Privacy-Aware Intrusion Detection in High-Speed Backbone Networks - Design and Prototypical Implementation of a Multi-Layered NIDS

Mario Golling[1], Robert Koch[1] and Gabi Dreo Rodosek[1]

Abstract: Network Intrusion Detection Systems are nowadays an integral part of network security. NIDS provide a layer of defense by monitoring and analyzing the traffic for signs of suspicious activities and alerting system administrators when potential violations are detected. In the context of large, high-speed networks, such as backbone networks, we make two observations: Firstly, devices capable of detecting intrusions on high-speed links of 10 Gbps and higher are rather expensive or must be built based on complex arrays. Secondly, legislation commonly restricts the way in which backbone network operators are allowed to analyze data. As a consequence, traditional Intrusion Detection approaches, where individual packets are scanned for suspicious patterns (commonly referred to as Deep Packet Inspection) are not applicable. Although Flow-Based Intrusion Detection offers several advantages in terms of processing requirements, the aggregation of packets into flows obviously entails a loss of information. To bridge the gap, within a previous publication, we have proposed a multi-layered approach that combines the advantages of both types of Intrusion Detection. The first layer comprises Flow-Based Intrusion Detection, making a pre-selection of suspicious traffic. Additional Packet-Based Intrusion Detection is subsequently performed on the pre-filtered packet stream to (i) facilitate in-depth detection, (ii) avoid the problem of a costly infrastructure, (iii) obey to the various legal barriers on network traffic inspection and to (iv) reduce the oftentimes high false alarm rate of Flow-Based Intrusion Detection Systems. Within this publication, we refer to our previously develop concept of Multi-Layered Intrusion Detection and extend it by demonstrating the corresponding prototype and a practical evaluation. As such. we shortly recall the underlying concept and then present a prototypical implementation and evaluate it with real data.

Keywords: Network Security, Intrusion Detection, High-Speed Networks, Flow-Based Intrusion Detection, Multi-Layered Intrusion Detection, Legal Inspection

1 Introduction

Attacks on computer systems are constantly rising throughout the last couple of years, both in terms of quality and quantity. Especially when attacks are performed in a distributed manner, their devastating power can have a significant impact on Backbone Providers, as for example seen in early 2013 when the Spamhaus project was targeted by Distributed Denial of Service (DDoS) attacks with more than 300 Gbps - enough to overload several Internet exchanges. In addition to DDoS attacks, in particular large scale activities (primarily worms and botnets) are also of special relevance for high-speed network operators, since they also consume a great amount of resources (e.g., see [Ar14, GHK14]).

[1] Munich Network Management Team (MNM-Team), Universität der Bundeswehr München, Werner-Heisenberg-Weg 39, D-85577 Neubiberg, {mario.golling, robert.koch, gabi.dreo}@unibw.de

As a result of the increase in the importance of defending against IT attacks, Intrusion Detection, *the process of monitoring the events occurring in a computer system or network and analyzing them for signs of possible incidents, which are violations or imminent threats of violation of computer security policies, acceptable use policies, or standard security practices* [SM07], is constantly gaining attention. Focused on *identifying possible incidents, logging information about them, attempting to stop them, and reporting them to security administrators*, Intrusion Detection Systems (IDSs) are nowadays an integral part of any IT defence strategy and as such a necessary addition to the security infrastructure of nearly every organization [SM07].

As presented in more detail in Section 2, the balance between the demands of High-Speed Backbone Providers (especially high-speed links, high throughput, the need for a low false alarm rate, a particular interest in certain *large-scale* attack types and, derived from legal standards, the necessity to stay inlined with data protection requirements) on the one hand and the capabilities of IDSs (reliable detection in high-speed backbone environments in compliance with legal requirements is currently not possible or only very limited requiring a high financial effort resp. specific situations / constellations) on the other hand is *not* in line. To improve this situation, Section 3 firstly illustrates our conceptual design. Based on this, Section 4 then implements a corresponding Proof-of-Concept (PoC), before an evaluation with the help of real-world data is presented in Section 5. Finally, Section 6 concludes this paper, giving a final summary as well as an outlook.

2 Background

Especially in relation to the subject of this publication, Intrusion Detection in High-Speed Backbone Networks, Network-Based IDSs (NIDSs), *which monitor network traffic for particular network segments or devices and analyze the network and application protocol activity to identify suspicious activity* [SM07] are of high relevance, since their counterpart, Host-Based IDSs (HIDSs), *which monitor the characteristics of a single host and the events occurring within that host for suspicious activity* [SM07] are not applicable due to the fact that High-Speed Backbone Providers are normally used to forward data and as such are neither the source nor the destination of the data.

Next to the distinction of IDSs in HIDSs and NIDSs, other subdivision can be made as well (e.g., see Debar et al. [DDW00]). Traditional Intrusion Detection approaches rely on the inspection of individual packets, often referred to as **Deep Packet Inspection (DPI)**, where individual packets are scanned for signs of suspicious activities; some of the more popular IDSs that make use of this approach are Snort, Suricata, Bro (*open source*) or from vendors such as Cisco or McAffee. However, high link speeds/throughputs, especially in backbone networks, seriously constrain this approach. With link speeds of more than 40 Gbps, DPI-Based IDSs are either very expensive (e.g., because they have to built based on complex arrays) or not even available. Furthermore, legislation, especially in the European Union, seriously restricts the use of DPI and often requires a solid legal justification *before* DPI can be applied.

To overcome both constraints, **Flow-Based Intrusion Detection** can be applied. A flow is defined as *a set of IP packets passing an observation point in the network during a certain* *time interval; all packets belonging to a particular flow have a set of common properties* and focuses primarily on packet header fields and packet characteristics. Consequently, compared to DPI-Based Intrusion Detection, Flow-Based IDSs have to handle a considerable smaller amount of data, which is of advantage in terms of privacy and link speed (allowing to perform a detection in high-speed environments). Given that flow export technologies, such as NetFlow and IPFIX, aggregate packets into flows, such an IDS is usually capable of monitoring the aggregated traffic using commodity hardware. Next to this, flow export technologies are nowadays embedded in the vast-majority of high-end packet forwarding devices (Routers and Switches) and are already widely used for network management, so deploying Flow-Based IDSs comes at almost no cost which in turn makes this approach economically attractive.

However, this entails *three major drawbacks*, which currently exclude Flow-Based IDSs from their large-scale practical usage:

1. *The number of recognizable attacks is severely restricted.* In comparison to DPI-based IDSs a decision has to be made on the basis of an evaluation of a significantly lower amount of data which particularly excludes the detection of payload-based attacks except a volume-based detection wrt. abnormal behavior of the "amount" of payload.
2. *The detection time is increased.* Normally multiple flows are generated on the network components. Thereafter, these flows are exported to a Flow Collector in a certain interval before then, in the third step, an analysis can be carried out by a Flow-Based IDSs.
3. *The number of false alarms increases (partly).* Based on the fact that relatively little information can be analyzed, for certain classes of attacks, no unambiguous distinction between benign and malignant traffic can be made. Although this does not apply to all classes of attacks, it can be said that in comparison with a DPI-Based IDS, a Flow-Based IDS has a higher percentage of false positives.

As shown in Figure 1, Flow-Based IDSs are capable of detecting *those attacks* that are of high relevance for a backbone network operator. On the other hand, quite a number of different approaches are available, each of them addressing specific aspects (see [Sp10]).

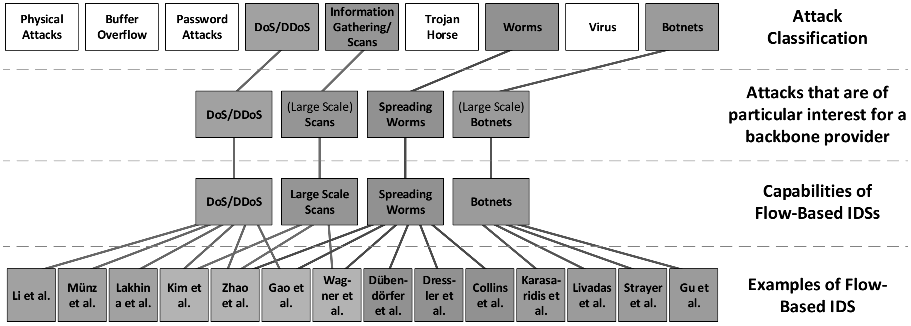

Fig. 1: Capabilities of Flow-Based IDSs vs Special Interests of Backbone Providers wrt. Intrusion Detection [GHK14]

Many of these approaches are either of theoretical nature and/or focus on very specific

cases (for example, the recognition of (non-large-scale) attacks in SSH connections which - with regard to Backbone Providers - is of secondary importance).
Nevertheless, with regard to (i) the three major drawbacks and (ii) Backbone Providers, it has to be noted, that the first two arguments (restricted number of recognizable attacks and increased detection time) are of lower importance. In addition, the compliance of Flow-Based IDSs with data privacy regulations as well as their low prize has to be emphasized, too. However, due to argument number 3 (low accuracy), Flow-Based IDSs are marginally used.

3 Design

In this Section, the design of the Multi-Layered Architecture is presented briefly. In order to be widely deployable, the architecture deliberately makes use of existing state-of-the art approaches. The main components, together with their interactions, are shown in Figure 2; the architecture is a slightly simplified version of the one already presented in [GHK14] and [GKS14] to allow for (i) a smoother introduction to the topic and (ii) to perform an appropriate implementation and evaluation. In particular, the implementation / evaluation as such is *the* main added value of this publication.

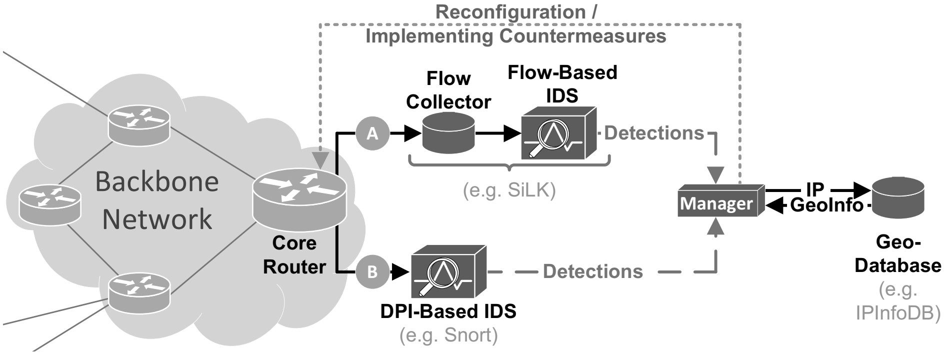

Fig. 2: Architecture for Privacy-Aware Intrusion Detection in High-Speed Backbone Networks

Components
The architecture makes use of both (i) an Flow-based IDS *and* (ii) an DPI-Based IDS. While the Flow-based IDS constantly analyzes the entire data traffic, the DPI-based IDS is activated by the Manager to investigate / analyze exactly this part of the suspicious traffic - and not more. As such, the Manager controls the data-streams, and activates / configures the DPI-based IDS. To make sure that the DPI-Based IDS receives the optimal data-stream, the Manager can reconfigure the Core Router (Reconfiguration). Upon detection of an attack, the Manager can also reconfigure (Implementing Countermeasures) the Router to drop the attack traffic (e.g., by using an Access Control List / creating a sinkhole, by load balancing requests to multiple facilities or by informing the Administrator about the suspicious traffic).

Geolocation

The knowledge of the geographical origin of an attack may also be of great importance. As for example the security company Mandiant has revealed in a study [Ma13], a large part of *all* attacks on US authorities originate from a single building in Shanghai / China. As a result, within the architecture, geo-localization (also called Geolocation), the mapping of a logical address, here the IP of a host, with the physical respectively geographic location is used, too. To this end, the architecture uses a Geolocation Service based on a Geolocation Database which is constantly updated. In particular if, due to the lack of performance of the DPI-Based IDS, not all incidents of the Flow-Based IDS can be investigated, Geolocation (next to the already existing weighting of the Flow-Based IDS) is used as an additional distinguishing feature to differentiate between the important and the less important incidents.

In this work, Geolocation is used in two ways: Static Geo-Reputation and Dynamic Geo-Correlation. In analogy to e-mail white-and blacklisting, IP addresses obtain a specific score (ip scoring mechanism), within the *Static Geo-Reputation* process. Especially for organizations with which there is a special relationship of trust, whitelisting is conceivable. Here, this implies that the probability of a large-scale attack starting from e.g., another provider who is known to have very high security standards is reduced (but not equal to zero). On the other hand, also the idea of blacklisting seems to be generally applicable to Intrusion Detection. Organizations / providers that are known to have a low level of security obtain a higher scoring.

In contrast to the static assignment of IP addresses in those that tend to be benign IP addresses (whitelist) and those that tend to be malignant (blacklisting), a correlation of IPs (to other IPs known to be malicious) can also be established dynamically (*Dynamic Geo-Correlation*). Starting from the origin of an incident and by taking into account historic data, the degree of similarity can also be defined. The Dynamic Geo-Correlation approach of this paper is build upon different presumptions, which are:

1. *Temporal correlation.* After a certain time, a Geo-Correlation shall no further be given; e.g., after a certain timer has expired.
2. *Spatial correlation.* A new connection only gets correlated to an already known, malicious connection if the distance is lower than a specific threshold.
3. *Fog of time.* The older the information (IP known to be malicious in the past), the less important shall this information be; as a consequence, the distance in which a Geo-Correlation is assumed is decreasing over time.
4. *Accuracy level.* The more accurate the Geolocation is (which is also closely linked to the density of the population), the smaller the distance for Geo-Correlation and vice versa; hence, the less precise the Geolocation, the greater the distance for Geo-Correlation.

4 Implementation

Following one of the underlying objectives of our research - to combine as many as possible *already available solutions* in an appropriate manner - for the implementation, well-established solutions are used. Referring to this, the examples of Figure 2 - written in grey - correspond to the solutions used. To this end, the authors would like to point out in particular three aspects:

1. *Flow Collector/Flow-Based IDS*. Although - as already mentioned - in theory, a wide variety of solutions is available for the analysis of flows (see [Si15] for more details), unfortunately only a fraction of these solutions are also available as a real product / software solutions. In the open source community, there are just a few popular tools [Ma09]: nfdump (and its web component nfSen), SiLK (System for Internet-Level Knowledge, which is developed by Carnegie Mellon) and Stager (a system for aggregating and presenting network statistics).
   For the following reasons, SiLK [CE16] has been chosen:

   - All open source projects offered much better opportunities to make own modifications in comparison to commercial tools such as IBM Aurora, NetQoS Reporter Analyzer, Caligare Flow Inspector, and Arbor Peakflow (see also [Ma09]).

   - Stager, a representative of open source software, with the last update on July 8th, 2010 seems to be no longer maintained and is therefore also not considered.

   - In choosing between nfdump / nfsen and SiLK, SiLK was finally the choice, firstly because it was preferred in several web discussions (see [Op12] resp. [Op14]) and, secondly, it turned out to be better suited for the relevant use case as well as the extensions needed.

2. *DPI-Based IDS*. Here, the choice is comparatively easy. Snort, as the de facto state-of-the art in the field of open source Intrusion Detection is our solution of choice.
3. *Geo-Database*. In terms of Geolocation, we also have a state-of-the art solution in use, in this case the IPInfoDB with the use of the corresponding API.

In the first step, we have set up a new virtual machine (VirtualBox) with Ubuntu 15.10. Using a virtual machine was based on the fact that we thus are able to quickly switch between different computers and consequently are able to run this VM on a high-performance machine at a later point in time. Inside the VM, both IDSs were installed. For the actual prototype, the Manager, C was used. As Section 5 will carry out in more detail, for reasons of traceability and optimization no *live evaluation* was performed. Instead, within the first step of the evaluation, live data traffic was recorded using special hardware accelerated network interface card (here with the use of the so-called HANIC Appliance from Invea-Tech; see e.g., [Vi14] for more information about the appliance). This has the advantage that external influences (changes in traffic patterns etc.) can be excluded and / or are kept constant (ceteris paribus) and we (i) obtain a better comparison of systems / versions / parametrizations and (ii) are able to compare individual states of development of our prototype in a better way. Consequently, the communication between the Manager and the Router was not implemented with the PoC.

5 Evaluation

5.1 Evaluation Setup

As previously mentioned, (i) to simplify the evaluation and (ii) to reduce external effects as well as to (iii) to allow a better comparison of individual systems, traffic data was recorded in advance with the aid of special hardware. To this end, at the uplink of the University of Twente (40 GBit/sec uplink of the university with their parent ISP), all incoming as well as outgoing data stream was collected and recorded in September 2014, comprising a period of several days. Accordingly no filtering has been applied. Special focus has been placed on obtaining a *representative data stream*. Therefore, in advance, the authors were especially driven from the following considerations: (i) The data stream was deliberately recorded over several days (including weekends); (ii) it has been ensured that external unusual effects such as semester breaks were reduced to a minimum.

In the next step, we have enriched the data set of the University of Twente to (i) have a more solid and representative portfolio of attacks and (ii) to increase the throughput to the level that is typical for a Backbone Provider. The latter was necessary because, despite the fact that the university has a connection of 40 Gbit/s, this connection was - on average - only used at a relatively low level (low throughput). The use of synthetic attacks was also needed to have a good and typical portfolio of attacks. Here, we have especially made use of the publications of Arbor Networks (e.g., see [Ar14]).

In order to know the Ground Truth (within the scope of this PoC), we have then analyzed the original data set of Twente (slowly) with the use of (several) DPI-based IDSs for the presence of attack traces. Although the authors know that by doing so most likely not all attacks are revealed, for the lack of alternatives, the results obtained hereby are used as (relative) Ground Truth. Concerning the synthetic portion (background noise and synthetic attacks), of course, we have taken the real values as baseline.

With the help of two additional virtual machines (both also use Ubuntu 15.10) and a virtual switch (where next to the two traffic generators the evaluation machine was connected) the actual evaluation took place.

The sampling rate we used was 1-in-100 (frequency in which the packets are analyzed; on average, 1 in every 100 packets is captured and analyzed) with a polling-interval of 30 sec (timeframe in which the network device exports Flows to the collector); these settings are essentially based on a compromise of [PP09, bl15] and [nM15].

5.2 Findings

Due to brevity, we would like to discuss here especially the practical example of a Distributed Denial of Service attack. Figure 3 shows an appropriate excerpt.

Subsequently, we assume that the following factors are necessary for a positive identification:

1. A positive initial identification by the Flow-Based IDS,

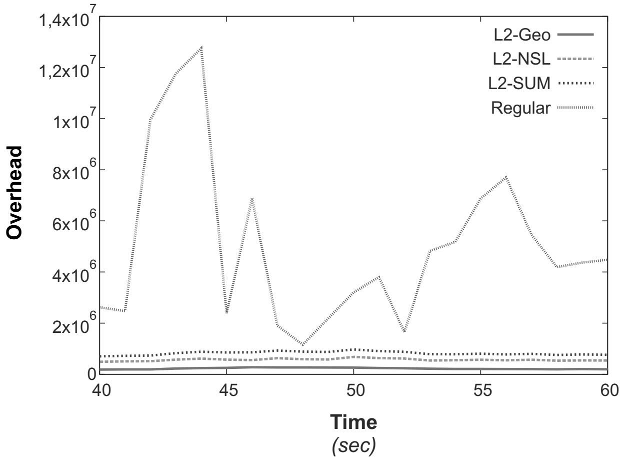

Fig. 3: Resource consumption of a DDoS; L2-Geo represents the Geolocation-time; L2-NSL refers to the NSLOOKUP-Time, L2-SUM corresponds L2-Geo + L2-NSL and Regular represents the regular overhead of the DPI-Based IDS

2. A precise geographical knowledge of both communication partners and other information (in order to consider possible contractual limitations - in particular, possible contractually guaranteed non-blocking agreements); for this, within the scope of the prototype, both NSLOOKUP resp. DIG, *and* the Geolocation of IPINFODB is used and all traffic (i) from/to Spain resp. (ii) governmental organization of the Netherlands is *never* blocked,
3. A positive verification from the DPI-Based IDS; this is particularly necessary to reduce the false alarm rate of the Flow-Based IDS. This is mainly due to the fact that the objective of a Backbone Provider is not to identify as many attacks as possible (as Intrusion Detection is not in the main focus of a Backbone Provider). Instead, a Backbone Provider rather likes to reduce the false alarm ratio to the absolute minimum possible.

Figure 3 and 4 are to be interpreted as follows: At the beginning of the detection of a DDoS attack, the overhead is relatively high, because new connections are initiated from different places. This results in a high overhead for both IDSs, but especially for the DPI-based IDS (see Figure 3 - regular). In spite of this overhead, with regard to Geolocation and NSLOOKUP the overhead is comparatively moderate, but also comprises quite high fluctuations (see Figure 4).

The reason why the overhead in Figure 4 decreases with time is because after a certain time, no new lookups are needed anymore. The real added value of the architecture is in particular (i) in the area of privacy and (ii) of maintaining contractually guaranteed obligations (in particular the non-blocking of certain communication relationships). In comparison to the direct use of a DPI-Based IDS without the Flow-Based IDS, the data load has been reduced with a factor of approx. one third, and - this is one of the main advantages - also guarantees

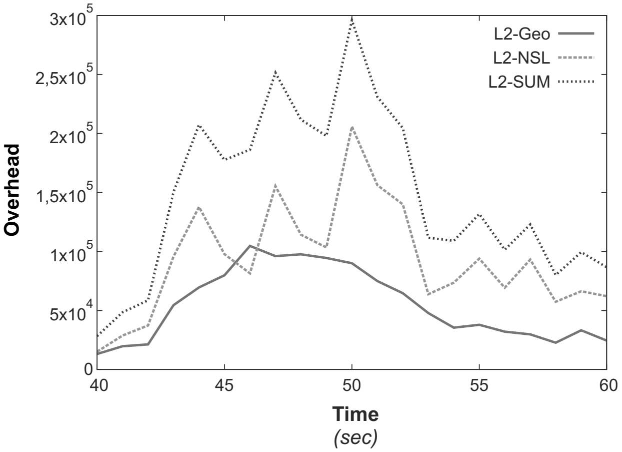

Fig. 4: Resource consumption of a DDoS; L2-Geo represents the Geolocation-time; L2-NSL refers to the NSLOOKUP-Time, L2-SUM corresponds L2-Geo

that data protection requirements are met. The use of Geolocation and NSLOOKUP also ensures a better applicability of existing business contracts.

6 Conclusions and Outlook

In this paper, we have presented an architecture for privacy-aware Multi-Layered Intrusion Detection, which aims at (i) reducing costs by being deployable on commodity hardware, (ii) overcoming legal limitations with respect to traffic analysis (a clear motivation in terms of a Flow-Based IDS alert is needed *before* DPI is performed) and (iii) compliance with specific aspects such as the contractually guaranteed non-blocking of data traffic of certain contractors. To this end, a generic architecture has been defined and a first implementation was realized. With the help of real-world traffic, the main benefits of the architecture were shown, too.

For the future, amongst other things, we plan to investigate in more detail: First, we like to examine the influence of the sampling rate; e.g. by using a sampling of 1:1 (i.e., everything is sampled), or 1:50. Second, we like to look for ways and means to consider the contracts between Backbone Provider and costumer in more detail. Third, we are trying to improve Intrusion Detection through inter-domain exchange of knowledge of attacks, both between "trusted partners" as well as between partners with whom there is no special trust relationship.

Acknowledgement

This work was partly funded by FLAMINGO, a Network of Excellence project (ICT-318488) supported by the European Commission under its Seventh Framework Programme.

References

[Ar14] Arbor Networks: , Worldwide Infrastructure Security Report - 2014 Volume IX, 2014.

[bl15] blog.sflow.com/: , sFlow Sampling rates, 2015.

[CE16] CERT/NetSA at Carnegie Mellon University: , SiLK (System for Internet-Level Knowledge). [Online]. Available: http://tools.netsa.cert.org/silk, 2016. [Accessed: January 24, 2016].

[DDW00] Debar, Hervé; Dacier, Marc; Wespi, Andreas: A revised taxonomy for intrusion-detection systems. In: Annales des télécommunications. volume 55. Springer, pp. 361–378, 2000.

[GHK14] Golling, Mario; Hofstede, Rick; Koch, Robert: Towards Multi-layered Intrusion Detection in High-Speed Backbone Networks. In: Proceedings of the 6th International Conference on Cyber Conflict (CyCon). IEEE, pp. 1–17, 2014.

[GKS14] Golling, Mario; Koch, Robert; Stiemert, Lars: Architektur zur mehrstufigen Angriffserkennung in Hochgeschwindigkeits-Backbone-Netzen. In: 7. DFN-Forum Kommunikationstechnologien, Beiträge der Fachtagung, 16.-17. Juni 2014, Fulda. LNI. GI, pp. 131–140, 2014.

[Ma09] Mansmann, Florian; Fischer, Fabian; Keim, Daniel A; Pietzko, Stephan; Waldvogel, Marcel: Interactive analysis of netflows for misuse detection in large IP networks. 2009.

[Ma13] Mandiant Corporation: , APT1 - Exposing One of China's Cyber Espionage Units. Mandiant Intelligence Center Report, 2013. http://intelreport.mandiant.com/Mandiant_APT1_Report.pdf, last seen on 27/04/2014.

[nM15] nMon Corp: , Configuring switches to send sFlow, 2015.

[Op12] Open Discussion at www.reddit.com: , Netflow Tools... nfdump vs SiLK tool suite. Anyone have experience?, October 2012. Post filed 15 Oct 2012, http://www.reddit.com/r/netsec/comments/11iszm/netflow_tools_nfdump_vs_silk_tool_suite_anyone/, last seen on 28.07.2015.

[Op14] Open Discussion at http://www.gossamer-threads.com/: , oss netflow collector/trending/-analysis, May 2014. Post filed 2 May 2014, http://www.gossamer-threads.com/lists/nanog/users/171395, last seen on 28.07.2015.

[PP09] Phaal, Peter; Panchen, Sonia: , Packet Sampling Basics, 2009.

[Si15] Simon Leinen: , FloMA: Pointers and Software, April 2015. Last updated 11 April 2015, https://www.switch.ch/network/projects/completed/TF-NGN/floma/software.html, last seen on 28.07.2015.

[SM07] Scarfone, Karen; Mell, Peter: Guide to intrusion detection and prevention systems (idps). NIST special publication, 800(2007):94, 2007.

[Sp10] Sperotto, Anna; Schaffrath, Gregor; Sadre, Ramin; Morariu, Cristian; Pras, Aiko; Stiller, Burkhard: An overview of IP flow-based intrusion detection. Communications Surveys & Tutorials, IEEE, 12(3):343–356, 2010.

[Vi14] Viktor Pus and Lukas Kekely and Martin Spinler and Vaclav Hummel and Jan Palicka: , HANIC 100G: Hardware accelerator for 100 Gbps network traffic monitoring, 2014.