

High-Performance Agent System for Intrusion Detection in Backbone Networks

Martin Reháček¹, Michal Pěchouček¹, Pavel Čeleda², Vojtěch Krmíček²,
Jiří Moninec², Tomáš Dymáček², and David Medvigy¹

¹ Department of Cybernetics and Center for Applied Cybernetics,
Faculty of Electrical Engineering, Czech Technical University in Prague
Technická 2, 166 27 Prague, Czech Republic
{mrehak,pechouc}@labe.felk.cvut.cz

² Institute of Computer Science, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
{celeda,vojtec,moninec}@ics.muni.cz

Abstract. This paper presents a design of high-performance agent-based intrusion detection system designed for deployment on high-speed network links. To match the speed requirements, wire-speed data acquisition layer is based on hardware-accelerated NetFlow like probe, which provides overview of current network traffic. The data is then processed by detection agents that use heterogenous anomaly detection methods. These methods are correlated by means of trust and reputation models, and the conclusions regarding the maliciousness of individual network flows is presented to the operator via one or more analysis agents, that automatically gather supplementary information about the potentially malicious traffic from remote data sources such as DNS, whois or router configurations. Presented system is designed to help the network operators efficiently identify malicious flows by automating most of the surveillance process.

1 Introduction

With the increasing number of Internet users, provided services and current generation of online multimedia services, the speed of most Internet links has increased significantly. This increase rendered many past methods for intrusion detection obsolete, and current backbone networks lack efficient technology for real-time detection of malicious traffic. While it may be technically possible to perform traffic analysis by means of existing signature matching techniques running on dedicated high-performance hardware, the high rate of false positive, and high cost of associated human supervision makes systematic surveillance uneconomical.

To address the above situation, and to enable the operators of backbone and large enterprise networks to analyze current threats in near real-time, we present a design of an autonomous system able to detect malicious traffic on high-speed networks and to alert the operators efficiently. 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 is essential for project success. This is due to the fact that the traditional agent techniques are not well suited for efficient low-level traffic processing.

In the work presented in this paper, we aggregate the network data to capture the information about network flows, unidirectional components of TCP connections (or UDP, ICMP equivalent) identified by shared source and destination addresses and ports, together with the protocol, and delimited by the time frame used for data acquisition (see Section 3.1). This information provides no hint about the content of the transmitted data, but by detecting the anomalies in the list of flows acquired over the monitoring period, we can detect the anomalies and possible attacks, albeit with limited effectiveness.

2 Related Work

In order to detect an attack from the flow information on the backbone level, especially without any feedback from the affected hosts, we have to analyze the patterns in the traffic data, compare them with normal behavior and conclude whether the irregularity corresponds to a known attack profile or not. This approach to Network Intrusion Detection, typically based on the flow information captured by network flow monitor is currently an important field of research into *anomaly based intrusion detection*. Numerous existing systems, based on traffic volume analysis modeled by Principal Component Analysis (PCA) methods [1], models of entropy of IP header fields for relevant subsets of traffic [2,3], or just count of the flows corresponding to the selected criteria [4] offer each a particular valid perspective on the network traffic.

The MINDS system [4] represents the flow by basic NetFlow aggregation features (srcIP, srcPrt, dstIP, dstPrt, protocol) and complements them by the number of the flows from the same srcIP, to the same dstIP and their combinations with dstPrt and srcPrt respectively. These properties are assessed both in time and number of connections defined windows, to account for slow scanning. The system proposed by Xu *et al.* [2] for traffic analysis on backbone links also uses the NetFlow based identity 5-tuple. The context of the single connection is defined by the normalized entropy of srcPrt, dstPrt and dstIP dimensions of the set of all connections from the srcIP of the flow in the current time frame. Another perspective anomaly detection mechanism can be based on the observation of traffic volumes in high-speed network. Lakhina *et al.* [5] uses statistical modeling to identify the anomalous origin-destination-aggregated flows. The method is based on Principal Component Analysis. In another work of the same authors [3], the PCA method is used to model the normal and residual entropy, to remove the systematic elements of the data before clustering. The clustering then emphasizes the anomalous characteristics of the traffic.

The above listed systems are appropriate for direct deployment on backbone network, with a specific limitations of MINDS that was designed to protect

a large, open network common in university environment. Besides the above-mentioned sample of backbone anomaly detection mechanisms, there are numerous research and commercial systems designed to protect local networks. A typical representative of recently developed system is a SABER [6], which addresses not only threat detection, but attempts to actively protect the system by automatically generated patches. Technical perspective on many existing IDS systems, including SNORT [7] and other *signature matching* techniques that detect intrusions by detecting patterns specific to known attacks in network traffic, can be found in [8]. A good, even if slightly outdated review of classic research and systems in the domain is provided by [9].

3 Architecture

In our approach, we have decided not to develop a novel detection method, by rather to integrate several methods [1,2,3,4] with an extended trust models of a specialized agent. This combination allows us to correlate the results of the used methods and to combine them to improve their effectiveness. Most anomaly detection methods today are not fit for commercial deployment due to the high ratio of false positives (legitimate traffic classified as malicious) or false negatives (malicious traffic classified as legitimate). While their current level of performance is a valid scientific achievement, the costs associated with supervision of such systems are prohibitive for most organizations. Therefore, our main research goal is to combine the efficient low-level methods for traffic observation, with multi-agent detection process to detect the attacks with comparatively lower error rate see Table 2, and to provide the operator with efficient incident analysis layer presented in Section 3.3. This layer supports operator's decisions about detected anomalies by providing additional information from related data sources. The layer is also responsible for visualization of the anomalies and the detection layer status.

The architecture consists of several 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 wire-speed during the network traffic acquisition and preprocessing, the higher layers will use the preprocessed data to infer the conclusions regarding the degree of anomaly and consecutively also the maliciousness of the particular flow or a group of flows. 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. System can be split into these layers, as shown in Figure 1:

- **Traffic Acquisition and Preprocessing Layer:** The components in this layer acquire the data from the network using the hardware accelerated NetFlow probes [10] and perform their preprocessing. This approach provides the real-time overview of all active unidirectional connections on the observed link. In order to speed-up the analysis of the data, the preprocessing layer will aggregate meaningful global and per-flow (or group of) characteristics and statistics.

• **Cooperative Threat Detection Layer:** This layer will principally consist of specialized, heterogeneous agents that would seek to identify the anomalies in the preprocessed traffic data by means of their extended trust models [11]. Their collective decision regarding the degree of maliciousness of a flow with certain characteristics shall use a reputation mechanism. The agents will run inside the A-globe agent platform [12] and will 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 that will help the operator to analyze the output of the detection layer, by putting the anomaly information in context of other relevant information. When the detection layer detects suspicious behavior on the network it is reported to Mycroft. Mycroft opens the new case and retrieves relevant information from available data sources. Network operator can explore and evaluate the reported case subsequently. Another part of this layer is a set of lightweight, specialized visualization agents that will allow the operator to follow only the selected characteristics of the system.

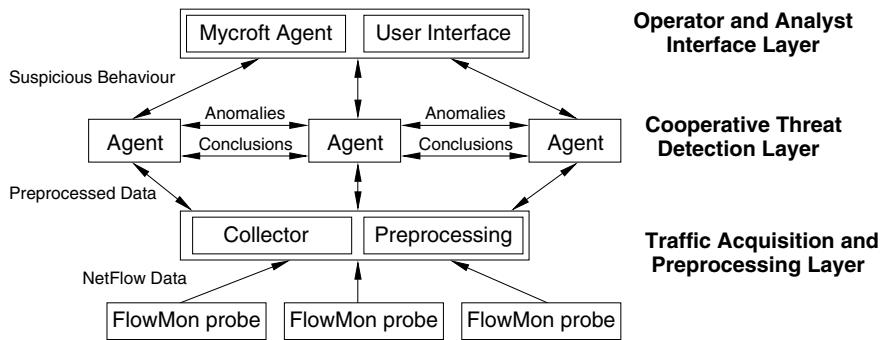


Fig. 1. System overview, with network probes, acquisition and preprocessing layer at the bottom, agent platform for anomalies detection in the middle and visualization on the top

3.1 Traffic Acquisition and Preprocessing Layer

The traffic acquisition and preprocessing layer is responsible for acquiring network traffic, preprocessing data and providing traffic characteristics to upper system layers. We use the flow characteristics, based on information from packet’s headers.

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. For the flow monitoring we use NetFlow protocol developed by Cisco Systems [13].

The amount of traffic in nowadays high-speed networks increases continuously and traffic characteristics change heavily in time (network throughput fluctuation due to time of day, server backups, DoS attacks, scanning attacks, etc.). 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 (see Figure 2). FlowMon probe is a passive network monitoring device based on the COMBO hardware [10], which provides high performance and accuracy. The FlowMon probe is preferred due to implemented features which contains packet/flow sampling, several sampling types, active/inactive timeouts, flow filtering, data anonymization, NetFlow protocol version 5 and 9 support. The FlowMon probe handles 1 Gb/s traffic at line rate in both directions and exports acquired NetFlow data to different collectors. Detailed evaluation of these crucial capabilities is described in Section 4.



Fig. 2. FlowMon - COMBO6X PCI-X 64/66MHz hardware accelerated card [14]

The collector stores incoming packets with NetFlow data from FlowMon probes into database. The collector provides interface to graphical and text representation of network traffic, flow filtration, aggregation and statistics evaluation, using source and destination IP addresses, ports and protocol.

To process acquired IP flows by upper system layers the preprocessing must be performed on several levels and in different manners. Packets can be sampled (random, deterministic or adaptive sampling) on input and the sampling information is added to NetFlow data. On the collector side the same flows are aggregated to reduce the amount of data without information loss and several statistic characteristics (average traffic values, entropy of flows) are computed.

Even after their deployment in monitored network, the probes can be reprogrammed to acquire new traffic characteristics. The system is fully reconfigurable and the probes can adapt their features and behavior to reflect the changes in the agent layer. As we can see in Section 4, presented solution can acquire unsampled flow data from even very fast links. The proposed traffic acquisition and preprocessing layer is able to provide real-time traffic characteristics to detect anomalies by upper system layers.

3.2 Cooperative Threat Detection Layer

Cooperative threat detection is based on the principles of trust modeling [15] that are an integral part of agent research. However, there are three important features [11] that must be added to trust modeling to cover our domain-specific requirements:

- **Uncertain Identity Modeling:** Baseline trust models evaluate the behavior of individual agents, whose identity is guaranteed (to an extent) by the multi-agent platform or similar computational environment. In the network domain, we have to evaluate the trustfulness of network flows, and while they can be distinguished as unique identities, this distinction is unpractical from the intrusion detection perspective. We represent the connections in a metric space, and use the associated distance function to assess the similarity of the flow representations in this space. All detection agents use the same NetFlow 5-tuple to construct the identity representations, but may obtain different results regarding the similarity due to the use of different distance functions. For example, one agent can emphasize the similarity of srcIP address (and likely host), while the others may concentrate on ports that are more application specific. This variability makes the agent perspective on the system multi-faceted, and the attacks are less likely to avoid multiple agents.
- **Context Modeling:** Merely representing the flow identities in the metric space and evaluating their trustfulness gives unsatisfactory results, as it ignores the most important information from the NetFlow data – the information about the other, similar flows in the current traffic sample. This information constitutes the context of the trusting decision [16], and together with the identity defines the Identity-Context metric space, where the detection agents assess the trustfulness of flow representations. Each of the agents uses its own particular context space, typically based on the existing anomaly detection methods. For instance, we can complement the information about the flow by the number of flows from the same srcIP and same dstPrt, or with an entropy of dstIP addresses computed over all flows with the same srcIP. The use of this information is twofold; each agent uses it to place the flow representations in the Identity-Context space of its trust model, and in the same time to provide the information about the degree of flow anomaly to other trusting agents.
- **Implicit Feedback Mechanism:** The principal input of classic trust models is a result of past cooperations with the partner: quality of service, degree of success, on-time delivery and other domain specific parameters. In our case,

the system is deployed on backbone network and it is very difficult to obtain the feedback that can be associated with the current traffic on the network; cooperative IDS (like `dshield.org`) typically provide unsynchronized feedback, and not all the threats are Internet-wide. Obtaining the feedback from the connected operators or organizations is even more difficult: while the IETF has several working groups focusing on incident response interoperability, the bulk of the work is not suitable for real-time data processing, and concentrates on human-to-human interaction. Therefore, we use the information regarding the flow anomaly *as assessed by the other agents* to replace the direct feedback, therefore connecting the anomaly detection between diverse agents.

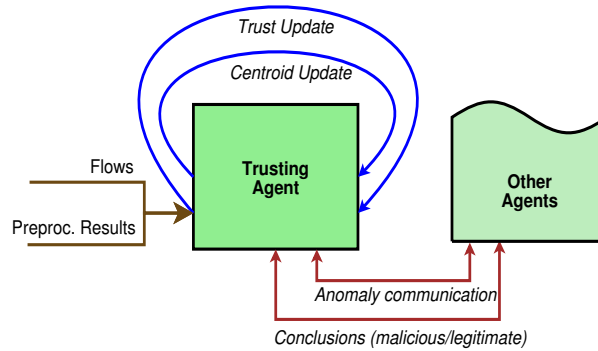


Fig. 3. Overview of detection (trusting) agent operations

While processing the information about the network flows (see Fig. 3), each trusting agent receives an identical copy of network flows list and associated pre-extracted statistics. Then, it uses its specific preprocessing to determine the anomaly of each flow (in most cases working only with already extracted statistics) and to communicate the list of anomalies to other agents. In its turn, the agent also receives the anomalies from the others, and starts the flow processing by its internal trust model. As we have implicitly suggested above, the trustfulness is not associated to individual flows, but rather to selected objects in the Identity-Context space. Individual flow is therefore represented by its identity (i.e. NetFlow 5-tuple) and the associated context is retrieved to determine its position in the Context subspace. Then, we retrieve the positions of nearby centroids from the current trust models and update their trustworthiness with an aggregated degree of flow anomaly as determined by the other agents. When there is no appropriate cluster in the vicinity of the observed flow, a new cluster is created. The details of the approach are presented in [11].

Performance of an isolated detection agent would be the same as a performance of the anomaly detection method it is based on. As we have suggested above, the agents base their evaluation of trustfulness not only on their local results, but also on the anomaly opinions of other agents (see Fig. 4). We argue that

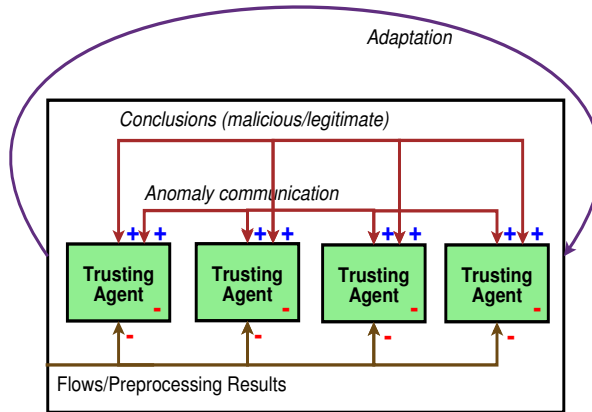


Fig. 4. Overview of collaboration between detection agents

this cross-correlation will help to filter-out most false positives on the level of individual agents, reducing the number of incidents to evaluate with other agents. In the second phase of evaluation, each agent selects the flows it considers as malicious and shares these flows with other agents. Agents then use a simple voting protocol to reach a collective conclusion regarding the estimated maliciousness of anomalous flows, further reducing the number of incidents reported. Collectively accepted flows are then sent to analyst interface layer for further filtering based on user preferences, and possibly assisted analysis by analyst.

The decision whether a given flow is trusted or untrusted depends on the typical degree of anomaly in the observed network traffic. This parameter varies widely with network type – the number of anomalies is typically low on corporate or government networks, but is significantly higher on public Internet backbone links, or in the university settings. To avoid the problems with manual tuning of the system, we use a fuzzy-inference process integrated with the trust model to decide whether the given flow is malicious, by computing its inference with Low and High trust values. These values are determined similarly to the trustfulness of individual flow representations, but are represented as two fuzzy intervals [17].

3.3 Operator and Analyst Interface Layer

The Cooperative Threat Detection Layer is coordinated by a super-agent Mycroft. Mycroft is again a multi-agent system. This system is constructed for context based inference over information synthesized from various data sources. The name of this system refers to the so called Mycroft problem well known from Doyle’s Mycroft Holmes - a Sherlock Holmes’ brother. Every detected suspicious behavior on the network is reported to the agent Mycroft by the detection layer. Mycroft opens the new case subsequently and retrieves relevant information from data sources to which it is connected. Then the network operator can explore and evaluate the reported case together with contextual information retrieved

from connected data sources. Using of this intelligent multi-agent system brings the advantage of wide adaptation:

- adaptation to the guarded network (Where is the detection performed),
- adaptation to the detection layer status (How is the detection performed),
- adaptation to suspicious behavior on the network (What is the subject of detection),
- adaptation to the given network operator (Who supervises detection),
- adaptation to the purpose of interaction with operator (Why are detection results reported).

All these adaptations are possible thanks to the following construction: Individuals are classified into categories to express their properties. The individual can be any object relevant to the suspicion detection or any elementary relation between such objects relevant to the suspicion detection. Individuals are classified into categories with a measure from the interval $\langle -1, 1 \rangle$. Value -1 stands for "certainly not in given category", value 1 stands for "certainly is in given category" and value 0 stands for "cannot decide". This classification is called the elementary fact and is realized always in some context. The context is nothing else than a set of elementary facts. All the information retrieved from data sources is disassembled into elementary facts. This approach allows data utilization in a way that fits actual situation. Presented construction is a straight extension of the so called diamond of focus presented in [18] for the first time. Formal definition of systems referred to as Knowledge and Information Robots is provided in [19]. Multi-agent system Mycroft is one of the representatives of Knowledge and Information Robots class.

• **Adaptation to the Guarded Network:** Mycroft uses available data sources to support network operator's evaluations and decisions. The system can be taught using quite formalized natural language what kind of information given data source contains, how to combine this information with all other information that Mycroft already knows or is capable to acquire and how to access this data source. Mentioned teaching is to all intents and purposes done without programming. Summary of typical data sources is listed in Table 1. Communication with data sources is realized by the set of adapters for various types of data sources. This use-case saves operator's time and work in the decision making process.

• **Adaptation to the Detection Layer Status:** Multi-agent system can adapt to changes in the detection layer. It monitors the current status of the detection layer and of each agent present in the detection layer. It is also able to communicate with each agent. If the settings of detection layer change, the multi-agent system Mycroft can deal with it using rules or can be simply re-taught to be ready for cooperation with new detection layer configuration. Interaction with the detection layer is shown in Figure 5.

• **Adaptation to Suspicious Behavior on the Network:** Multi-agent system can adapt to various types of suspicious behavior on the network. When suspicious behavior on the network is reported by the detection layer, Mycroft

Table 1. Table of data sources: first column contains data source description, second column contains importance for network operator (1 – most important, 3 – less important)

<i>Topology</i>		<i>User information</i>	
Routing tables of network hardware	1	Users allowed to log on target computers	1
Trace route utility	2	Users currently present in room with computer	2
Active network hardware configuration files	1	Users currently logged on target computer	1
SNMP information	2	Profiles of typical user behavior	3
Routing tables of computers	3		

<i>Rules and policy</i>		<i>Available services</i>	
Firewall configuration	1	Port scan utility	2
NAT configuration	1	OS detection utility	3

<i>Address resolution</i>		<i>Administrative information</i>	
DNS configuration	1	Computer configuration	1
DHCP address assignment	2	Ports vs. typical services database	2
ARP tables	2	Physical location of device	1
WHO IS service	2	Known attacks database	2

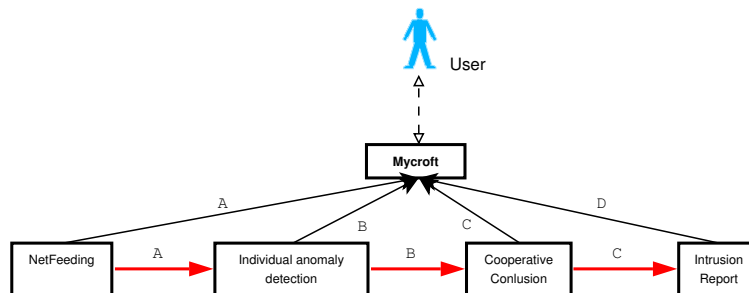


Fig. 5. Mycroft – detection layer interaction, Mycroft observes detection layer behavior **A** - NetFlow data and statistics, **B** - Each Agent’s individual anomaly detection, **C** - Agent’s negotiation and conclusions, **D** - Intrusion detection reports

asks connected data sources to get relevant information for given type of suspicious behavior. A pattern matching based transmutation is used to select the relevant information.

Using additional knowledge provided by these data sources Mycroft can perform basic evaluations of the traffic situation and present this evaluations to the network operator concurrently with the detection report. In some basic cases, using pre-learned transmutation patterns, system can even evaluate this

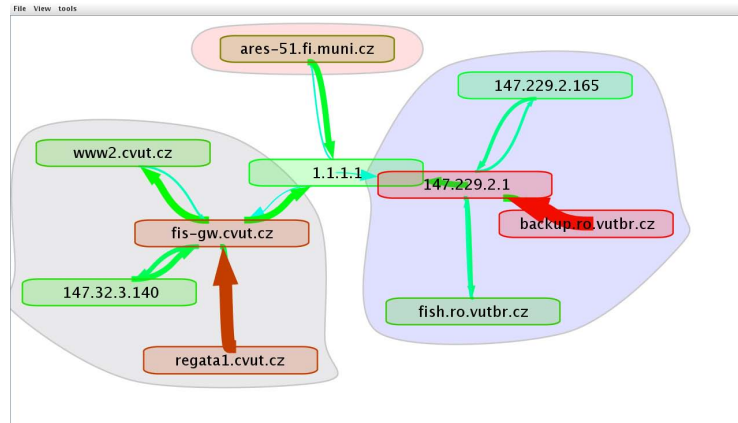


Fig. 6. Flow data visualization example. Oval nodes in graph represent network devices identified by IP addresses. Arrows show direction of the flows.

suspicion as false positive and drop it. Again this is possible by the means of the knowledge provided by suitable data sources.

If a new type of suspicious behavior on the network reveals, the system can deal with it using rules or can be recaught. There is no need of reprogramming.

- **Adaptation to Purpose of Interaction with the Operator:** There are various types of purposes of interaction with the operator (decision making request, notification, warning, etc.). Multi-agent system can adapt to these types of purposes and provide suitable interaction with the network operator.

- **Adaptation to Given Operator:** It means that multi-agent system is able to follow operator's procedures, habits and preferences. It learns during routine work. When operator is not satisfied he can ask the system for different presentation of given information.

Mycroft uses suitable visualization to support visual analytics and decision-making process. It makes the interaction between operator and the whole system efficient and helps operator to explore visual patterns while exploring the data. The visualization and interaction with operator approach is very close to Human-Centered Computing approach [20].

Mycroft uses various types of graphs as visual representation of network traffic situations (see Figure 6) with dynamic user control techniques:

- level of detail selection,
- filtering,
- detail on demand showing.

Level of detail selection allows network operator to explore given data on the level of subnets, particular IP addresses, ports or on the detailed level of individual flows. Filtering allows operator to filter out unimportant information. The operator can focus just on particular subnet or port and on the flows with

given properties. Operator can also use detail on demand focus to get additional relevant information from available data sources.

4 System Evaluation and Performance

The performance is becoming a key concern of Network Intrusion Detection Systems (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.

The Figure 7 shows flow statistics for various IP packet's sizes transmitted on Ethernet at a rate of a gigabit per second. The tests were performed with the Spirent AX/4000 broadband test system [21]. The test system generated 5 flows with 500000 packets per flow for each packet size. The results of FlowMon probe correspond to the maximal theoretical throughput on gigabit Ethernet network. On the other hand the results of software NetFlow probe (nProbe [22]) are misrepresented for small IP packets. The nProbe was used on Linux OS with Intel PCI-X network interface card and default kernel configuration.

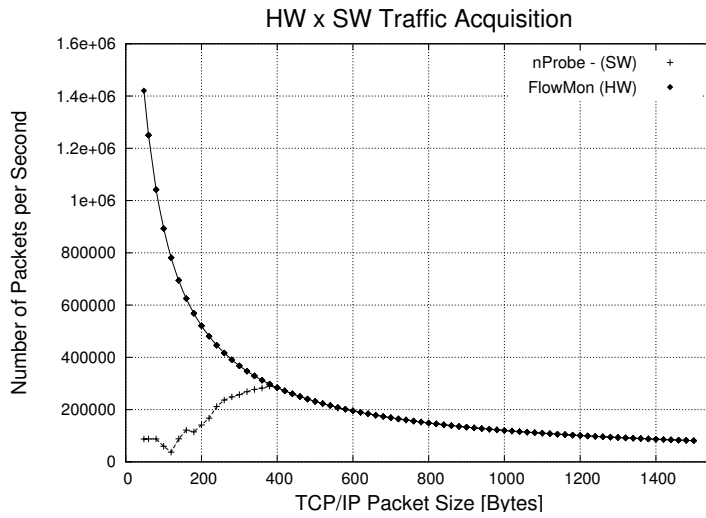


Fig. 7. Flow statistics acquired by SW and HW accelerated probe

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 exported flows are sampled or limited to maximal number of exported flows per second e.g. 5000 flows/s. In comparison with FlowMon probe, user defined extensions can't be added to the routers and the adaptability is very limited.

In our work we are focusing on the impact of packet sampling on anomaly detection. The articles [23,24] study whether existing sampling techniques distort traffic features that are critical for effective anomaly detection. They show that packet sampling methods (random packet sampling, random flow sampling, smart sampling, and sample-and hold sampling) 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.

Table 2. Multi-agent system performance overview. The system process backbone link traffic with average load of 800 Mb/s.

<i>Layer</i>	<i>Processed Data</i>	
	<i>Input</i>	<i>Output</i>
Operator	Security Incidents - incidents at a certain priority levels	Incident Handling - resolving up to 10 high priority incidents per hour
Operator and Analyst Interface Layer	Network Anomalies - detected threats with additional network information	Detected Incidents - priority-based incidents up to 100 incidents/minute
Cooperative Threat Detection Layer	Network Traffic Statistics - aggregated flow statistics up to 100000 flows/minute	Detected Threats - network traffic anomalies up to 10000 threats/minute
Traffic Acquisition and Preprocessing Layer	Network Traffic - packets 125000 packets/s	Flow Statistics - flows 3800 flows/s

The Table 2 shows the performance of multi-agent system. The observed network traffic is processed by several layers to handle high amount of data in current networks. Each layer has specific physical and performance limits e.g. number of incidents which can be handled by human operator. To overcome such limitations, the system is fully scalable and all layers can be distributed. The multi-agent system adapts the detection behavior to reduce the number of false positives and negatives so the final number of incidents fits the limits of human operator.

5 Conclusions and Future Work

Our work presents a design of multi-agent system for network intrusion detection that is optimized for deployment on backbone networks. The designed system addresses two main limitations of existing intrusion detection systems – efficiency and effectiveness. 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. Each detection

agent in the system is based on existing anomaly detection technique, which defines its perception of network flow identities in its trust model. Its private view of the data is complemented by the opinions of other agents regarding the anomaly of flows in the current traffic, therefore collaboratively improving the effectiveness of anomaly detection process. When the agents reach a conclusion regarding the untrustfulness of a particular subset of flows, they submit this observation to user-interface agent that automatically retrieves context information (DNS records, history, etc.) to allow rapid analysis by human supervisors.

At the time of this writing, the first preliminary version of the complete system is being integrated and the whole system is still under active development. Therefore, we don't present any definitive experimental results regarding its effectiveness of the complete system, but only the performance evaluations of critical components at the current integration stage. The data used for system testing are acquired on Masaryk University network, connected to the Czech national educational network (CESNET). In our future work, we will provide detailed experimental evaluation of system deployment, and also analyze its performance in countering a wide selection of currently observed malicious traffic.

Acknowledgment. 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 grants 1M0567 and 6840770038.

References

1. Lakhina, A., Crovella, M., Diot, C.: Characterization of Network-Wide Anomalies in Traffic Flows. In: ACM SIGCOMM conference on Internet measurement IMC '04, pp. 201–206. ACM Press, New York (2004)
2. Xu, K., Zhang, Z.L., Bhattacharya, S.: Reducing Unwanted Traffic in a Backbone Network. In: USENIX Workshop on Steps to Reduce Unwanted Traffic in the Internet (SRUTI), Boston, MA (2005)
3. Lakhina, A., Crovella, M., Diot, C.: Mining Anomalies using Traffic Feature Distributions. In: ACM SIGCOMM, Philadelphia, PA, pp. 217–228. ACM Press, New York (2005)
4. Ertoz, L., Eilertson, E., Lazarevic, A., Tan, P.N., Kumar, V., Srivastava, J., Dokas, P.: MINDS - Minnesota Intrusion Detection System. In: Next Generation Data Mining, MIT Press, Cambridge (2004)
5. Lakhina, A., Crovella, M., Diot, C.: Diagnosis Network-Wide Traffic Anomalies. In: ACM SIGCOMM '04, pp. 219–230. ACM Press, New York (2004)
6. Sidirolou, S., Keromytis, A.D.: Countering network worms through automatic patch generation. *IEEE Security & Privacy* 3, 41–49 (2005)
7. Sourcefire, Inc.: Snort- Intrusion Prevention System (2007) Accessed in (January 2007), <http://www.snort.org/>
8. Northcutt, S., Novak, J.: Network Intrusion Detection: An Analyst's Handbook. New Riders Publishing, Thousand Oaks, CA, USA (2002)

9. Axelsson, S.: Intrusion detection systems: A survey and taxonomy. Technical Report 99-15, Chalmers Univ (2000)
10. CESNET, z. s. p. o.: Family of COMBO Cards. (2007), <http://www.liberouter.org/hardware.php>
11. Reháček, M., Pěchouček, M., Gregor, M.: Trust Modeling with Context Representation and Generalized Identities. Technical report, Gerstner Laboratory, CTU in Prague (2007)
12. Šišlák, D., Reháček, M., Pěchouček, M., Rollo, M., Pavlíček, D.: A-globe: Agent development platform with inaccessibility and mobility support. In: Unland, R., Klusch, M., Calisti, M. (eds.) Software Agent-Based Applications, Platforms and Development Kits, pp. 21–46. Birkhauser Verlag, Berlin (2005)
13. Cisco Systems: Cisco IOS NetFlow (2007), <http://www.cisco.com/go/netflow>
14. Čeleda, P., Kováčik, M., Konří, T., Krmíček, V., Špringl, P., Žádník, M.: FlowMon Probe. Technical Report 31/, CESNET, z. s. p. o (2006) (2006), <http://www.cesnet.cz/doc/techzpravy/2006/flowmon-probe/>
15. Sabater, J., Sierra, C.: Review on computational trust and reputation models. *Artif. Intell. Rev.* 24, 33–60 (2005)
16. Reháček, M., Gregor, M., Pechoucek, M., Bradshaw, J.M.: Representing context for multiagent trust modeling. In: IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2006 Main Conference Proceedings) (IAT'06), pp. 737–746. IEEE Computer Society, Los Alamitos (2006)
17. Reháček, M., Foltýn, L., Pěchouček, M., Benda, P.: Trust Model for Open Ubiquitous Agent Systems. In: Intelligent Agent Technology, 2005 IEEE/WIC/ACM International Conference. Number PR2416, IEEE, Los Alamitos (2005)
18. Staníček, Z.: Universal Modeling and IS Construction. PhD thesis, Masaryk University, Brno (2003)
19. Procházková, F.: Universal Information Robots a way to the effective utilisation of cyberspace. PhD thesis, Masaryk University, Brno (2006)
20. Jaimes, A., Gatica-Perez, D., Sebe, N., Huang, T.S.: Human-centered computing: Toward a human revolution. *Computer* 40, 30–34 (2007)
21. Spirent, C.: Spirent AX/4000 Broadband Test System (2007), <http://www.spirentcom.com/>
22. Deri, L.: nProbe - An Extensible NetFlow v5/v9/IPFIX GPL Probe for IPv4/v6 (2007), <http://www.ntop.org/nProbe.html>
23. Mai, J., Chuah, C.N., Sridharan, A., Ye, T., Zang, H.: Is sampled data sufficient for anomaly detection? In: IMC '06: Proceedings of the 6th ACM SIGCOMM on Internet measurement, pp. 165–176. ACM Press, New York (2006)
24. Brauckhoff, D., Tellenbach, B., Wagner, A., May, M., Lakhina, A.: Impact of packet sampling on anomaly detection metrics. In: IMC '06: Proceedings of the 6th ACM SIGCOMM on Internet measurement, pp. 159–164. ACM Press, New York (2006)