Krzysztof Cabaj, Zbigniew Kotulski, Paweł Szałachowski
Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska

Grzegorz Kołaczek
Instytut Informatyki
Politechnika Wrocławska

Jerzy Konorski
Wydział ETI
Politechnika Gdańska

Implementation and testing of Level 2 security architecture for the IIP System

The IIP System (IIPS) defines a number Parallel Internets (PIs), one IPv6-based, two post-IP and a management network, running over a common physical substrate. It has a four-level architecture, with Level 2 responsible for creation of virtual resources of the PIs. This paper elaborates on a three-tier security architecture proposed earlier to address Level 2 threats of alien traffic injection and IIPS traffic manipulation or forging. Early experience with the implementation of the security architecture is reported and preliminary experiments carried out in a small-scale IIPS testbed are discussed.

1. Introduction

The Future Internet Engineering project under the name IIP [1] envisages a physical substrate shared by four Parallel Internets (PIs), together constituting a communication infrastructure named the IIP System (IIPS) serving to exchange IIPS protocol data units (IIPS-PDUs). Each PI uses virtualized links and nodes to define its own protocol stack. Two PIs are post-IP solutions named Data Stream Switching (DSS), and Content Aware Network (CAN), the third is IPv6 QoS oriented, and the fourth (MGT) is the IIPS management network. The IIPS architecture consists of four Levels, where Level 2 is responsible for creation of PI virtual links and nodes.

Level 2 security measures are proposed to address an external intruder manipulating IIPS traffic or injecting alien traffic into the IIPS in order to disrupt its functionality, and a virtual IIPS node being compromised by an internal intruder who can forge IIPS traffic to achieve specific harmful goals. Thus conceivable Level 2 attacks include traffic *injection*, *replay/resequencing*, *ruffling* (disruption of IIPS-PDU spacing via frame capturing and hold-up) and *forging* (generation of fake IIPS traffic). In [2], a three-tier Level 2 security architecture was proposed that encompasses the following mechanisms. In the 1st tier, a hash-based message authentication code (HMAC) is appended to all IIPS-PDUs to prevent *injection* and *replay or resequencing*; a HMAC test is performed at each IIPS node upon reception of an IIPS-PDU. In the 2nd tier, anomalous traffic or node behavior caused by ruffling or forging attacks is detected by observing appropriately defined *security-relevant events* (SREs). SREs are analyzed by a local anomaly detection (LAD) module, acting as part of a nodal Local Security Agent (LSA). Anomalies indicative of attacks, translated into local reputation metrics, are reported to the 3rd tier, at which a PI's Master Security Agent (MSA) derives the level of trust that can be placed in each PI node. Using a PI-wide anomaly detection (PI-AD) module, the MSA also captures anomalies of a larger scope.

In this paper, based on our early experience with the integration of the IIPS Level 2 security architecture with the IIPS management network, we present a few notes on the implementation and preliminary testing of selected security mechanisms in the 2nd and 3rd tiers.

2. Integration Testbed at The National Institute of Telecommunication

The prototypes of all software security modules, namely LAD, LSA, PI-AD and MSA are integrated with modules provided by the project group developing the MGT PI into a testbed localized at the National Institute of Telecommunication (see [2] for a preliminary report). For testing purposes, four IIPS machines are used. Three of them are IIP system nodes with LAD and LSA modules installed. The fourth one is used as a central node, in which the PI-AD and MSA modules, as well as management monitoring server, are installed. All these machines are connected via IPv6-based management network which is logically and physically independent from the other PIs' traffic. All security related communication uses SNMP version 3 (SNMPv3) with the main security features (authentication, authorization and encryption) enabled. Figure 1 presents a logical view of the testbed.

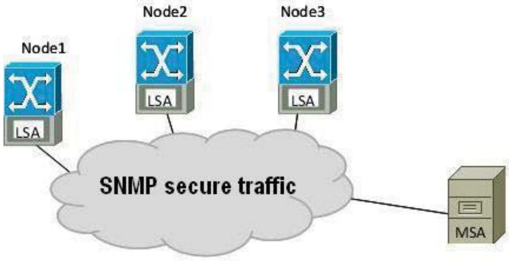
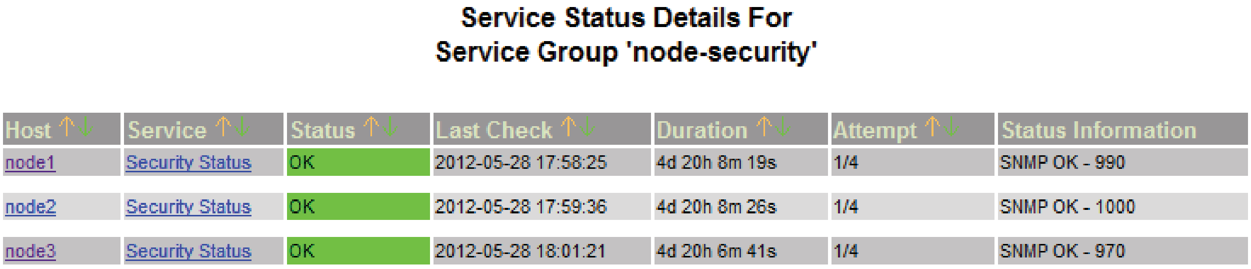


Fig. 1. Security and management integration testbed
at the National Institute of Telecommunication

Integration of the IIPS Level 2 security and management functionality has been achieved in that trust values provided by the MSA module appear in the management monitoring console. Figure 2 presents a monitoring operator's sample view associated with security services. The number at the end of the rightmost column after "SNMP OK" represents the trust value of a given node as a number ranging from 0 to 1000, for example 990 for node1, where 1000 signifies the highest possible trust level.



Service Status Details For
Service Group 'node-security'

Host ↑↓	Service ↑↓	Status ↑↓	Last Check ↑↓	Duration ↑↓	Attempt ↑↓	Status Information
node1	Security Status	OK	2012-05-28 17:58:25	4d 20h 8m 19s	1/4	SNMP OK - 990
node2	Security Status	OK	2012-05-28 17:59:36	4d 20h 8m 26s	1/4	SNMP OK - 1000
node3	Security Status	OK	2012-05-28 18:01:21	4d 20h 6m 41s	1/4	SNMP OK - 970

Fig. 2. Sample operator monitoring view for security services (calculated trust value) generated automatically by the MGT PI software

The integrated prototypes of the LSA and MSA security modules are implemented in Python, whereas for reasons of efficiency required by the underlying data mining and time series analysis algorithms, the LAD and PI-AD modules are implemented in C++. In the following section, interfaces and information flow from LAD modules to the management console is presented. Raw data in which anomalies are discovered are gathered by a LAD module. For this purpose three interfaces are used: SNMP direct access, syslog log messages with predefined attribute values, and a specially developed diagnostic interface between LAD and HMAC modules. The first interface is used by the time series analysis; a LAD module queries local SNMP daemon cyclically for interesting MIB object identifiers (OIDs) associated with system performance counters, such as CPU utilization, or the number of transmitted or erroneous frames. The two remaining interfaces are event-driven, and used for data mining analysis of SREs associated with Linux firewall logs, SNMP proxy logs and HMAC violation events. Logs generated by Linux ip6tables firewall and SNMP proxy are delivered to the LAD module via syslog protocol and a testbed rsyslog program. Additional rules encoded in the rsyslog configuration file redirect interesting logs to the LAD module listening on UDP socket at port 54321. The UDP-based diagnostic interface between HMAC and LAD transports a frame failing the HMAC test to the LAD module. In the current implementation, only the first 64 bytes of the frame, followed by a 3-byte header, are sent.

If any anomalies are discovered in analyzed data, this information is sent to the trust calculation module implemented in LSA. Locally calculated trust values are periodically gathered by MSA for calculation of the global trust level associated with each IIP node. Additionally, the employed data mining algorithms permit LAD to inform MSA about any unusual events. This information is used by PI-AD for detection of various scanning attacks. In any of the above situations, communication is provided via secure SNMP *Inform* messages. Additionally, MSA provides direct access to relevant information for authenticated and authorized users. Using this interface, the Management modules acquire information concerning a given node's global trust level.

3. Data-mining anomaly detection

Data mining anomaly detection is based on frequent sets discovery. A detailed description of used techniques and their advantages in detection of particular attacks that can be performed in the IIPS management network are provided in [2]. Anomaly detection is performed in two stages: locally at each node and globally at the PI-AD module integrated into the MSA module.

Local anomaly detection is performed at LAD module, which for performance reasons is implemented using C++ language. LAD module provides two interfaces for SREs to be analyzed: the standard syslog text protocol and a dedicated diagnostic interface for HMAC hardware and software module, developed within the project. Data received on both interfaces are preprocessed and for later analysis treated as a collection of item sets, each consisting of items represented by integer numbers. Among the item sets corresponding to all SRE received in a given time range, the so-called *frequent sets* are discovered. For example, any detected frequent set contains an item corresponding to an IIPS node address is a sign of a high-traffic attack. An example activity that can produce such a pattern is a massive nmap scanning attack or a brute-force attempt to SNMP password guessing. Similar kinds of activity are directly reported via SNMPv3 to the LSA module, which calculates nodal trust values using the information provided. This process can be tuned by proper adjusting of two detection parameters such as the *window length* and *minimal support*. The former, expressed in seconds, tells how long SREs have to be aggregated before a frequent set discovery. The latter is a criterion for a frequent set discovery, and informs in how many item sets given set of items has to appear. These parameters can be dynamically tuned using the provided management interface, described later.

Upon a frequent set discovery, all item sets corresponding to the SREs collected in a given time window are reviewed, and those that are not associated with any detected frequent set are transferred to the PI-AD module via SNMPv3. All such item sets received from all the IIPS nodes are analyzed once again in a similar manner. During this second stage of analysis, all attacks that are intentionally slowed down or affect more than one IIPS node can be detected. Similarly as in the case of LAD module, the results of the analysis can be adjusted by tuning the window length and minimal support. All anomalies associated with frequent sets detected by the PI-AD module are directly reported to the MSA module as global anomalies.

4. Time series analysis anomaly detection

Anomalous behavior of an IIPS node can be observed in selected features of the node states. To exploit the temporal context of the observations, a time series analysis method has been adopted to detect anomalies manifesting themselves through network traffic and/or nodal resource utilization level. The relevant behavioral features observed at an IIPS node over time are mainly memory and CPU usage and numbers of received and transmitted frames within each Parallel Internet (PI). For example, abnormally high CPU usage or received traffic volume are typical of DoS (in particular, traffic injection) and all-purpose forging attacks, whereas traffic ruffling attacks create abnormal statistics of traffic bursts. The adopted method of time-series analysis extends Burgess' work [3] to fulfill the specific requirements of anomaly detection in the IIP node behavior, in particular to account for simultaneously received traffic from the three PIs. Another novelty is the native integration between the LAD and reputation modules of the IIPS.

LAD performs the proposed time series analysis in three steps. In the feature selection step, relevant behavioral features are selected. In the parameter estimation step, historical (training) data on the selected feature values are compared with the current feature value to learn how indicative the feature is of possible anomalies. A model of nodal behavior is constructed by iterating these two steps. The last step is detection of anomalies, as indicated by a large discrepancy between the statistics of the selected feature values and baseline statistics derived from training data. This is done, among others, by noticing the periodicity in successive feature values and selecting a characteristic period to capture significant correlations between them.

To briefly explain the adopted method, let the feature values be related to received traffic. The considered time series is

$$X = (x_1, \ldots, x_l,, \ldots) \tag{1}$$

where x_l is the number of received bytes in the l^{th} time window. The size of the time window should maximize the accuracy of anomaly detection. This and other parameter values can be configured using receiver operating characteristic (ROC) curves that visualize the performance of anomaly detection under various threshold settings. Two time subseries derived from X are also analyzed by LAD, namely X_P and X_T, with

$$x_{P,l} = \frac{1}{P}\sum_{k=0}^{P-1} x_{l-k}\,, \quad x_{T,l} = \frac{1}{T}\sum_{k=0}^{T-1} x_{l-kP}, \tag{2}$$

where P and T are the averaging intervals (P is called the characteristic period of X). That is, X_P and X_T represent current averages of the feature values over, respectively, the latest characteristic period and a number of recent characteristic periods given a fixed time shift with respect to the period start. For the two time subseries, standard deviations $\sigma_{P,l}$ and $\sigma_{T,l}$ over appropriate averaging intervals are computed with respect to exponential moving averages $\overline{X}_{P,l}$ and $\overline{X}_{T,l}$. Finally, local deviations are expressed as

$$\delta_{P,l} = \left| x_l - \overline{X}_{P,l} \right|, \ \delta_{T,l} = \left| x_l - \overline{X}_{T,l} \right| \tag{3}$$

If the observation of x_l implies a detection of an anomaly, its potential adverse impact is reflected in the following heuristic *severity* measure, further passed by LAD to the reputation agent:

$$c_l = \sqrt{(\delta_{P,l} / \sigma_{P,i})^2 + (\delta_{T,l} / \sigma_{T,i})^2} \Big/ 3\sqrt{2} \tag{4}$$

(c_l = 1 if the right-hand side above exceeds 1). Note that if the current feature value is close to average i.e., agrees with the historical experience, the severity is close to 0, whereas if each of the local differences is treble the corresponding standard deviation, a maximum severity anomaly is inferred. The detected anomaly's complementary attribute called *intensity* is taken to be the average value of severity during the characteristic period of X.

A distinctive property of the IIPS architecture is that an IIPS node concurrently supports four, in general independent, streams of network traffic carried within the four constituent PIs (IPv6 QoS, DSS, CAN and Management). Therefore an important requirement for the implemented anomaly detection method is the ability to recognize anomalies at the global traffic level. This implies that the severity and intensity values have to be calculated taking into account traffic of all four PIs. The global severity combines severities associated with each PI using the following formula:

$$global\_c_l = \frac{\gamma}{4}(\beta_{DSS} DSS\_c_l + \beta_{CAN} CAN\_c_l + \beta_{IPvQoS} IPv6QoS\_c_l + \beta_{MGT} MGT\_c_l) \tag{5}$$

where the PI severities are calculated as above with respect to the traffic from the corresponding PI, the β-coefficients sum up to one and allow to balance the impact of the particular PI on the global severity level, and γ is the number of PIs whose severities have exceeded a predefined threshold. In the example presented below in Fig. 3, the threshold equals 0.3, and for simplicity all the β-coefficients have been set to 0.25. Because of the parameter γ, not only the severity values calculated for a particular PI, but also the number of affected PIs counts for the global severity level (as dictated by the intuitive understanding of "severity"). If multiple PIs are affected by a detected anomaly, the value of the global severity rises accordingly.

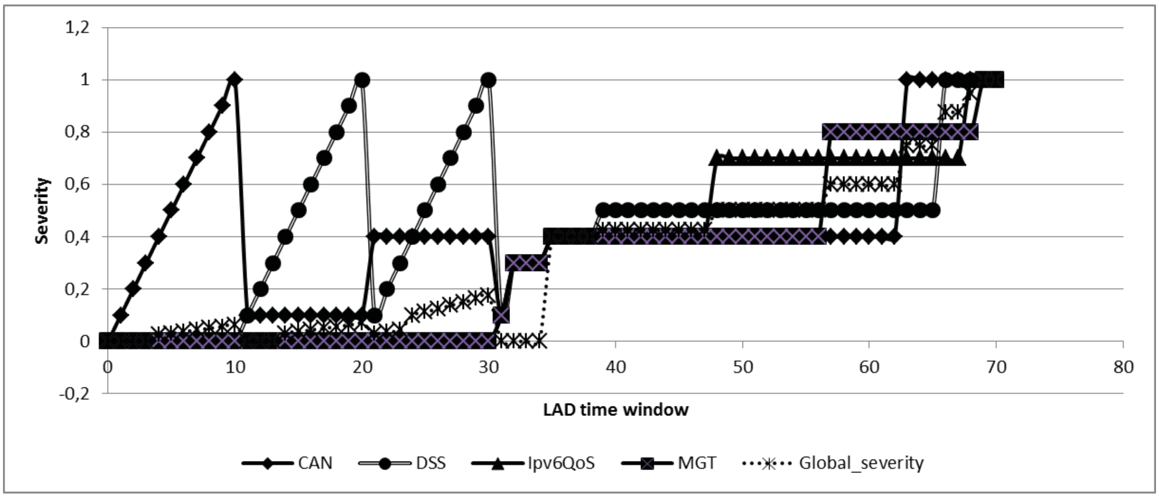


Fig. 3. Global severity level and PI severities

The presented test scenario demonstrates the ability of LAD to detect some types of attacks that potentially arise at IIPS Level 2, referred to as traffic injection and ruffling. As a precondition, the number of received bytes per PI and time window is delivered to LAD every 10 s from a local IIPS management and monitoring module using the SNMP *Get* message with an appropriate OID value. In all experiments, traffic conforming to specific attack characteristics was generated artificially using the D-ITG platform [4].

The first experiment, confined to a single PI, illustrates the traffic injection attack performed by Node3 against Node1 in Fig. 1. Initially, typical traffic is generated and input to Node2, using the Pareto distribution of IIPS-PDU interarrival times with mean set to 15 ms and standard deviation set to 75 ms (which corresponds to the scale and shape coefficients of 10 and 3, respectively), and a constant frame size 5000 B. LAD at Node2 tracks the number of received bytes in successive time windows and creates a typical traffic time series. As a result, zero-severity observations follow. Traffic injection is modeled by having Node3 generate a small volume of additional frames with a normal distribution of interarrival times with mean 20 ms and standard deviation 5 ms, and a constant frame size 500 B. They are injected into the original frame stream after 20 LAD time windows. Node2 then recalculates the time series to reflect the changed traffic volume. As the injected traffic affects more time series elements, the resulting intensity and severity values increase, see Fig. 4. Subsequently, LAD learns the changed traffic pattern and ceases to attribute nonzero intensity and severity to it.

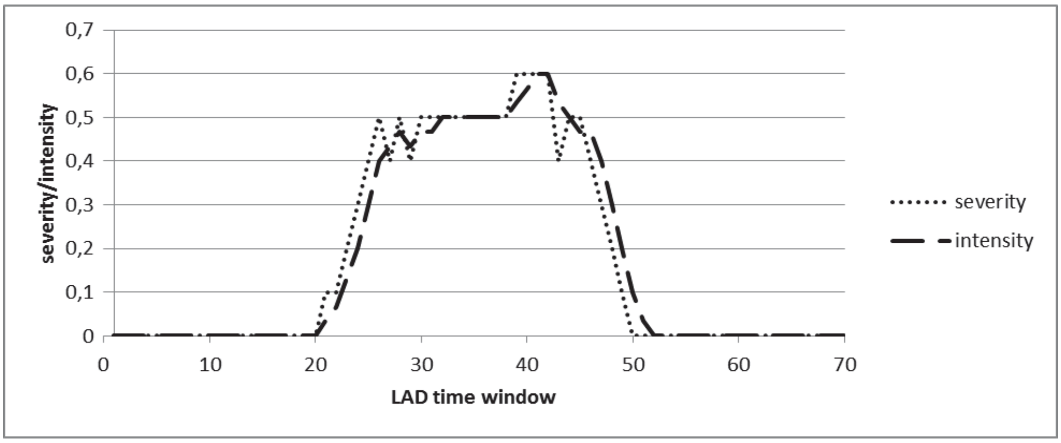


Fig. 4. Detection of traffic injection attack

The next scenario presents the difference between the anomaly detection in aggregated traffic from all four PIs and separate anomaly detection in traffic related to each PI. In Fig. 5 and Fig. 6, the volume of aggregate traffic at an IIPS node is constant. This means that no anomalies would be detected when the LAD analyzed either case globally, even though anomalous behavior is present in Fig. 6, while in Fig. 5 it is not. The situation will change when traffic volumes in each PI are analyzed separately. This time the anomalous behavior in the DSS and IPv6 QoS PIs will be detected between the 30th and 40th LAD time windows in Fig. 6. The global severity level will immediately indicate a security problem at the IIPS node.

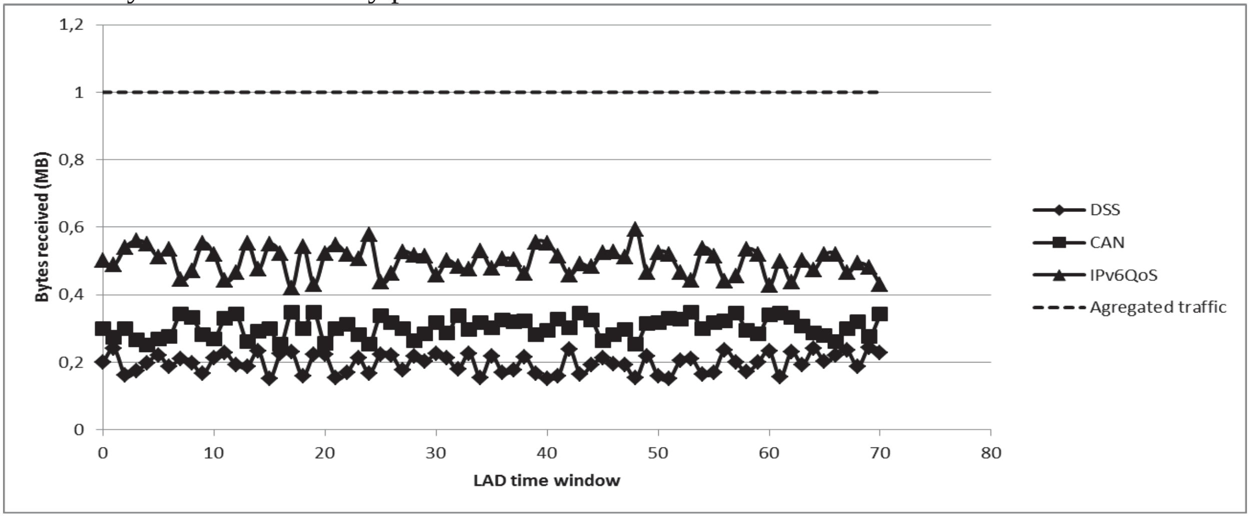


Fig. 5. IIPS node traffic without anomalies

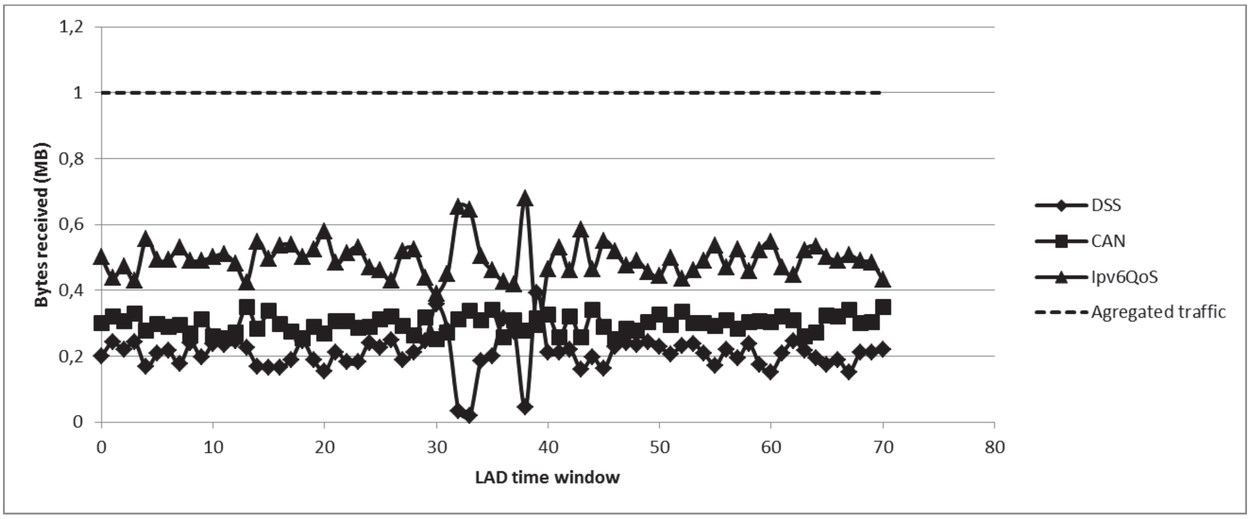


Fig. 6. IIPS node traffic with anomalies in the DSS and IPv6 QoS PIs

Before the method can be applied in the real IIPS, the algorithm needs thorough search for the best configuration parameters values. This can be performed using ROC curves, which plot false-alarm versus detection rates. The false-alarm rate is the number of false anomalies divided by the number of all anomalies that occurred during a test period. The detection rate is the proportion of the correctly detected anomalies to all anomalous events. Each point of the curve is obtained for a given threshold above which a severity value is classified as an anomaly. For example, to select the correct P value to be substituted into (2), five experiments with different P values have been conducted. The experiments used the traffic injection attack scenario described earlier. The obtained ROC curves in Fig. 7 suggest that P should equal 10 s, since the obtained detection rate is the highest and reaches 90% across a range of false-alarm rates. Similar experiments should be performed to optimize the other parameters. As the IIPS is still under development and its traffic characteristics can differ significantly from those in existing systems, the above results based on artificial traffic generation should at present be mainly regarded as a proof of concept.

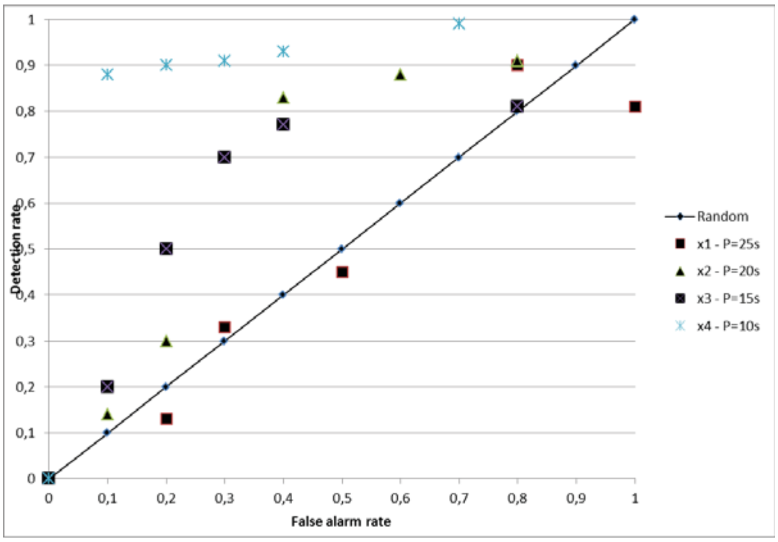


Fig. 7. ROC curves for different *P* values

5. Reputation system

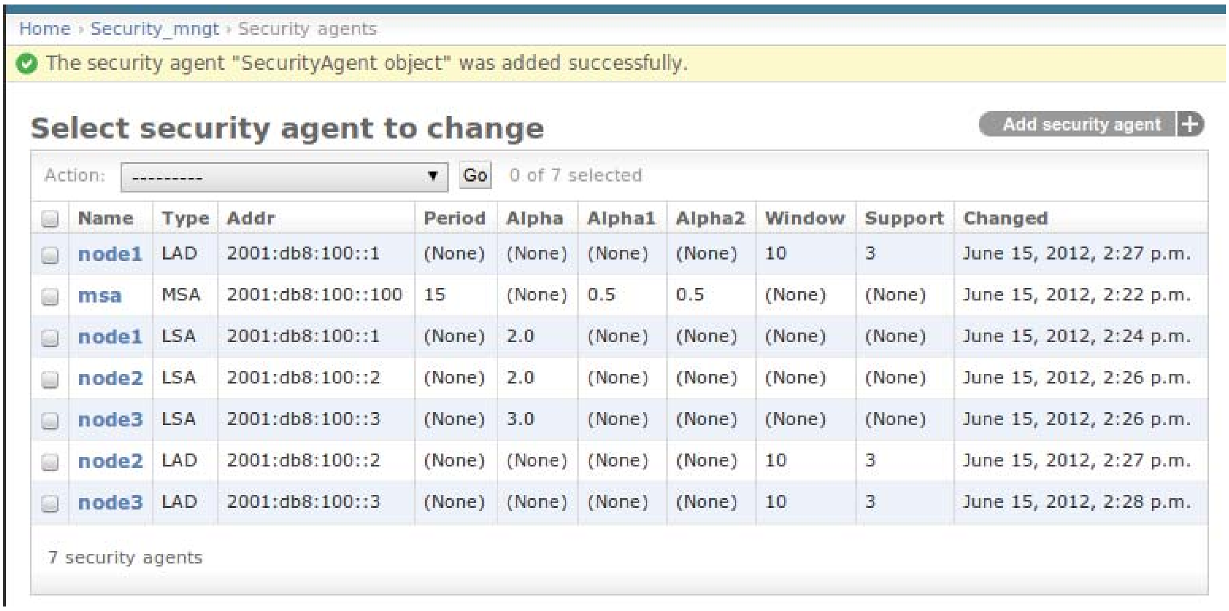
In deciding the type of a reputation system to be implemented in the IIPS security architecture, one can select between the recommendation- and validation-based types. The former type has all network management entities (in this case, the LSAs) recommend trust and reputation values in respect of a certain node according to their subjective experience of interactions with that node. Thus second-hand information is often relied upon. In the latter type, trust and reputation values are calculated according to a LSA's own subjective experience of events and estimation of the threat level they cause the IIPS. Thus first-hand information is only used. We have selected the latter type and adopted a centralized approach whereby LSAs report their experience to the MSA for global trust value calculation. Our approach has two advantages. First, it is insensitive to the numerous attacks and inaccuracies arising in a recommendation-based reputation system. Second, it leaves the LSAs a degree of freedom in the calculation of local trust and reputation values, the central expert system being restricted to setting some parameters of the local trust calculation algorithms. It therefore facilitates on-line trust and reputation calculation in a local neighborhood and enables prompt reaction to dynamic changes of the threat level.

The reputation system designed for the IIPS is oriented towards Level 2 threats and attacks and employs the modules and methods implemented in the IIPS security architecture i.e., the LAD modules (which detect attacks originated at an IIPS node and directed against one or several other nodes), and the HMAC protection modules (which enable detection of attacks from within a path between two neighboring IIPS nodes). Thus the designed reputation system can validate the trust level with respect to both IIPS nodes and inter-node links.

The assumed centralized architecture of the reputation system implies that LSAs observe the traffic received from neighbor nodes and estimate the related threats by calculating the severities of detected anomalies as explained earlier in this paper (see also [2]). The results of this process are periodically reported to the MSA, which uses them to consolidated the calculation of trust and reputation values for all IIPS nodes. The centralized structure of the reputation system makes it possible to keep track of the nodes' behavior on a global (IIPS-wide) scale, enabling detection of attacks that are impossible to be discovered locally. The downside of such a solution is that MSA is a single point of failure vulnerable to an attack, such as DoS. However, in the case of any MSA dysfunction, each LSA can continue to estimate its neighbors' trust and reputation values according to its local experience. In the meantime, the MSA can be quickly restored at a different node using e.g., the approach of [5]. Subsequently, supplied with recommendations from all LSAs the new

MSA can restart consolidation of recommendations and calculate the global trust and reputation of all IIPS nodes.

As dictated by the IIPS security architecture, each LSA estimates the trust of each its neighbor nodes using two approaches: IIPS-PDU verification via HMAC test and anomaly detection algorithms performed by the local LAD module. Combining the HMAC-based verification and certain anomaly detection algorithms one can assign trust levels to inter-node links. Specifically, one is able to determine if the reason of an anomaly detected in traffic received from a neighbor node is that node's improper behavior or the threat arises along the link from that node. Another indication could be unusual traffic behavior (e.g., a huge amount of unexpected data) observed without the local LAD reporting any neighbor node's improper behavior. On the other hand, if the source of observed disturbances is an IIPS node, the analysis of consolidated recommendations by the MSA yields a definite indication that this is the case. Typically, this occurs when negative (low-trust) recommendations are received from all neighbor nodes of the culprit node.



✓ The security agent "SecurityAgent object" was added successfully.

Select security agent to change

0 of 7 selected

□	Name	Type	Addr	Period	Alpha	Alpha1	Alpha2	Window	Support	Changed
□	node1	LAD	2001:db8:100::1	(None)	(None)	(None)	(None)	10	3	June 15, 2012, 2:27 p.m.
□	msa	MSA	2001:db8:100::100	15	(None)	0.5	0.5	(None)	(None)	June 15, 2012, 2:22 p.m.
□	node1	LSA	2001:db8:100::1	(None)	2.0	(None)	(None)	(None)	(None)	June 15, 2012, 2:24 p.m.
□	node2	LSA	2001:db8:100::2	(None)	2.0	(None)	(None)	(None)	(None)	June 15, 2012, 2:26 p.m.
□	node3	LSA	2001:db8:100::3	(None)	3.0	(None)	(None)	(None)	(None)	June 15, 2012, 2:26 p.m.
□	node2	LAD	2001:db8:100::2	(None)	(None)	(None)	(None)	10	3	June 15, 2012, 2:27 p.m.
□	node3	LAD	2001:db8:100::3	(None)	(None)	(None)	(None)	10	3	June 15, 2012, 2:28 p.m.

7 security agents

Fig. 8. IIP Security System management interface for configuration of security agents of different types (MSA, LSA, LAD)

The reputation system is equipped with a user interface in the form of a Web-based control panel with a possibility of setting all parameters for each individual node and for the MSA that manages the global reputation system (see Fig. 8). For the MSA and LSAs these parameters are the sampling interval at which LSAs report their trust estimates to the MSA, and the anomaly detection algorithms' parameters needed to calculate anomaly intensities and severities. In the current implementation, these parameters can be tuned through a graphical interface. All the parameters must be fixed according to expert knowledge and long-time observation of the system functioning, although there is no general law that dictates the calculation of the values of these parameters. Their specific values make the reputation system more or less sensitive to different types of attacks, and a suitable time constant value is set to adjust the speed of reaction to changes in received traffic characteristics. The basic Web-based interface for the IIPS security architecture is designed to be user-friendly and at the same time transparent to the networks node structure. The Web application used as the interface is based on Django framework [6]. Since it is a high-level Python language Web interface, it is compatible with the NetSNMP package which has been implemented for secure SNMP communication. The design of its main management window allows for presentation of LSA and MSA parameters along with addition and subtraction of LSA nodes according to a dynamically changing PI topology. The application allows for adjustment of each of the above parameters of the IIPS security architecture in an easy and intuitive way. After the parameters values are updated, the whole configuration is stored and then shared using a flexible relational database management system (in the current implementation, SQLite [7]). Both the MSA and LSAs update values of their parameters by downloading them from the database. Access to the security management system is protected in the usual way; in the basic solution, the username/password authentication is required, but its future extended version will be protected by using https (SSL with certificates). Communication between the database and security agents is realized by encrypted and authenticated SNMPv3 *Inform* messages which contain the parameter configuration.

6. Network management and security system management protection

As mentioned above (cf. [2]), the reputation system communicates using SNMPv3. To protect the security (in particular, reputation) system communication as well as other management communication, we have introduced another element of the IIPS security architecture, namely SNMP-PROXY. It is now integrated with the IIPS management system. Moreover, the IIPS security management interface is linked to the management system console, making it an inherent part of the IIPS. The SNMP-PROXY functionality is implemented to provide two important goals. First, it protects the IIPS management traffic from different attacks. Second, it provides the SRE logging subsystem focused on SNMP traffic. The architecture of SNMP-PROXY is shown in Fig. 9.

The IIPS management traffic protection is achieved by using SNMPv3. SNMP-PROXY waits for a connection from a network, then it performs the encryption and authentication functionalities for each connection. Upon detection of a connection, the received request is relayed to the SNMP management server, where it is processed and sent back. In this configuration the SNMP management server can use an insecure version of SNMP, but all SNMP traffic must be passed through SNMP-PROXY. The Proxy requires three security parameters: security name (username) and the encryption and authentication passwords. In our current implementation, the connection is established if and only if all these parameters are set properly. SNMP-PROXY is implemented using pySNMP [8], and it allows the use of selected cryptographic algorithms. As the message authentication tool (packages authentication), the HMAC scheme based on MD5 or SHA-1 is available, whereas secrecy is optionally provided by encryption with DES, 3DES or AES (with 128-bit, 192-bit or 256-bit keys).

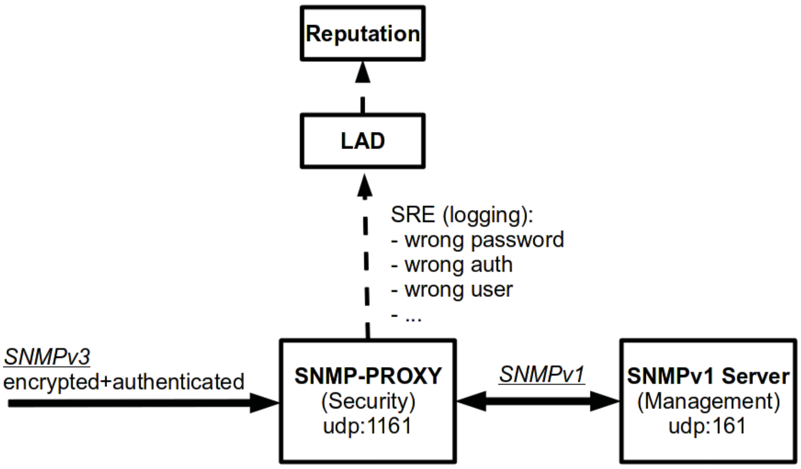


Fig. 9. SNMP-PROXY architecture

To provide SRE logging system, SNMP-PROXY gathers security-related information and passes it to the anomaly detection modules. Suspicious actions logged by proxy and considered as possible anomalies are: a wrong username, a wrong encryption/authentication password, a version of SNMP not equal to 3, and attempts at unencrypted/unauthenticated transmission.

7. Conclusion

We have presented an outline of the current implementation of the IIPS security architecture meant to address Level 2 security threats. After a brief reminder of the role of the main components their principles of operation, selected implementation details and preliminary testing have been described. At present, it is recognized that the next step will consist in the full integration of the proposed architecture into the IIPS functionality, in particular into the IIPS management system (the MGT PI), as well as testing its resilience to various Level 2 attacks in a real IIPS traffic environment. These tasks are expected to be completed later this year.

References

1. W. Burakowski, H. Tarasiuk, and A. Beben, *System IIP for supporting „Parallel Internets (Networks)”*, FIA meeting, Ghent 2010, fi-ghent. fiweek.eu/files/2010/12/1535-4-System-IIP-FIA-Ghent-ver1.pdf
2. Krzysztof Cabaj, Grzegorz Kołaczek, Jerzy Konorski, Piotr Pacyna, Zbigniew Kotulski, Łukasz Kucharzewski, Paweł Szałachowski, *Security architecture of the IIP System on resources virtualization level* (in Polish), Telecommunication Review - Telecommunication News, Vol.84(80), No.8-9, pp.846-851 (2011).
3. M. Burgess, *Two dimensional time-series for anomaly detection and regulation in adaptive systems*, Proc. IFIP/IEEE 13th Int. Workshop on Distributed Systems: Operations and Management, DSOM 2002, pp. 169-185
4. D-ITG Distributed Internet Traffic Generator, http://www.grid.unina. it/software/ITG/
5. Grzegorz Oryńczak and Zbigniew Kotulski, *Self-healing central server for hybrid P2P systems*. Annales UMCS Informatica (2012), to appear
6. https://www.djangoproject.com/
7. http://www.sqlite.org/
8. http://pysnmp.sourceforge.net/