Real-Time and Resilient Intrusion Detection: A Flow-Based Approach

Rick Hofstede, Aiko Pras

Real-Time and Resilient Intrusion Detection: A Flow-Based Approach

Rick Hofstede, Aiko Pras

Design and Analysis of Communication Systems (DACS)
University of Twente
Enschede, The Netherlands
{r.j.hofstede,a.pras}@utwente.nl

Abstract. Flow-based intrusion detection will play an important role in high-speed networks, due to the stringent performance requirements of packet-based solutions. Flow monitoring technologies, such as NetFlow or IPFIX, aggregate individual packets into flows, requiring new intrusion detection algorithms to deal with the aggregated data. These algorithms are subject to constraints on real-time and accurate detection of intrusions, due to the nature of current flow monitoring technologies. In this paper, we propose a framework for flow-based intrusion detection, aiming to detect intrusions in real-time, and to be resilient against negative effects of attacks on monitoring systems. This research is still in its initial phase and will contribute to a Ph.D. thesis after four years.

1 Introduction

Monitoring high-speed networks is a crucial and challenging task. Packet-level monitoring technologies became almost unfeasible, due to limited processing power and storage capacity on monitoring systems. Especially for monitoring links with line speeds of 10 Gbps and higher, scalable monitoring technologies are inevitable. Flow monitoring technologies, such as Cisco's NetFlow [1] or IPFIX [2], were designed for this purpose, and are widely deployed in routers, switches and dedicated probes. Following this trend, security analysis applications should evolve from packet-based to flow-based solutions [3]. Research has already been performed on flow-based anomaly detection in past years, such as in [4–6].

Besides line speeds and available bandwidth, also the size and frequency of attacks are increasing. When these attacks target monitoring systems (either directly or indirectly), these systems - as well as the associated data - will be affected [7]. For the specific case of flow monitoring, this applies to flow *Exporters* and *Collectors*, which represent the flow data export and collection processes, respectively. *Exporters* account traffic statistics by aggregating packets that share certain properties, known as the *flow key*[1]. Thereafter, this aggregated flow data is sent to flow *Collectors* for persistent storage and analysis.

[1] A commonly used flow key consists of source/destination IP addresses and port numbers, transport-layer protocol, IP Type of Service (ToS) and SNMP input interface.

In order for intrusion detection systems (IDSs) to be reliable, detection should be done as accurate and early as possible. However, due to the design of current flow monitoring technologies, flow-based monitoring systems are subject to the following problems:

1. *Exporters* send flow records to *Collectors* after expiration. Several expiration rules can be implemented [1, 8], but timeout-based expiration rules are required. As a consequence, data analysis applications, which retrieve flow data from a *Collector*, always receive the data after a certain delay. Besides that, many *Collectors* make the stored flow data only available after a certain time interval, which should also be added to the total incurred delay.
2. When an *Exporter* is monitoring an ongoing attack, it might become overloaded and start to behave differently [7]. *Collectors* might start to experience problems as well, because of the immense amount of flow records they need to process (*e.g.* for filtering, compressing and generating statistics) as part of an attack, for example.

To overcome these delay and performance issues, a scalable and resilient solution is needed for detecting and regulating (*i.e.* to avoid overloading the equipment) attack traffic for analysis by IDSs. We believe it to be a promising approach to move a part of the intrusion detection logic to the *Exporter*, such that detection algorithms are not subject to delays during the flow export. Besides that, *Exporters* and *Collectors* should share their knowledge about ongoing anomalies, in order to take countermeasures against overload as a consequence of an attack.

The remainder of this paper is organized as follows: Section 2 will discuss the research questions for this work, followed by a description of the proposed approach in Section 3. After that, we conclude and close this paper in Section 4.

2 Research Questions

The goal of this research is ***to design a scalable framework for real-time and resilient intrusion detection***. A flow-based approach will be chosen in order to satisfy the requirement of scalability. In the remainder of this paper, we refer to this goal as *the framework*. To conduct the research presented above, we define the following research questions:

1. *What are the implications for flow monitoring architectures when intrusion detection needs to be done in real-time?*
2. *How can scalability of flow-based IDSs be improved, in order to make these systems resilient against attacks?*
3. *How can requirements for real-time and resilient intrusion detection be modeled in a framework?*

The approach used for answering each of these research questions will be discussed in the next section.

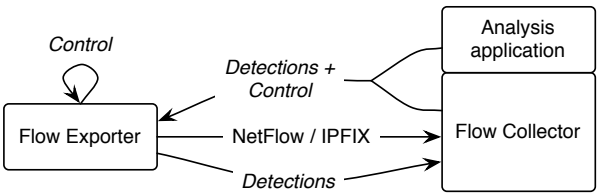

Fig. 1. Flow-based framework for real-time and resilient intrusion detection

3 Approach

Research question 1, as presented in the previous section, aims to get a thorough understanding of performance bottlenecks in current *Exporters* and *Collectors*, which prevent these systems from serving flow data to IDSs in real-time. A real-time intrusion detection framework might require some architectural changes to flow monitoring systems, as anomaly detection should be done as close as possible to the flow *Exporter* (*i.e.* the measurement point). This study will be performed by conducting a literature study, combined with real-world measurements.

The goal of research question 2 is to investigate how scalability of flow-based IDSs can be improved. A naive approach would be to increase the sampling rate in various stages of the analysis process, for decreasing the load of the involved systems. Also flow expiration timeouts could be increased for the same purpose. Although both changes will result in a higher flow record aggregation, the quality and usefulness of the flow data for intrusion detection will be negatively impacted [9, 10]. Hence, a tradeoff should be made between 1) flow data quality and 2) scalability of the framework, which will be covered by research question 2.

By means of research question 3, we will propose a framework for flow-based intrusion detection, in which the knowledge obtained in the previous research questions will be incorporated. Nowadays, a single, one-directional data stream is used between *Exporters* and *Collectors*, by means of which flow records are transferred. Our aim is to enrich this architecture with the following complementary data streams, as depicted in Figure 1:

1. *Exporter* to *Collector*. After moving a part of the intrusion detection logic to the *Exporter*, detected anomalies should be shared with the *Collector*.
2. *Collector* to *Exporter*. Intrusions detected by the *Collector* should be shared with the *Exporter*. Both the *Collector* and the *analysis application* can instruct the *Exporter* to adjust the data export process. This can be done for several reasons, such as performance constraints during ongoing attacks. Finally, the *Collector* sends status information about its available processing capacity, to which the *Exporter* can adapt the NetFlow/IPFIX data stream.
3. *Exporter* to *Exporter*. The *Exporter* should monitor its own available processing capacity and change the data granularity accordingly, in order to stabilize its load.

After proposing the framework, its correctness and applicability will be validated by implementing a proof-of-concept. The result will be validated against a ground truth in a lab environment. After that, the accuracy and resilience of the proposed solution will be verified in a backbone network, such as the SURFnet [11] network.

4 Final Considerations

When flow-based technologies are used in an intrusion detection context, flow data should be accurate and the infrastructure resilient. The nature of these technologies, however, poses constraints on these requirements. This Ph.D. research aims to design a framework that should satisfy the requirements on flow-based IDSs. The main goal of this work, as described before, should be achieved within a period of four years, as part of a Ph.D thesis.

Acknowledgments

This research is supported by the EU FP7-257513 UniverSelf Collaborative Project and the SURFnet GigaPort3 Project for Next-Generation Networks.

References

1. Claise, B.: Cisco Systems NetFlow Services Export Version 9. RFC 3954 (Informational) (October 2004)
2. Sadasivan, G., Brownlee, N., Claise, B., Quittek, J.: Architecture for IP Flow Information Export. RFC 5470 (Informational) (March 2009)
3. Zseby, T., Boschi, E., Brownlee, N., Claise, B.: IP Flow Information Export (IPFIX) Applicability. RFC 5472 (Informational) (March 2009)
4. Sperotto, A., Schaffrath, G., Sadre, R., Morariu, C., Pras, A., Stiller, B.: An Overview of IP Flow-Based Intrusion Detection. IEEE Communications Surveys Tutorials **12**(3) (2010) 343–356
5. Sperotto, A.: Flow-Based Intrusion Detection. PhD thesis, University of Twente (October 2010)
6. Münz, G., Carle, G.: Real-time Analysis of Flow Data for Network Attack Detection. In: Proceedings of the 10th IFIP/IEEE International Symposium on Integrated Network Management (IM 2007). (2007) 100–108
7. Sadre, R., Sperotto, A., Pras, A.: The Effects of DDoS Attacks on Flow Monitoring Applications. In: Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS 2012). (2012) To appear.
8. Quittek, J., Bryant, S., Claise, B., Aitken, P., Meyer, J.: Information Model for IP Flow Information Export. RFC 5102 (Standards track) (January 2008)
9. Bartos, K., Rehak, M., Krmicek, V.: Optimizing Flow Sampling for Network Anomaly Detection. In: 7th International Wireless Communications and Mobile Computing Conference (IWCMC 2011). (2011) 1304–1309
10. Duffield, N., Lund, C., Thorup, M.: Properties and Prediction of Flow Statistics from Sampled Packet Streams. In: Proceedings of the ACM SIGCOMM Internet Measurement Workshop. (2002) 159–171
11. SURFnet. http://www.surfnet.nl/en (2012) Accessed on March 29, 2012.