, and the identity of at least one communicating party. Such information can be used to monitor and enforce security policies in an organisation. We also discovered that the use of information from the unencrypted parts of an encrypted connection for a network anomaly detection is only briefly investigated by researchers. Information about communicating parties can be leveraged to discern the type of encrypted traffic. For example, the list of supported cipher suites provided by a client when establishing a secure connection can help to identify the client. We believe that the use of unencrypted parts from the initiation of an encrypted connection should be explored in more detail.

Before starting the analysis of the encrypted network traffic, it is necessary to identify it. Thus, we surveyed approaches to classifying and analysing encrypted traffic in journals, conference papers, proceedings of specialised workshops, and technical reports. We studied works presented in selected computer science journals, mainly the Communications Surveys & Tutorials, Computer Networks, International Journal of Network Management, and Transactions on Network and Service Management. We also surveyed international conferences such as IMC, PAM, CISDA, CNSM, IM, and NOMS over the period 2005-2014. There are payload-based methods, which use knowledge of a packets' structure, and feature-based methods, which use characteristics specific to the protocol flow. For the payload-based classification, there are several open-source traffic classifiers which can identify encrypted traffic using pattern matching. The initiation of communication often has a strictly defined structure; therefore, the patterns can be constructed for specific protocols. The main difference between various classifiers is that some of them require traffic from both directions of the communication to correctly classify the flows.

Feature-based traffic classifiers have been intensively researched over the last decade. Many statistical and machine-based learning methods have been applied to the task of traffic classification. Despite this, there are no conclusive results to show which method has the best properties. The main reason is that the results depend heavily on the datasets used and the configuration of the methods. We have applied the multilevel taxonomy of Khalife et al. [32] and categorised the existing methods. Our results show that most of the authors use private datasets, sometimes in combination with public ones. For this reason, the individual results are not directly comparable. Most of the methods use supervised or semi-supervised machine learning algorithms to classify flows and even determine the application protocol of a given flow. Most methods target encryption protocols, such as SSH, SSL/TLS, and encrypted BitTorrent, and use similar methods. However, there are also some novel works which apply innovative approaches to refine the classification up to deriving the content of the encrypted connections.

Most authors of feature-based classification methods claim that their approach is privacy sensitive as it does not require the traffic payload. However, privacy issues are much wider. In 2013, the Cyber-security Research Ethics Dialog & Strategy Workshop [33] started a discussion about the influence of cyber-security research on the privacy of Internet users. Researchers need to keep in mind that their research activities have a significant impact on infrastructure security, network neutrality, and privacy of end users.

V. Relevance and Impact

This section presents several results based on our research of the application flow monitoring. Apart from these results, we believe our research to be beneficial to anyone interested in deploying flow monitoring and analysing flow data. Ideas from our research are already applied in products of Flowmon Networks a.s., which is an university spin-off company focused on flow monitoring.

We performed an analysis of HTTP traffic in a large-scale environment which uses application flow monitoring with HTTP protocol processing [7]. In contrast to previously published analyses, we were the first to classify patterns of HTTP traffic which are relevant to network security. We described three classes of HTTP traffic which contain brute-force password attacks, connections to proxies, HTTP scanners, and web crawlers. Using the classification, we were able to detect up to 16 previously undetectable brute-force password attacks and 19 HTTP scans per day in our campus network. The activity of proxy servers and web crawlers was also observed. Symptoms of these attacks may be detected by other methods based on basic flow monitoring, but detection using the analysis of HTTP requests is more straightforward. We, thus, confirm the added value of application flow monitoring in comparison to the traditional method.

The exhaustion of IPv4 address space increases pressure on network operators and content providers to continue the transition to IPv6. The IPv6 transition mechanisms such as Teredo and 6to4 allow IPv4 hosts to connect to IPv6 hosts. On the other hand, they increase network complexity and render many methods ineffective to observe IP traffic. In [8], we

extended our flow-based measurement system to involve transition mechanisms information to provide full IPv6 visibility. Our traffic analysis focused on IPv6 tunnelled traffic and used data collected over one week in the Czech national research and education network. The results expose various traffic characteristics of native and tunnelled IPv6 traffic, among others the TTL and HOP limit distribution, geolocation aspect of the traffic, and list of Teredo servers used in the network. Furthermore, we showed how the traffic of IPv6 transition mechanisms has evolved between 2010 and 2013.

The importance of IP address geolocation has increased significantly in recent years, due to its applications in business advertisements and security analysis, among others. Current approaches perform geolocation mostly on-demand and in a small-scale fashion. As soon as geolocation needs to be performed in real-time and in high-speed and large-scale networks, these approaches are not scalable anymore. To solve this problem, we proposed two approaches to large-scale geolocation in [9]. Firstly, we presented an exporter-based approach, which added geolocation data to flow records in a way that is transparent to any flow collector. Secondly, we presented a flow collector-based approach, which added native geolocation to NetFlow data from any flow exporter. After we had presented prototypes for both approaches, we demonstrated the applicability of large-scale geolocation by means of use cases. Our prototypes have shown to be scalable enough for deployment on the 10 Gbps Internet connection of the Masaryk University.

Performing research on live network traffic requires the traffic to be well documented and described. The results of such research are heavily dependent on the particular network. The paper [10] presents a study of network characteristics, which can be used to describe the behaviour of a network. We proposed a number of characteristics that can be collected from the networks and evaluate them on five different networks of Masaryk University. The proposed characteristics cover IP, transport, and application layers of the network traffic. Moreover, they reflect strong day-night and weekday patterns that are present in most of the networks. Variation in the characteristics of the networks indicates that they can be used for the description and differentiation of the networks. Furthermore, a weak correlation between the chosen characteristics implies their independence and contribution to the network description.

VI. Future Directions

We have proposed three novel concepts for application flow monitoring as a part of the thesis. Each of our proposals could be implemented separately; however, we believe that a combination of all proposed approaches is not only possible but would provide the greatest benefits for the application flow monitoring.

The first proposed concept is EventFlow [11]. It is an extension of the application flow monitoring which allows preserving relations between HTTP and DNS application flows that are a part of single user action, most typically browsing a web page. We have described an architecture of the EventFlow extension and its limitations. A prototype implementation of the EventFlow has been introduced and evaluated on a packet trace from an ISP network. We have shown that a significant number of flow records can be recognised as a part of a single user action.

MetaFlow has been proposed as an approach to monitoring of encapsulated traffic. We have argued that the current practice of extending flow records with information from each layer is not flexible and has several disadvantages. MetaFlow aims to create a hierarchy of encapsulated flows so that each flow is able to retain a simple structure while the information about the relations between flows is preserved. We have also discussed how to create a unique flow identifier, which is necessary for building the MetaFlow tree structure.

The last proposal has expanded the MetaFlow concept to application layer events. By decoupling application events from basic flow records, we can significantly simplify both flow creation process and flow data processing systems. Moreover, it allows us to solve issues produced by the forceful splitting of flows based on application payload. We believe that separating application events from basic flow monitoring is the future of application flow monitoring. Furthermore, this approach provides new opportunities for novel research not only for flow monitoring but also for flow data processing systems.

Apart from the directions described in the thesis, we plan to focus on the quality of the generated flow data. Different approaches and configurations influence the data and study of the impact on methods using the data is highly significant.

Publications

[1] Petr Velan. “Application-Aware Flow Monitoring”. Doctoral thesis, Dissertation. Masaryk University, Faculty of Informatics, Brno, 2018. URL: https://is.muni.cz/th/a2fxd/.

[2] Petr Velan. “Improving Network Flow Definition: Formalization and Applicability”. In: *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*. April 2018, pp. 1–5.

[3] Petr Velan, Tomáš Jirsík, and Pavel Čeleda. “Design and Evaluation of HTTP Protocol Parsers for IPFIX Measurement”. In: *Advances in Communication Networking*. Vol. 8115. Heidelberg: Springer Berlin Heidelberg, 2013, pp. 136–147. ISBN: 978-3-642-40551-8.

[4] Petr Velan and Viktor Puš. “High-Density Network Flow Monitoring”. In: *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. May 2015, pp. 996–1001.

[5] Viktor Puš et al. “Hardware Accelerated Flow Measurement of 100 Gb Ethernet”. eng. In: *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. Ottawa, Canada: IEEE Xplore Digital Library, May 2015, pp. 1147–1148.

[6] Petr Velan et al. “A Survey of Methods for Encrypted Traffic Classification and Analysis”. In: *International Journal of Network Management* 25.5 (2015), pp. 355–374. DOI: 10.1002/nem.1901.

[7] Martin Husák, Petr Velan, and Jan Vykopal. “Security Monitoring of HTTP Traffic Using Extended Flows”. In: *2015 10th International Conference on Availability, Reliability and Security*. August 2015, pp. 258–265.

[8] Martin Elich et al. “An Investigation Into Teredo and 6to4 Transition Mechanisms: Traffic Analysis”. In: *IEEE 38th Conference on Local Computer Networks Workshops (LCN Workshops)*. Sydney, Australia: IEEE Xplore Digital Library, October 2013, pp. 1046–1052. ISBN: 978-1-4799-0540-9.

[9] Pavel Čeleda et al. “Large-Scale Geolocation for NetFlow”. In: *IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*. Ghent, Belgium: IEEE Xplore Digital Library, May 2013, pp. 1015–1020. ISBN: 978-1-4673-5229-1.

[10] Petr Velan et al. “Network Traffic Characterisation Using Flow-Based Statistics”. In: *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*. April 2016, pp. 907–912.

[11] Petr Velan. “EventFlow: Network Flow Aggregation Based on User Actions”. In: *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*. April 2016, pp. 767–771.

[12] Luuk Hendriks et al. “Flow-Based Detection of IPv6-specific Network Layer Attacks”. In: *Security of Networks and Services in an All-Connected World: 11th IFIP WG 6.6 International Conference on Autonomous Infrastructure, Management, and Security, AIMS 2017, Zurich, Switzerland, July 10-13, 2017, Proceedings*. Cham: Springer International Publishing, 2017, pp. 137–142. ISBN: 978-3-319-60774-0.

[13] Luuk Hendriks et al. “Threats and Surprises behind IPv6 Extension Headers”. In: *2017 Network Traffic Measurement and Analysis Conference (TMA)*. IEEE Xplore Digital Library, June 2017, pp. 1–9.

[14] Petr Velan. “Practical Experience with IPFIX Flow Collectors”. In: *IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*. Ghent, Belgium: IEEE Xplore Digital Library, May 2013, pp. 1021–1026. ISBN: 978-1-4673-5229-1.

[15] Petr Velan and Pavel Čeleda. “Next Generation Application-Aware Flow Monitoring”. English. In: *Monitoring and Securing Virtualized Networks and Services*. Vol. 8508. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2014, pp. 173–178. ISBN: 978-3-662-43861-9.

[16] Petr Velan and Radek Krejčí. “Flow Information Storage Assessment Using IPFIXcol”. In: *Dependable Networks and Services*. Vol. 7279. Lecture Notes in Computer Science. Heidelberg: Springer Berlin Heidelberg, 2012, pp. 155–158. ISBN: 978-3-642-30632-7.

[17] Petr Velan, Husák Martin, and Daniel Tovarňák. “Rapid Prototyping of Flow-Based Detection Methods Using Complex Event Processing”. In: *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*. April 2018, pp. 1–3.

Acknowledgment

The publication of this paper and the follow-up research was supported by the ERDF "CyberSecurity, CyberCrime and Critical Information Infrastructures Center of Excellence" (No. CZ.02.1.01/0.0/0.0/ 16_019/0000822).

References

[18] 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).

[19] Ward van Wanrooij and Aiko Pras. “Data on Retention”. In: *Proceedings of the 16th IFIP/IEEE Ambient Networks International Conference on Distributed Systems: Operations and Management*. DSOM’05. Barcelona, Spain: Springer-Verlag, 2005, pp. 60–71. ISBN: 978-3-540-29388-0.

[20] Center for Strategic and International Studies. *The Economic Impact of Cubercrime and Cyber Espionage*. July 2013. URL: http://csis.org/files/publication/60396rpt_cybercrime-cost_0713_ph4_0.pdf (Accessed on March 2, 2018).

[21] 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).

[22] Cisco Systems, Inc., San Jose, CA and USA. *Network Based Application Recognition (NBAR)*. URL: https://www.cisco.com/c/en/us/products/ios-nx-os-software/network-based-application-recognition-nbar/index.html (Accessed on March 2, 2018).

[23] Jessica Steinberger et al. “Anomaly Detection and Mitigation at Internet Scale: A Survey”. In: *Proceedings of the 7th IFIP WG 6.6 International Conference on Autonomous Infrastructure, Management, and Security: Emerging Management Mechanisms for the Future Internet - Volume 7943*. AIMS’13. Barcelona, Spain: Springer-Verlag, 2013, pp. 49–60. ISBN: 978-3-642-38997-9.

[24] Daniela Brauckhoff et al. “Impact of Packet Sampling on Anomaly Detection Metrics”. In: *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*. IMC ’06. Rio de Janeriro, Brazil: ACM, 2006, pp. 159–164. ISBN: 1-59593-561-4.

[25] Rick Hofstede et al. “Measurement Artifacts in NetFlow Data”. In: *Passive and Active Measurement*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 1–10. ISBN: 978-3-642-36516-4.

[26] Cisco Systems, Inc., San Jose, CA and USA. *Cisco Visual Networking Index: Forecast and Methodology, 2016–2021*. June 2017. URL: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf (Accessed on March 2, 2018).

[27] ntop. *Unveiling Application Visibility in ntop and nProbe (both in NetFlow v9 and IPFIX)*. November 2011. URL: http://www.ntop.org/nprobe/unveiling-application-visibility-in-ntop-and-nprobe-both-in-netflow-v9-and-ipfix/ (Accessed on September 27, 2017).

[28] Yves Younan. *25 Years of Vulnerabilities: 1988-2012*. 2013. URL: https://courses.cs.washington.edu/courses/cse484/14au/reading/25-years-vulnerabilities.pdf (Accessed on March 2, 2018).

[29] Luca Deri. “Passively Monitoring Networks at Gigabit Speeds Using Commodity Hardware and Open Source Software”. In: *Proceedings of the Passive and Active Measurement Conference*. 2003, pp. 1–7.

[30] Nevena Vratonjic et al. “The Inconvenient Truth About Web Certificates”. In: *Economics of Information Security and Privacy III*. New York, NY: Springer New York, 2013, pp. 79–117. ISBN: 978-1-4614-1981-5.

[31] Internet Security Research Group (ISRG). *Let’s Encrypt Stats*. March 2018. URL: https://letsencrypt.org/stats/ (Accessed on March 2, 2018).

[32] Jawad Khalife, Amjad Hajjar, and Jesus Diaz-Verdejo. “A multilevel taxonomy and requirements for an optimal traffic-classification model”. In: *International Journal of Network Management* 24.2 (2014), pp. 101–120. ISSN: 1099-1190. DOI: 10.1002/nem.1855.

[33] CAIDA. *Cyber-security Research Ethics Dialog & Strategy Workshop*. May 2013. URL: http://www.caida.org/workshops/creds/1305/ (Accessed on February 22, 2018).