2014 6th International Conference on Cyber Conflict
P.Brangetto, M.Maybaum, J.Stinissen (Eds.)
2014 © NATO CCD COE Publications, Tallinn

Towards Multi-layered Intrusion Detection in High-Speed Networks

Mario Golling
Faculty of Computer Science
Universität der Bundeswehr München
Neubiberg, Germany
mario.golling@unibw.de

Robert Koch
Faculty of Computer Science
Universität der Bundeswehr München
Neubiberg, Germany
robert.koch@unibw.de

Rick Hofstede
Design and Analysis of
Communication Systems (DACS)
University of Twente
Enschede, The Netherlands
r.j.hofstede@utwente.nl

Abstract: 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 suspicious patterns. However, the rapid increase of link speeds and throughputs – especially in larger networks such as backbone networks – seriously constrains this approach. First, 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. Second, legislation commonly restricts the way in which backbone network operators can analyse the data in their networks. To overcome these constraints, flow-based intrusion detection can be applied, which traditionally focuses only on packet header fields and packet characteristics. Flow export technologies are nowadays embedded in most high-end packet forwarding devices and are widely used for network management, which makes this approach economically attractive.

In the context of large, high-speed networks, such as backbone networks, we make two observations with respect to flow-based and packet-based intrusion detection. First, 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. Second, the quantity of information is not constrained when packet-based intrusion detection is performed, but its application is often unfeasible, due to stringent processing requirements. To bridge this gap, we propose a multi-layered approach that combines the advantages of both types of intrusion detection. Our approach is centred around the idea that 1) a first layer of detection comprises flow-based intrusion detection, that makes a pre-selection of suspicious traffic, and 2)

additional packet-based intrusion detection is subsequently performed on a pre-filtered packet stream to facilitate in-depth detection. We demonstrate how this approach avoids the problem of a costly infrastructure, and obeys the various legal barriers on network traffic inspection.

Keywords: *Network Security, Intrusion Detection, High-speed Networks, Flow-Based Intrusion Detection, Legal Inspection*

1. INTRODUCTION

Network attacks have always been present since the birth of the Internet, but high link speeds and the ease of performing and participating in attacks have made this problem the order of the day. Internet insecurity is a worldwide problem that has generated a multitude of costs for businesses, governments, and individuals. When attacks are performed in a distributed fashion, their devastating power can easily overwhelm individual end hosts. For example, the Spamhaus project was targeted by Distributed Denial of Service (DDoS) attacks in early 2013 with more than 300 Gbps of traffic, enough to overload several Internet exchanges [1]. Throughout the last couple of years, in addition to DDoS attacks, in particular worms and botnets also represent special challenges for network operators, since they also tend to consume a great amount of resources [2-5].
To approach the detection of attacks, one of the well-established security solutions nowadays are Intrusion Detection Systems (IDSs). Intrusion detection is defined by the National Institute of Standards and Technology (NIST) as follows [6]:

Intrusion detection is the process of monitoring the events occurring in a computer system or network and analysing 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.

Intrusion detection is usually been performed based on packet payloads. This approach, commonly referred to as Deep Packet Inspection (DPI), provides full visibility in the network traffic, which comes at the expense of scalability. As soon as intrusion detection has to be performed on links with speeds of 10 Gbps and higher, more complex/expensive hardware is needed to cope with the large amount of traffic.
To overcome the scalability problem of packet-based/payload-based intrusion detection, flow-based intrusion detection has been extensively researched [7]. This type of intrusion detection is performed on traffic aggregates, rather than individual network packets, but accuracy and detail are sacrificed for the sake of scalability. Many network operators have flow monitoring facilities at their disposal [8], so deploying them comes at almost no cost. We therefore consider flow-based intrusion detection a viable approach for operators of high-speed networks.
In this paper, we present an approach that exploits the advantages of both, flow-based and packet-based intrusion detection and overcomes many legal obstacles by operating in a multi-layered fashion; we use flow-based intrusion detection as the first layer of detection for identifying potential incidents, while more detailed intrusion detection is used as the second

stage for analysing only the part of the traffic stream that has been reported as suspicious by the first stage. In particular, we focus on backbone networks as a typical example of high-speed networks.

The remainder of the paper is structured as follows. In Section 2, we discuss background information in the field of intrusion detection. An example scenario that highlights the context of this work is described in Section 3. Our multi-layered architecture is discussed in Section 4, followed by an in-depth discussion of how the various architectural components are managed in Section 5. In Section 6, we discuss our first thoughts on the implementation. Section 7 provides an insight on our ideas regarding the evaluation. Finally, we close this work in Section 8 by discussing the next steps to be taken.

2. BACKGROUND

In this section, existing approaches to intrusion detection are briefly introduced. To be able to classify individual systems, we start by presenting a classification scheme for IDSs, which will serve as a basis to classify and evaluate existing approaches according to these criteria.

A. Classification Schemes for Intrusion Detection

Due to the fact that IDSs have been an active research area for several decades, quite a lot of work has been done on the classification of these systems. A classification or taxonomy is a hierarchical structure of a field of knowledge into groups [9]. Here, several properties have to be satisfied (see e.g., [10, 11]): mutual exclusiveness, completeness, traceability, conveniently, clarity and acceptance. However, no generally accepted taxonomy is available for the classification of IDSs and various classifications of very different levels of detail can be found in the literature [9]. The taxonomy published by Debar et al. [12, 13] is used widely [9]. Next to Debar et al., the taxonomy of Axelsson [14] is also generally considered to be a main contribution in this area [7]. In the following, we will briefly describe selected elements of Debar et al. and Axelsson, which are generally used to classify intrusion detection approaches [9]:

Detection Method: With regard to detection, three approaches can be distinguished [6]:

- *Knowledge-based techniques* are based on the idea of comparing currently observed activities (e.g., packets that pass the IDS) to investigate and examine them for the presence of already known attack traces (e.g., using string comparison operations).
- *Behaviour-based techniques* describe the process of comparing definitions of what activities are considered normal with the current events observed to identify significant deviations, using models to predict the expected state of a system. If the predicted and the measured value differ more than a specified threshold, an alert is raised.
- *Compound:* There are also approaches that form a compound decision in view of a model of both, the knowledge-based approach as well as the behaviour-based approach.

Behaviour on Detection/Response: If an IDS does not only monitor events and analyse them for signs of possible incidents, but also attempts to stop detected incidents, it is commonly referred to as an Intrusion Prevention System (IPS) [6]. IDSs are therefore considered as passive, while IPSs are considered reactive.

Audit Source Location: IDSs/IPSs can also be classified based on the audit source location. Although not in the scope of this paper, host-based IDSs, which monitor the characteristics of a single host and the events occurring within that host for suspicious activity, are also one way to classify IDSs. As this publication is focussing mainly on backbone network operators, in the following we focus on network-based IDSs, which monitor network traffic for particular network segments or devices and analyse the network and application protocol events to identify suspicious activities [6].

Time of Detection: Three main classes can be identified. Attempts that perform detection (i) in real-time or (ii) near real-time, and those that process data with a considerable delay, postponing detection; (iii) non-real-time.

Link Speed: This categories indicates whether an approach is able to work in high-speed environments. With this paper, connections of around 1 Gbps are considered as low link speed, whereas high-speed links usually have a data rate of 10 Gbps and higher.

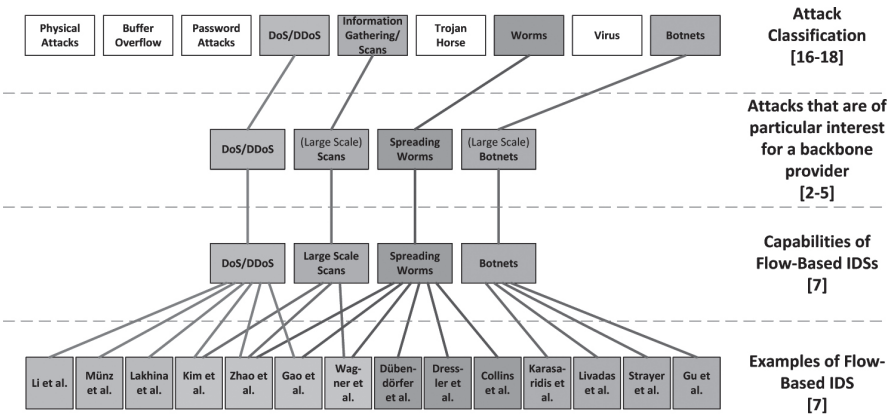
Layer of Detection: Although not considered by Debar et al. and Axelsson, IDSs can also be distinguished based on the layer on which the detection is performed. Header-based IDSs consider only header information, while payload-based IDS investigate both the header and the payload of a packet.

B. Overview of Existing Approaches to Intrusion Detection

We consider three existing approaches to intrusion detection in this work, which will be discussed in the remainder of this subsection:

Flow-based intrusion detection: Flow export technologies, such as NetFlow and IPFIX, are shipped with most high-end routers [7]. Traffic information is collected and stored in flow records that provide an overview of network usage at different levels of granularity. In [15], 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.* Besides management purposes, flows can also be used to perform intrusion detection. With such an approach, the communication patterns within the network are analysed. Compared to traditional network-based IDSs, 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). This is mainly due to the aggregation of packets into flows, which comes at the expense of information granularity for the IDS. Figure 1 gives an overview of attacks that can be detected by flow-based IDSs. For the sake of clarity, it must be noted that in this classification, a virus is regarded as a worm that only replicates itself on the (infected) host computer and needs user interactions to propagate to other hosts [16-18].

FIGURE 1: CAPABILITIES OF FLOW-BASED IDS



As shown in Figure 1, on the one hand, flow-based IDSs are capable of detecting those attacks that are of special interest for a backbone network operator. On the other hand, quite a number of different approaches are available, each of them addressing specific aspects (see [7] for more details). However, the process of metering and exporting flows on a router, the collection of flows and the subsequent analysis consume a relatively large amount of time (up to several minutes [19]), introducing a certain delay within the intrusion detection process

Protocol-based/statistic-based intrusion detection: In contrast to flow-based IDSs, protocol-based/statistical IDSs are also performing decisions based on meta-data (i.e., packet header information), but here on *every* packet instead of an aggregated set of packets. One of the key advantages is that a decision is performed on a larger set of data. Furthermore, the process of generating the meta-data does not consist of multiple steps, but is performed by the IDS itself. Due to the fact that only packet headers are investigated, the approach is also capable of handling multiple Gbps *(medium link speed)*.

Protocol-based IDSs monitor the dynamic behaviour and state of protocols. This method focuses on reviewing the strictly formatted data of network traffic (known as protocols) and searches for benign protocol activity for each protocol state to identify deviations. Unlike traditional behaviour-based intrusion detection, which uses host or network-specific profiles, protocol-based analysis relies on universal profiles that specify how particular protocols should and should not be used. *Stateful protocol analysis methods* (which is a synonym for protocol-based analysis) use protocol models, which are typically based on protocol standards from software vendors and standardization bodies (e.g., IETF) [6]. Each packet is wrapped in predefined layers of different protocols. A protocol-based IDS unwraps and inspects these layers, according to the protocol standards or RFCs. Anything that violates or is outside of these standards is likely malicious.

Statistical-based IDSs rely on statistical models such as the Bayes' Theorem, to identify anomalous packets. These statistics are based on actual usage patterns. As a consequence, statistical systems can adapt to behaviours and therefore create their own rule usage-patterns.

Anomalous activity is measured by a number of variables sampled over time and stored in a profile. In the course of this paper, the term statistical-based IDS is used to classify such behaviour-based approaches that only consider header information (or parts thereof) to generate their statistics (and to perform intrusion detection). Compared to flow-based IDSs, here, approximately the same time is needed for analysis. This is due to the fact that (i) in the case of stateful protocol analysis, the states of the protocol must be investigated for a certain time window, to have a clear indication, or (ii) in case of a statistical-based IDS, a *significant* deviation from the normal state is needed (large dataset). Thus, a near-real-time detection is considered as well.

Payload-based intrusion detection: Within this category, intrusion detection is usually preformed by checking a data stream (including the payload) for the presence of typical patterns, called signatures (knowledge-based approach). Typically, payload-based IDSs like Snort use rules for matching payload data. To this end, however, the entire package contents must be analysed, which slows down the process of intrusion detection, which in turn makes these systems less suitable for using them in high-speed environments. Typical representatives of open source IDSs are Snort, Suricata and Bro. In addition, several commercial products also perform intrusion detection with the use of knowledge-based DPI approaches.

C. Applicability of Existing Approaches for High-Speed Backbone Network Operators

Table 1 provides a brief overview of methods and approaches for intrusion detection. The first column lists previously discussed approaches. The second column lists the typical detection method of the respective approach. The third column provides information whether the approach relies on analysing the header/payload. The fourth column indicates whether the approach is feasible for a high-speed environment. Column five displays the time needed for detection. Finally, column six list the resource-intensiveness resp. the financial efforts for the corresponding approach.

TABLE 1: OVERVIEW OF IDS APPROACHES

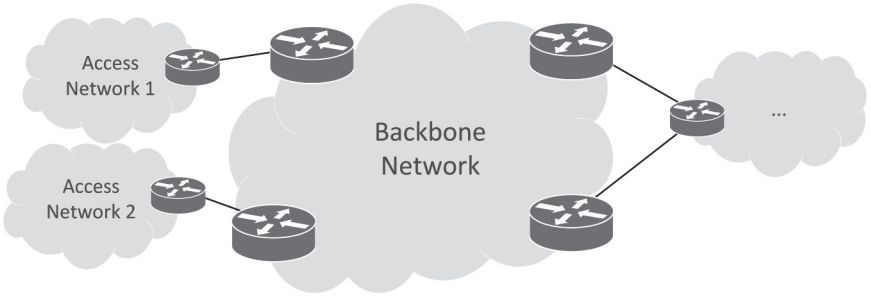
Approach	Typical Detection Method	Layer of Detection	Link Speed	Time of Detection	Financial Expenditure
Flow-Based	Behaviour	Header	High	Near Real-Time	Low
Protocol-Based	Knowledge	Header	Medium	Near Real-Time	Medium
Statistical-Based	Behaviour	Header	Medium	Near Real-Time	Medium
DPI-based	Knowledge	Payload	Low	Real-Time	High

Due to the large amounts of data in a backbone network, only flow-based IDSs can be used in practice. In addition, since customers do not explicitly pay network operators for security mechanisms, investments in IT security are very limited (low *Return on Security Investment (ROSI)*). Along with the ever-increasing data rates this in turn also leads to the fact that only flow-based IDS are used, since flow-based IDSs have by far the lowest financial expenditures [20].

3. SCENARIO

The primary focus of this work is on backbone networks, which we define as networks that do not provide network access to individual end hosts, and use links with speeds of 10 Gbps and higher. This is illustrated in Figure 2, where the backbone network has several edge routers that connect to other backbone networks and several access networks. These *access networks* can be residential Internet Service Providers (ISPs) or university campus networks, for example.

FIGURE 2: SIMPLIFIED BACKBONE NETWORK TOPOLOGY



Performing intrusion detection in backbone networks is subject to several challenges, both technical and legal. First, it is a resource-intensive process that requires expensive hardware to receive, pre-process, store and analyse the collected data. Second, backbone network operators often face legal constraints when performing DPI. DPI can be defined as scanning every byte of a packet payload and identifying a set of matching predefined patterns [21]. Although legislation in the area of packet inspection differs from country to country, the general tendency is that operators are not allowed to deal with data that can be traced back to individuals without permission. Exceptions are operational necessities, research, or court order. As a consequence, the backbone network operator in the context of this paper is generally not allowed to perform DPI, unless supported by a *clearly motivated occasion* or incident.

Many backbone network operators use flow export technologies for monitoring their networks. A recent survey among both commercial and research network operators has shown that 70% of the participants have devices that can export flows [8]. Flow export technologies, such as Cisco's NetFlow [22] or the recent standardization effort IPFIX [15], aggregate packets into flows. Deploying these technologies in backbone networks has several advantages. First, the aggregation of packets into flows significantly reduces the stringent requirements on data storage capacity and data analysis performance. Second, given that many high-end packet forwarding devices, such as routers and switches, already have flow export technologies embedded, deploying flow export comes at virtually no cost. And finally, backbone network operators have to save flow data anyway to comply with data retention laws. For example, network operators in Europe are forced to retain connection information for up to several years [23].

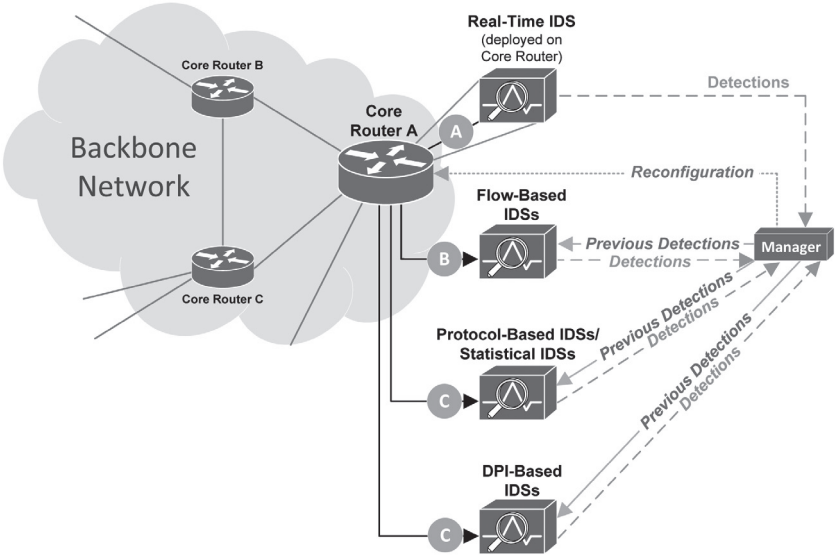
4. ARCHITECTURE

In this section we present our multi-layered architecture. We start by describing its main components and interactions in Section 4-A. In Section 4-B, we describe how existing systems can be integrated into our architecture.

A. Components and Interactions

The main components of our multi-layered architecture, together with their interactions, are shown in Figure 3. It has been designed with simplicity in mind and should be widely deployable.

FIGURE 3: COMPONENTS OF OUR ARCHITECTURE



The *Manager* controls all data-streams, and activates/configures the various IDSs. To make sure that every IDS receives the optimal data-stream, the *Manager* can reconfigure *Router A*. This router is equipped with a *Real-Time IDS* that performs the first layer of intrusion detection. Given that a router's main task is packet forwarding, this IDS is light-weight to not interfere with the router's critical operations.

Several data-streams can be identified in Figure 3:

- A Flow meta-data that can be retrieved directly from the router's Command-Line Interface (CLI);
- B Flow data, exported by means of Cisco's NetFlow [20] or the recent IETF standardization effort IPFIX [15];
- C Full packet streams, potentially pre-filtered by the router upon instruction by the *Manager*.

Key characteristic of the *Real-Time IDS* is that it constantly analyses the full traffic stream, without any form of sampling or filtering. In a previous work, we have shown that a similar approach is able to mitigate DDoS attacks in near real-time [23]. Upon detection of such an attack, the Real-Time IDS can reconfigure the router to drop the attack traffic, to make sure that neither the network itself, nor the monitoring infrastructure is overloaded. In addition, the *Manager* is informed about the attack by means of a standardized message exchange format, such as the Intrusion Detection Message Exchange Format (IDMEF); see [24] for an introduction and evaluation of IDS message exchange protocols.

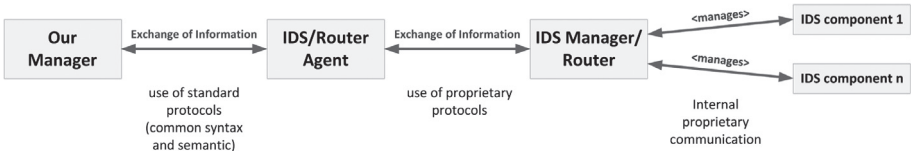
Besides the *Real-Time IDS*, the *Flow-Based IDS*s are also constantly monitoring the input data stream. Given that flow export technologies, such as NetFlow and IPFIX, aggregate packets into flows, such an IDS is usually capable of monitoring the aggregated traffic using commodity hardware. An example of a flow-based IDS is SSHCure, which detects SSH dictionary attacks and reports whether a host has been compromised [25]. The *Flow-Based IDS*s may be informed by the *Manager* about previous detections, and reports its own detections to the *Manager* again. Although not supported by current IDSs, the main idea of forwarding previous detection results to IDSs is to give as much information as possible and so to make the process of intrusion detection as reliable as possible.

In situations where the *Manager* decides to initiate a more extensive analysis of an attack, the *Protocol-Based IDS*s or *DPI-based IDS*s can be activated and instructed. The *Manager* decides which IDS/IDSs is/are most suitable for a particular attack. Before activating the other IDSs, the *Manager* has to reconfigure the router to pre-filter the traffic stream to only include the attack traffic. Analogously to the *Flow-Based IDS*s, these IDSs report their detections to the *Manager*. If an attack has been detected, the router is instructed to drop the attack traffic. If an attack could not be confirmed, the *Manager* will not dispatch any investigation about that particular traffic to the various IDSs anymore.

B. Use of Agents in Case of Proprietary Systems

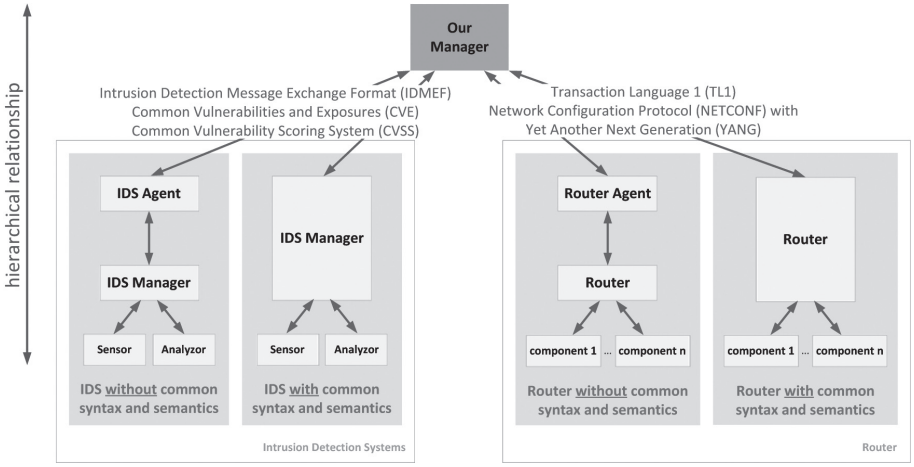
In this section, we discuss how existing systems that do not support standardized protocols for management (e.g., NETCONF) and information exchange (e.g., IDMEF) can be integrated into our architecture. The main idea, which is pursued in our approach, is to use specific agents (see Figure 4).

FIGURE 4: USING AGENTS IN CASE OF NON-STANDARDIZED PROTOCOLS



The agents are adapted for the individual system and thereby convert the standardized protocols used in our architecture into the proprietary counterpart used by the integrated system. This is done for the communication in both directions, i.e. from our *Manager* to the *IDS Manager* / Router, and for the reverse direction. The relationship between our *Manager*, the *IDS / Router Agent* and the *IDS Manager / router* is hierarchical. This means that our *Manager* uses the other Managers. Figure 5 visualizes this, as well as the standardized protocols and methods used.

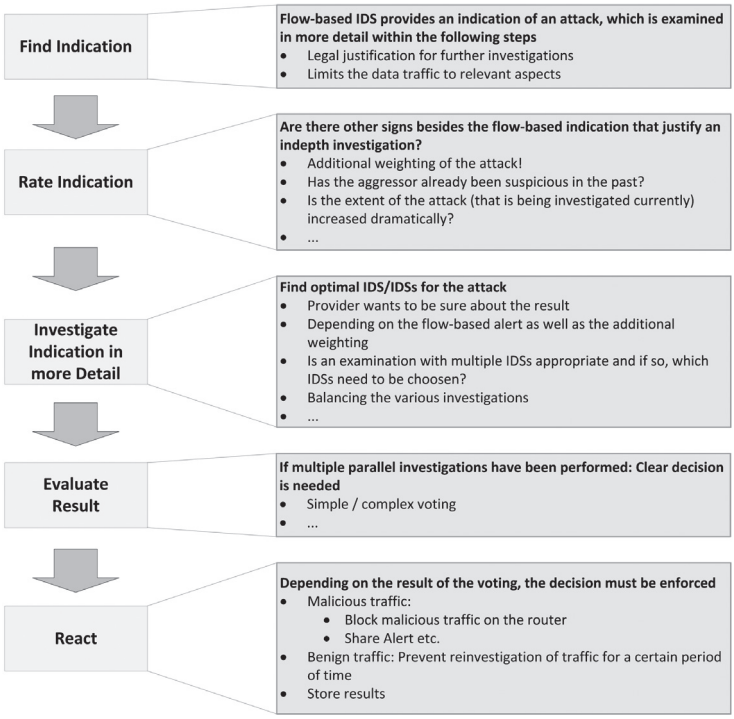
FIGURE 5: INTERACTIONS OF OUR MANAGER WITH EXISTING APPROACHES



5. MANAGER

The *Manager*, which is the architectural component that manages all other components, has a flow of operation as depicted in Figure 6. It consists of the following steps:

FIGURE 6: WORKFLOW OF THE MANAGER



Find Indication: The identification of the indication (*indication of an attack*) marks the beginning of a detailed investigation. For this, a flow-based IDS is used to look for signs of possible attacks. Since this investigation is not performed on packet payloads, both the individual privacy of the users is addressed in particular and the use of inexpensive IDS is made possible (especially since payload-based IDSs do have significant resource requirements). Hence, only few aspects of the data traffic are investigated.

Rate Indication: If an abnormality is detected, it is important to estimate the extent of the attack. Based on the alarm and the corresponding Common Vulnerabilities and Exposures (CVE)/Common Vulnerability Scoring System (CVSS), an assessment of the extent can be made.
While this gives a general assumption on the degree of damage an attack can cause, here, in addition to the scoring of the alert, supplemental criteria are used to estimate the specific severity. Such criteria are for example 1) whether an aggressor has already shown suspicious behaviour in the past, 2) whether the extent of the attack that is being investigated currently increases dramatically (e.g., the number of packets involved increases rapidly), or 3) whether a high number of similar attacks has been observed in the past. For this purpose, inter alia, a self-developed Geo-database is used in order to assist correlating attacks; see [26, 27].

Investigate Indication in more Detail: Depending on the overall scoring as well as individual aspects of the attack (type of attack), corresponding payload-based, protocol-based or statistical IDSs are to be identified. For example, if signs of an SSH-attack are observed by the flow-based IDS, the *Manager* may decide to investigate the relevant traffic by means of a statistical IDS (payload-based IDSs are not useful in this particular case, since SSH traffic is always encrypted). In contrast to this, when signs of a (non-encrypted) worm are detected, the *Manager* may directly involve a payload-based IDS.
As a backbone network operator wants to have a high-level of confidence before potentially mitigating an attack, involving multiple IDSs to investigate an incident may happen very often. The objective of the operator is to maximize the accuracy of the detection result and not to detect as many attacks as possible. However, the presence of several different types of IDSs does not necessarily imply that individual systems are very powerful. Since particularly transit customers don't spend a lot of money for security, an operator – as already mentioned – on the one hand wants to be sure that, if he blocks traffic that this decision in accordance to the law, but on the other hand, he will most likely not allocate powerful resources for that purpose. This leads to the situation that a relatively large number of systems may be available, but all of them with relatively little power. Therefore, it must be considered in advance, whether the specific request for an investigation can be carried out or not. This is mainly based on the scoring (see *Rate Indication*). The higher the score, the more important is a detailed investigation. If two investigations (with the same priority) are in conflict with each other, it is preferred to continue an on-going investigation, rather than to end and begin a new one.

Evaluate Result: Especially after several parallel investigations have taken place, the detection results need to be evaluated and compared. In case of contradictory results, an appropriate conflict resolution mechanism must be conducted. As a backbone network operator – as already stated – wants to be sure that the decisions made by him are solid, several models are conceivable.

On the one hand, this could mean that traffic is blocked only in the case of unanimity of all IDSs involved (which would subsequently lead to the fact that probably comparatively little traffic is blocked). On the other hand, a majority decision also seems to be conceivable. But also in this case, a clear vote seems to be essential, before a backbone network operator will make such a momentous decision like blocking traffic.

React: Once a decision is made, it must be enforced as well. In case of malicious traffic, corresponding packets must be blocked on the router. But even in the case of benign traffic, some actions need to be performed accordingly. E.g., it should be ensured that the traffic is not examined a second time (within a certain time period). In both cases, the result of the investigation is stored locally and also forwarded to other routers, which may include this result by means of the phase *Rate Indication*.

6. IMPLEMENTATION

For the realization of our architecture and implementation of a prototype, we use libraries and implement additional new modules and probes. As discussed before, the *Manager* is the central component of our architecture. It realizes the forwarding and selection of the network traffic, as well as the distribution based on the flow of operation presented in Section 5 as well as the configuration and assessment of alerts and their scores to the networks under consideration. The main routines of the controller are written in C programming language for the sake performance, combined with various open-source libraries.

The *Manager* contains a MySQL database, as well as different APIs to access and import data from various systems, such as CVSS and CVE details. With the help of the GUI of our Manager, the network security personnel can review and assess the relevance of the different threats. By this, the *Manager* is able to do additional weighting of possible attacks, including an estimation of the endangerment for the own network, and assigning examination orders to other IDSs. At the moment, the GUI is realized by a *ncurses* surface, but the upcoming prototype will be based on a Web interface. For further inspection of suspicious traffic, the *Manager* can forward the flows and network packets for a protocol analysis and further behaviour-based evaluations.

For our first prototype, we perform enhanced protocol analysis based on a special configured Snort IDS. Therefore, a standard Snort IDS is used with minimal functionality, disabling all signature-based detection schemes and only using the protocol analysis. In addition, we started to implement different modules for a behaviour-based protocol analysis. These modules are realized based on NFDUMP and the functionalities of the *nfreader* framework. Because of the comprehensive analysis of the protocols and the practical differences of their implementations in different operating systems, these modules will only be fully functional in a later release of our prototype.

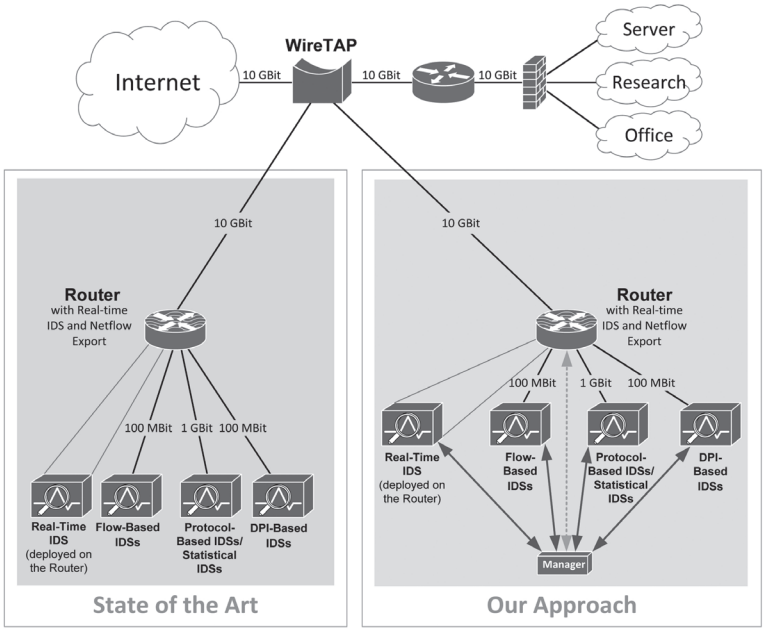
For the integration of knowledge-based and behaviour-based IDSs, a regular setup of Snort is used on the one hand, and a FlowMatrix system for the behaviour-based detection on the other hand.

The exchange of incident information between the different components and modules of our prototype is realized by IDMEF, for which the *LibIDMEF* is used [28].

7. EVALUATION

The first prototype is currently being tested extensively in our lab. For this purpose, a test set-up was built as described in more detail in Figure 7.

FIGURE 7: EVALUATION SETUP



With regard to our investigation, the department comprises three different networks: A server-network (with production systems), a research network (where various systems such as honeypots are tested, operated under specific conditions and evaluated) and a network for the office IT. By using a hardened system including a firewall and the application of additional protective measures, these three networks are separated intensively from each other (for more details see [29]).

With regard to state-of-the-art, the traffic is forwarded to the router. On the router itself, the *Real-Time IDS* is deployed and the Router also exports the data stream in the form of NetFlow V9 records, which are then analysed by the flow-based IDSs. In parallel, the protocol-based IDSs, statistical IDSs and DPI-based IDSs are supplied directly with data from the router. Here (as well as in our approach), the flow-based IDSs are connected with 100 Mbps (which is more than sufficient to handle the flow export records of the 10 Gbps link), while the protocol-based IDSs/statistical IDSs are connected with 1 Gbps and the DPI-based IDSs are connected at 100 Mbps. Of course, the flow export conditions are the same for state-of-the-art and our approach. In our approach the respective IDSs are connected using the same router model (Cisco 6513) and with the use of the *Manager* (as described in the previous sections).

Although it is too early to present results in detail, the first preliminary results are very promising. For the comparison, typical criteria such as 'probability of detection' (the ability of an IDS to identify positive results; proportion of malicious events that have been detected), false-alarm ratio (benign traffic that has been classified as malicious) and accuracy (proportion of true results, both true positives and true negatives) will be used. It may again be emphasized that the goal of our approach is not to identify as many positive results as possible (Probability of Detection), but to reduce the false alarm ratio.

8. CONCLUSIONS AND OUTLOOK

In this paper, we have presented a first step towards multi-layered intrusion detection, which aims both at reducing costs by being deployable on commodity hardware, and overcoming legal legislation with respect to traffic analysis (clearly motivated occasion in form of a flow-based alert is given before DPI is performed). Although a generic yet simple architecture has been defined and a first implementation realized, more steps have to be taken as future work before our IDS can be fully deployed in an operational environment. We shortly highlight these steps in the remainder of this section.

First, we plan to include more material on legislation in various countries with respect to network traffic analysis. As we want our multi-layer IDS to be as widely deployable as possible, this will be needed before finalizing the implementation.

The final design of our system will respect country-specific restrictions and possibilities. An auto-configuration based on the detected country will be provided, which can be tuned by the administrator. If modifications of the administrator violate the local restrictions, a warning will be given.

Second, after finishing the implementation, we plan to deploy it subsequently on campus-wide, region-wide and nation-wide scales. The goal of the various levels of deployment is twofold:

1. As operators of networks at different scales tend to use different devices and configurations, deploying our IDS in several networks allows us to validate its accuracy in multiple situations. For example, the flow data exported in campus networks is often exported with a sampling rate of 1:1 (i.e., everything is sampled), while nation-wide networks are often using sampling with a rate of 1:100, to reduce the data exported from the network. Our IDS should be able to cope with the difference in data granularity and should therefore be tested under these conditions, e.g., in terms of accuracy.
2. We have to get feedback from operators with respect to operational aspects. For example, we have to survey whether operators have technical facilities for deploying the various IDSs.

Third, we are trying to improve intrusion detection through inter-domain exchange of knowledge of attacks, both between "trusted partners" (in our case, within the so-called Joint Security Lab, consisting of various infrastructures operated by partners of Flamingo, a Network of Excellence project) and between partners with whom there is no special trust relationship. See [30] for an overview of our thoughts on this.

ACKNOWLEDGEMENTS

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:

[1] Ars Technica, “Can a DDoS break the Internet? Sure... just not all of it“ April 2013, accessed on 25 November 2013. [Online]. Available: http://arstechnica.com/security/2013/04/can-a-ddos-break-the-internet-sure-just-not-all-of-it/
[2] Arbor Networks, “Worldwide ISP Security Report“.
[3] Arbor Networks, “Worldwide Infrastructure Security Report“.
[4] Arbor Networks, “Worldwide Infrastructure Security Report - 2011 Volume VII“.
[5] Arbor Networks, ‘Worldwide Infrastructure Security Report - 2012 Volume VIII“.
[6] K. Scarfone and P. Mell, “Intrusion detection and prevention systems“ in Handbook of Information and Communication Security. Springer, 2010, pp. 177-192.
[7. A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, and B. Stiller, “An Overview of IP Flow-Based Intrusion Detection,“ IEEE Communications Surveys & Tutorials, vol. 12, no. 3, pp. 343-356, 2010.
[8] J. Steinberger, L. Schehlmann, S. Abt, and H. Baier, “Anomaly Detection and mitigation at Internet scale: A survey,“ in Proceedings of the 7th International Conference on Autonomous Infrastructure, Management and Security, AIMS’13, Lecture Notes in Computer Science, vol. 7943. Springer Berlin Heidelberg, 2013, pp. 49-60.
[9] R. Koch, B. Stelte, and M. Golling, “Attack Trends in present Computer Networks,“ in Proceedings of the 4th International Conference on Cyber Conflict (CyCon). IEEE, june 2012, pp. 1-12.
[10] J. D. Howard and T. A. Longstaff, “A common language for computer security incidents“ Sandia Report: SAND98-8667, Sandia National Laboratories, http://www.cert.org/research/taxonomy 988667.pdf, 1998.
[11] S. Jin, Y. Wang, X. Cui, and X. Yun, “A review of classification methods for network vulnerability,” in Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on. IEEE, 2009, pp. 1171-1175.
[12] H. Debar, M. Dacier, and A. Wespi, “Towards a taxonomy of intrusion-detection systems,“ Computer Networks, vol. 31, no. 8, pp. 805-822, 1999.
[13] H. Debar, M. Dacier, and A. Wespi, “A revised taxonomy for intrusion-detection systems,“ in Annales des télécommunications, vol. 55, no. 7-8. Springer, 2000, pp. 361-378.
[14] S. Axelsson, “Intrusion detection systems: A survey and taxonomy“ Technical report, Tech. Rep., 2000.
[15] 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), 2013.
[16] V. Igure and R. Williams, “Taxonomies of attacks and vulnerabilities in computer systems“ Communications Surveys & Tutorials, IEEE, vol. 10, no. 1, pp. 6-19, 2008.
[17] N. Weaver, V. Paxson, S. Staniford, and R. Cunningham, “A taxonomy of computer worms,“ in Proceedings of the 2003 ACM workshop on Rapid malcode. ACM, 2003, pp. 11-18.
[18] S. Hansman and R. Hunt, “A taxonomy of network and computer attacks,“ Computers & Security, vol. 24, no. 1, pp. 31-43, 2005.
[19] R. Hofstede, V. Bartos, A. Sperotto, and A. Pras, “Towards Real-Time Intrusion Detection for NetFlow and IPFIX” in Proceedings of the 9th International Conference on Network and Service Management, CNSM’13, 2013, pp. 227-234.
[20] M. Golling and B. Stelte, “Requirements for a Future EWS - Cyber Defence in the Internet of the Future” in Proceedings of the 3rd International Conference on Cyber Conflict (ICCC). IEEE, June 2011.
[21] S. Kumar, J. Turner, and J. Williams, “Advanced Algorithms for Fast and Scalable Deep Packet Inspection,“ in Proceedings of the 2006 ACM/IEEE symposium on Architecture for networking and communications systems, 2016, pp. 81-92.
[22] B. Claise, “Cisco Systems NetFlow Services Export Version 9“ RFC 3954 (Informational), 2004.
[23] W. John, S. Tafvelin, and T. Olovsson, “Passive Internet Measurement: Overview and Guidelines based on Experiences,” Computer Communications, vol. 33, no. 5, pp. 533-550, 2010.
[24] R. Koch, M. Golling, and G. D. Rodosek, “Evaluation of State of the Art IDS Message Exchange Protocols” in International Conference on Communication and Network Security (ICCNS), 2013.

[25] L. Hellemons, L. Hendriks, R. Hofstede, A. Sperotto, R. Sadre, and A. Pras, "SSHCure: A Flow-Based SSH Intrusion Detection System," in Dependable Networks and Services. Proceedings of the 6th International Conference on Autonomous Infrastructure, Management and Security, AIMS'12, Lecture Notes in Computer Science, vol. 7279. Springer Berlin Heidelberg, 2012, pp. 86-97.
[26] R. Koch, M. Golling, and G. D. Rodosek, "Advanced Geolocation of IP Addresses" in International Conference on Communication and Network Security (ICCNS), 2013.
[27] R. Koch, M. Golling, and G. D. Rodosek, "Geolocation and Verification of IP Addresses with Specific Focus on IPv6" in 5th International Symposium on Cyberspace Safety and Security (CSS 2013). Springer, 2013.
[28] „LibIDMEF," accessed on 25 November 2013. [Online]. Available: http://sourceforge.net/projects/libidmef/
[29] R. Koch, and M. Golling, "Architecture for Evaluating and Correlating NIDS in Real-World Networks" in Proceedings of the 5th International Conference on Cyber Conflict (CyCon), 2013.
[30] M. Golling, R. Koch, and G. D. R. Rodosek, "From Just-in-Time Intrusion Detection to Pro-Active Response by Means of Collaborated Cross-Domain Multilayered Intrusion Detection", poster presented at the 9th International Conference on Cyber Warfare and Security ICCWS-2014.