

# High-Speed Network Traffic Acquisition for Agent Systems\*

Pavel Čeleda, Vojtěch Krmíček  
Institute of Computer Science  
Masaryk University  
{celeda,vojtec}@ics.muni.cz

Martin Reháček<sup>1</sup>, David Medvigy<sup>†</sup>  
Center for Applied Cybernetics<sup>1</sup>, Department of Cybernetics  
Czech Technical University  
mrehak@labe.felk.cvut.cz

## Abstract

*This paper presents a design of high-speed network traffic acquisition subsystem suitable for agent-based intrusion detection systems. To match the performance requirements and to improve network traffic measurement, wire-speed data acquisition layer is based on hardware-accelerated probes, which provide real-time network traffic statistics. The network traffic is stored in collector servers and pre-processed data is then sent to detection agents that use heterogeneous anomaly detection methods. These methods are correlated by means of trust and reputation models, and the conclusions regarding the maliciousness of the traffic is presented to the operator. Presented system is designed to improve the performance of agent-based intrusion detection systems and allow them to efficiently identify malicious traffic. The main contribution of presented system is its ability to aggregate real-time network-wide statistics from geographically dispersed probes. Traffic acquisition system is designed for deployment on high-speed backbone networks.*

## 1. Introduction

With the increasing number of network users, services and the current generation of high-speed network links, the amount of transferred data has increased significantly. In order to detect an anomaly from traffic on the backbone level, network-wide traffic data is essential. These facts have rendered many past methods for intrusion detection obsolete and current backbone networks lack efficient technology for real-time detection of malicious traffic.

To address the above situation, and to enable the operators of backbone and large enterprise networks to analyze

---

\*This material is based upon work supported by the European Research Office of the US Army under Contract No. N62558-07-C-0001. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the European Research Office of the US Army. Also supported by Czech Ministry of Education grant 1M0567.

<sup>†</sup>Currently affiliated with NJIT, work performed during his research visit to CTU.

current threats in near real-time, we present a design of an autonomous system able to detect malicious traffic on high-speed networks. While the system reasoning is based on intelligent agents and multi-agent methods, the network traffic data acquisition and preprocessing in both dedicated adaptive hardware and specialized software are essential for project success. This is because traditional agent techniques are not well suited for efficient low-level traffic processing.

This work presents an agent interface for high-speed network traffic inspection and traffic statistics aggregation. We use the state-of-the-art approaches to match the performance requirements and to improve network traffic measurement. The system is fully distributed and the measurement nodes can adapt their functionality in real-time.

## 2. Related Work

Current agent-based network intrusion detection systems often lack of real-time traffic statistics. The data are typically read from off-line data sources like a files which contain some traffic samples e.g. worm spreading. Such traffic samples are only snapshots of traffic and won't allow us to observe careful attacks that are not as noisy and bursty as say a worm spread. The off-line data don't correspond long-time real live network traffic and the possibilities to efficiently design and verify new intrusion detection methods are very limited.

The modern intrusion detection methods aim to observe the network holistically to more effectively identify network traffic anomalies. Significant numbers of network nodes must be identified and observed over a long period of time. Such network nodes are typically geographically dispersed.

In order to detect an attack from the traffic information on the backbone level, especially without any feedback from the affected hosts, we have to analyze the patterns in the traffic statistics, compare them with normal behavior and conclude whether the irregularity corresponds to a known attack profile or not.

We use two principal types of network sensors (*i*) signature matching sensors and (*ii*) traffic flow sensors, to provide real-time traffic statistics.

The signature matching techniques that detect intrusions by detecting patterns specific to known attacks in network traffic, can be found in [5]. This technique is widely used in many existing IDS systems including SNORT [10].

The traffic flow approach detects intrusions by fusing information from flow measurements taken throughout a network. The flows are identified by the 5-tuple headers and provide information about the IP addresses, ports, byte counts, packet counts, and flows counts.

### 3. System Architecture

Dealing with high-speed networks requires an architecture that can handle large amounts of data and high data rates. We designed a layered architecture where each layer abstracts further from the low-level technical observations and measurements. This reduces the data sets that are reported to higher-level layers as much as possible.

The presented system architecture (introduced in [7]) consists of three layers with varying requirements on on-line processing characteristics, level of reasoning and responsiveness. While the low-level layers need to be optimized to match the high performance during the network traffic acquisition and preprocessing, the higher layers use the preprocessed data to infer the conclusions regarding the degree of anomaly and consecutively also the maliciousness of the particular traffic. Therefore, while the computation in the higher layers must be still reasonably efficient, the preprocessing by the lower layers allows us to deploy more sophisticated algorithms. The system can be split into three layers, as shown in Figure 1:

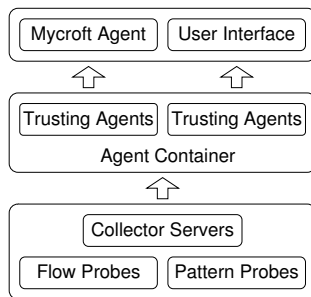


Figure 1. Agent-based IDS system

- **Traffic Acquisition and Preprocessing Layer:** The components in this layer acquire the data from the network using the hardware accelerated probes [2] and perform their preprocessing. The network traffic statistics are stored in collector servers. Preprocessed data is then sent to detection agents. This approach provides the real-time statistics of the network traffic on the observed networks.

- **Cooperative Threat Detection Layer:** This layer principally consist of specialized, heterogeneous agents that seek to identify the anomalies in the preprocessed traffic

data by means of their extended trust models [8]. Their collective decision regarding the degree of maliciousness of a traffic with certain characteristics use a reputation mechanism. The agents run inside the A-globe agent platform [9] and use its advanced features like agent migration and cloning to adapt the system to the traffic and relevant threats.

- **Operator and Analyst Interface Layer:** This layer is responsible for interaction with operator. The main component is the intelligent agent called Mycroft [6] that helps the operator to analyze the output of the detection layer, by putting the anomaly information in context of other relevant information.

### 4. High-Speed Network Traffic Acquisition

The state-of-the-art methods for network security analysis and intrusion detection rely on combining and correlating data from various sources such as link utilization, statistics about IP flows and detailed packet payload inspection. The network traffic acquisition layer use the pattern matching probes and flow probes to provide such statistics.

In general, flows are a set of packets which share a common property. The simplest type of flow is a 5-tuple, with all its packets having the same source and destination IP addresses, port numbers and protocol. Flows are unidirectional and all their packets travel in the same direction [3].

The amount of traffic in nowadays high-speed networks increases continuously and traffic characteristics change heavily in time. Performance of network probes must be independent of such states and behave reliably in all possible cases. The quality of provided data significantly effects the upper layers and chances to detect traffic anomalies.

Therefore we use hardware accelerated NetFlow probes which we have developed in Liberouter project. FlowMon probe is a passive network monitoring device based on the COMBO hardware [2], which provides high performance and accuracy. The FlowMon probe handles 1 Gb/s traffic at line rate in both directions and exports acquired NetFlow data to different collectors.

The collector servers store incoming packets with NetFlow data from FlowMon probes into database. The collectors provide traffic statistics via *tasd* (Traffic Acquisition Server Interface Daemon) interface to the A-Globe agent platform.

### 5 Collector - Agent Layer Interface

Traffic acquisition layer provides the data to detection layer periodically, at the end of each observation period. The goal of the detection process is to identify potentially malicious traffic and to report these traffic to operator and analyst interface layer elements for analysis and possible reaction.

Flow Aggregation	flows	packets	bytes	H(srcIP)	H(dstIP)	H(dstPort)	H(srcPort)
srcIP	×	×	×		×	×	×
dstIP	×	×	×	×		×	×
srcIP/dstPort	×	×	×				
dstIP/srcPort	×	×	×				

**Table 1. *tasd* preprocessed NetFlow data.**

Agent-side feed acquisition interface component handles the client part of connection to collector servers, performs data transformation and provides the data to detection agents in an efficient uniform format, regardless of their source. Detection agents then receive the data, perform anomaly detection and update their trust models.

In order to improve the efficiency of the solution, the individual detection agents don’t establish their own connection to collector servers, but receive the data through a shared interface component, via standard agent messages. This interface – *NetFeeder* – is implemented as an A-Globe service [9], with one instance running per agent container and providing following functionality:

- **TASI connection management:** TASI collector servers receive data from pre-configured traffic probes. The *NetFeeder* service is responsible for setting up, maintaining and terminating connection to collector servers. Typically every 5-minutes (or similar time interval) a preprocessed data (see Table 1) is sent by *tasd* to *NetFeeder*.

- **Flow conversion into internal ontology:** Due to the amount of data to process in a very restricted time frame, we need to limit the overhead related to agent processing (as well as other overhead), while maintaining traditional advantages of agent technology deployment: autonomy, encapsulation, parallel and distributed processing transparency and others. We have realized that the system needs to be designed for efficient data processing from the beginning. The number of flows on gigabit lines is counted in millions, and representing each flow as a full-fledged object (that would be composed of other objects for individual fields such as IP addresses), creating one instance of all flow objects per each detection agent and passing this information to individual agents through traditional decoupling conduits (serialization, IIOP, XML format) would create an enormous processing and memory overhead. First, we have realized that detection agents don’t actually need to modify the traffic data they receive. This fact has allowed us to take advantage of recent A-Globe feature: sending messages as references. Using this method of message transmission, the agents within the platform can share a single instance of data, eliminating the need for costly serialization process and greatly reducing the memory requirements. On the downside, this feature shall be used only with extreme care: if one agent manages to change the data, the change will be

reflected in all agent’s computation, breaking the principles of separation and encapsulation. In our implementation, we enforce separation by using unmodifiable collection classes for aggregates and by proprietary iterators.

The approach described above limits the number of objects as it requires only one instance of each flow representation per container. However, this is still a vast amount, and a mere creation of several millions of objects would come at significant cost. Therefore, we take advantage of the fact that detection agents use most flows only once. This fact has led us to consider a simple improvement: we have decoupled the flow class from the data about the flow it represents, and we only use this class as a cursor (i.e. iterator) over the flows set, which is physically represented by giant two-dimensional arrays<sup>1</sup>, with one row per each flow<sup>2</sup>. Therefore, when the agents iterate over the *Flows*, they actually use a single *Flow* instance for all flows in the set, greatly reducing the object creation/destruction overhead, while maintaining the same programmatic interface as with individual object instances. Long-lived flows, such as centroids in the trust models of detection agents, are cloned from the *Flows* and are backed by single-lined arrays.

For further reduction of computational costs, we have noticed that all detection agents retrieve additional aggregate information (i.e. context) for each flow categorization, and that this information is often the same for several agent types. As the cost of retrieval from the *HashMap* structures holding the statistics is not trivial, we have actually decided to perform this gathering at the level of *NetFeeder*.

## 6. System Evaluation and Performance

The performance is becoming a key concern of NIDS in high-speed networks. The results of traffic acquisition and processing vary depending on the amount of acquired data. Numerous existing NIDS are based on commodity hardware with open-source software and very limited hardware acceleration.

Existing flow monitoring systems are mostly based on exporting flow data from routers. The routers are dedicated for routing the data in networks and enabling the flow export has often negative impacts on overall router performance especially during attacks.

The articles [4, 1] study whether existing sampling techniques distort traffic features that are critical for effective anomaly detection. They show that packet sampling methods introduce fundamental bias that degrades the performance of the anomaly detection algorithms. To avoid such a misbehavior the *FlowMon* probe provides non-sampled data, without packet loss at a line rate.

<sup>1</sup>Technically, the arrays are one dimensional, but it is convenient to refer to 24 bytes relative to each flow as a line.

<sup>2</sup>In the code, we emphasize this counter-intuitive optimization by the fact that *Flow* class actually implements *Iterator<Flow>* interface.

## 6.1. Network Traffic Statistics

The proposed system for high-speed network traffic acquisition is deployed on network of Masaryk University. The university network consists of thousands of computers located in more than fifty buildings spread in Brno city. The metropolitan university network is connected to the Czech national educational network (CESNET).

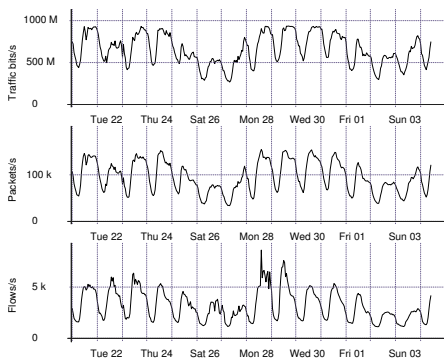


Figure 2. The backbone NetFlow statistics.

The deployed probes observe the backbone nodes, where the connections to other ISPs are located, and edge nodes where the students and university staff are connected.

The Figure 2 shows NetFlow statistics of traffic passed by network backbone node. They are visible nightly and days off traffic fluctuations. It is almost impossible for the human operator to observe the anomaly in overall traffic graph. The agent-based traffic preprocessing is necessary for the human operator to discover network anomaly.

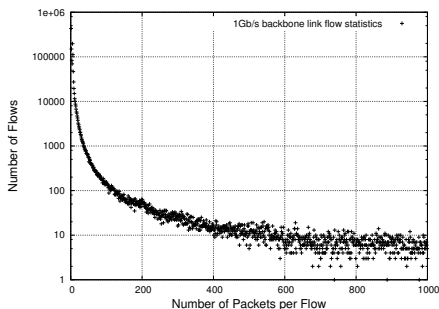


Figure 3. Number of packets per flow.

The Figure 3 shows the traffic flow structure. The most flows contain less than 10 packets with significant number of 1 packet's flows. Such small flows are often not observed if traffic sampling is used. Several Internet worms use for their spreading one packet technique e.g the Witty worm or SQL slammer worm. The current network traffic also contains a lot of small flows which must be handled to detect traffic anomaly.

## 7. Conclusions and Future Work

Our work presents a system for network traffic acquisition that is optimized for deployment on backbone networks. The designed system addresses two main limitations of existing agent-based intrusion detection systems – input of real-time network-wide traffic statistics and the overhead associated with distributed parallel processing of data.

Deployment on high-speed links implies the need to process the important quantity of data in near real-time, in order to prevent the spread of novel threats. Therefore, the individual agents do not acquire the data from the network directly, but receive the data already preprocessed, with the level of detail that is appropriate for anomaly-based intrusion detection.

The complete acquisition system is integrated and deployed on university network. We work on improvement of collaboration with upper system layers and on verifying of detected anomalies on network.

## References

- [1] D. Brauckhoff, B. Tellenbach, A. Wagner, M. May, and A. Lakhina. Impact of packet sampling on anomaly detection metrics. In *IMC '06: Proceedings of the 6th ACM SIGCOMM on Internet measurement*, pages 159–164, New York, NY, USA, 2006. ACM Press.
- [2] CESNET, z. s. p. o. Family of COMBO Cards. <http://www.liberouter.org/hardware.php>, 2007.
- [3] Cisco Systems. Cisco IOS NetFlow. <http://www.cisco.com/go/netflow>, 2007.
- [4] J. Mai, C.-N. Chuah, A. Sridharan, T. Ye, and H. Zang. Is sampled data sufficient for anomaly detection? In *IMC '06: Proceedings of the 6th ACM SIGCOMM on Internet measurement*, pages 165–176, New York, NY, USA, 2006. ACM Press.
- [5] S. Northcutt and J. Novak. *Network Intrusion Detection: An Analyst's Handbook*. New Riders Publishing, Thousand Oaks, CA, USA, 2002.
- [6] F. Procházka. *Universal Information Robots a way to the effective utilisation of cyberspace*. PhD thesis, Masaryk University, Brno, 2006.
- [7] M. Rehak, M. Pechoucek, P. Celeda, V. Krmicek, J. Moninec, T. Dymacek, and D. Medvigy. High-performance agent system for intrusion detection in backbone networks. In *Cooperative Information Agents XI*, number 4676 in LNAI/LNCS. Springer-Verlag, 2007.
- [8] M. Rehak, M. Pechoucek, and M. Gregor. Trust Modeling with Context Representation and Generalized Identities. Technical report, Gerstner Laboratory, CTU in Prague, 2007.
- [9] D. Šišlák, M. Rehak, M. Pechoucek, M. Rollo, and D. Pavlíček. A-globe: Agent development platform with inaccessibility and mobility support. In *Software Agent-Based Applications, Platforms and Development Kits*, pages 21–46, Berlin, 2005. Birkhauser Verlag.
- [10] Sourcefire, Inc. SNORT – Intrusion Prevention System. <http://www.snort.org/>, 2007.