Postprint of: Konorski J., Pacyna P., Kolaczek G., Kotulski Z., Cabaj K., Szalachowski P., A Virtualization-Level Future Internet Defense-in-Depth Architecture. In: Thampi, S.M., Zomaya, A.Y., Strufe, T., Alcaraz Calero, J.M., Thomas, T. (eds) Recent Trends in Computer Networks and Distributed Systems Security. SNDS 2012. Communications in Computer and Information Science, vol 335. Springer, Berlin, Heidelberg. DOI: 10.1007/978-3-642-34135-9_29

A Virtualization-Level Future Internet Defense-in-Depth Architecture

Jerzy Konorski[1], Piotr Pacyna[2], Grzegorz Kolaczek[3], Zbigniew Kotulski[4], Krzysztof Cabaj[4], Pawel Szalachowski[4]

[1] Gdansk University of Technology, Poland, jekon@eti.pg.gda.pl
[2] AGH University of Technology, Poland, [3] Wroclaw University of Technology, Poland
[4] Warsaw University of Technology, Poland

Abstract. An EU Future Internet Engineering project currently underway in Poland defines three Parallel Internets (PIs). The emerging IIP System (IIPS, abbreviating the project's Polish name), has a four-level architecture, with Level 2 responsible for creation of virtual resources of the PIs. This paper proposes a three-tier security architecture to address Level 2 threats of alien traffic injection and IIPS traffic manipulation or forging. It is argued that the measures to be taken differ in nature from those ensuring classical security attributes. A combination of hard- and soft-security mechanisms produces node reputation and trust metrics, which permits to eliminate or ostracize misbehaving nodes. Experiments carried out in a small-scale IIPS testbed are briefly discussed.

Keywords: Future Internet; virtualization; security architecture; HMAC; anomaly detection; reputation system.

1 Introduction

The EU Future Internet (FI) Engineering project currently underway in Poland (named IIP, which abbreviates its Polish name) focuses on the idea of a physical communication substrate shared by three Parallel Internets (PI), each running a different protocol stack over a set of virtualized links and nodes [1]. This is in line with existing FI approaches, cf. [2], [3], [4] and Fig. 1a. Two post-IP PIs are named Data Stream Switching (DSS), and Content Aware Network (CAN), and one is IPv6 QoS oriented. A testbed embodiment of this idea, the *IIP System* (IIPS), is physically based on Ethernet links over which IIPS protocol data units (IIPS-PDUs) are transmitted. In each link, virtual links are created to connect virtual nodes adjacent in a PI topology, the task of separation of the PIs' traffic and performance being left to nodal schedulers. IIPS architecture consists of four Levels (Fig. 1b), where Level 1 is the physical infrastructure and Level 2 is responsible for creation of PI virtual links and nodes.

This paper addresses two IIPS Level 2 security concerns. First, an external intruder (*outsider*) might manipulate IIPS traffic or inject alien traffic into IIPS in order to disrupt IIPS functionality. Second, a virtual machine (VM) implementing a virtual IIPS node can be compromised by an internal intruder (*insider*) and so is not a trusted entity. In particular, it can forge IIPS traffic to instigate harmful actions or states at an

IIPS node; an attack upon a single VM in a PI may also affect other PIs. To address these concerns, Level 2 security measures are proposed instead of classical perimeter protection or protocol- and application specific measures. In Section 2 we briefly comment on existing work on FI security. In Section 3 we characterize Level 2 security threats to IIPS and our defense approach. In Section 4 we outline the proposed Level 2 security architecture. We believe this novel approach transcends its project context and applies to any networking environment where multiple virtual protocol stacks are embedded in a common/public physical substrate. The envisaged cooperation of hard- and soft-security mechanisms including local anomaly detection (Section 5) and a reputation system (Section 6) permits to eliminate or ostracize distrusted IIPS nodes. We present these mechanisms with a view of their implementation. Experiments in a small-scale IIPS testbed are discussed in Section 7.

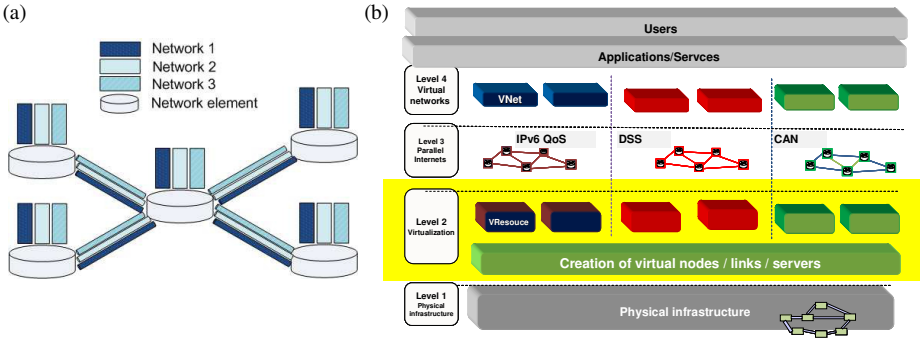


Fig. 1. Virtualization-based FI; a) virtual network infrastructure [3], b) IIPS architecture [1].

2 Current Work on FI Security

In many FI projects, trust and security appear jointly as an important building block. A common perception is the need for addressing trust and security concerns from a project's initial stages [5]. The FI X-ETP Group [6] lists security build-up at design time as a key challenge and presents a concept of a trust architecture. Emerging threats in the FI urge work on FI security before they materialize [7]. The 4WARD project [8] presents a concept of an information-centric architecture with security-aware object identifiers. In the follow-up SAIL project [9], content- rather than channel-oriented security services are developed as part of the NetInf architecture. Effectsplus, an FP7 funded Coordination & Support Action [10], analyses current trust and security work to identify key areas and players. References to trust and security-related pages with a work-in-progress are offered e.g., by EFII [11] and FIA [12], cf. also [13].

Network and resource virtualization is present in several FI projects ([8], [14], [15]). It is also the leading motive of IIPS. In a promising approach of the NetSE project [16], the contemporary Internet migrates towards the FI through the deployment of dedicated software modules called Cognitive Managers. Each of them is responsible for specific virtual resource abstractions and has an in-built Supervisor and Security Module that among others ensures selected security attributes.

3 Level 2 Security Threats and Defense Approach

A *threat* is a possibility of damage arising from a specific IIPS vulnerability, and an *attack* refers to an intruder's activity which exploits this vulnerability. Here we only address IIPS Level 2 security threats and attacks i.e., related to IIPS traffic over virtual links in a PI topology. *External* threats relate to generation of fake IIPS traffic or illegal modification of IIPS traffic outside IIPS. IIPS-PDUs are multiplexed over a common Ethernet infrastructure along with alien traffic, where outsider attacks via VLAN hopping, IIPS-PDU capture or corruption are relatively easy to launch. Their impact depends on the outsiders' capabilities, such as injecting alien PDUs, sensing, buffering and/or modification of IIPS-PDUs; however, with adequate PI perimeter protection, fake or modified IIPS traffic can often be recognized as such. *Internal* threats are posed by compromised VMs. An insider controlling the VM can spoof a virtual node, forge or modify IIPS-PDUs, and append correct security tags to get the traffic past perimeter protection. This may lead to more serious damage than an outsider can inflict, and not necessarily confined to a single PI. Straightforward attacks are traffic *injection*, *replay/resequencing*, *ruffling* (disruption of IIPS-PDU spacing via IIPS-PDU capture and hold-up) and *forging* (generation of fake though IIPS-formatted traffic). While the first three mainly induce "quantitative" harm at an IIPS node (e.g., extra processing effort or a perception of poor inter-PI performance isolation), traffic forging has a "qualitative" effects––it may disrupt the core functionality of, or create any undesirable state at an IIPS node.

Contemporary security measures are often *model-based*––they rely on a repository of misuse signatures corresponding to specific vulnerabilities and attacks. In IIPS, these vulnerabilities and defenses are higher-level protocol dependent, thus cannot be addressed by the proposed architecture. On the other hand, symptoms of Level 2 attacks are less specific and so harder to capture without an awareness of higher-level protocol semantics. Within a *policy-based* approach, which we take here, no attempt is made to predict possible attack vectors; instead, anomalous traffic or node behavior is defined and watched for (we especially relate this to IIPS-PDU contents, timing or sequence, as well as IIPS node state). The proposed security measures prevent an outsider from traffic injection or IIPS traffic modification, and reliably detect traffic replay/resequencing, ruffling and forging. Thus they differ substantially from classical measures ensuring data authentication, confidentiality, and non-repudiation.

4 Defense Tiers

The proposed defense-in-depth architecture features three tiers (Fig. 2). It is primarily meant as an integration platform for various state-of-the-art security mechanisms within each tier, enabling easy replacement by more effective ones when they arise.

1st tier. To block entry of injected, replayed or resequenced traffic, integrity and authentication are assured over a virtual link by appending a hash-based message authentication code (HMAC) [17] to all IIPS-PDUs. Each pair of neighboring virtual nodes share an HMAC key and a IIPS-PDU counter. Both the IIPS-PDU contents (including relevant IIPS headers) and its sequence number are protected, thus any

received physical (Level 1) frame can be verified as alien traffic or an in-/out-of-sequence IIPS-PDU. In the former case the frame is dropped and its relevant fields are passed to the 2nd and 3rd tier for further inspection. HMAC constitutes a uniform 1st-tier security measure for the whole IIPS irrespective of the IIPS-PDU format or PI affiliation. Unlike e.g., IPSec or TLS, it is not tied to any protocol stack. To fully utilize the virtual link, both HMAC and IIPS-PDU drop modules are implemented in a four-port netFPGA board [18], with HMAC-SHA-512 message digest employed.

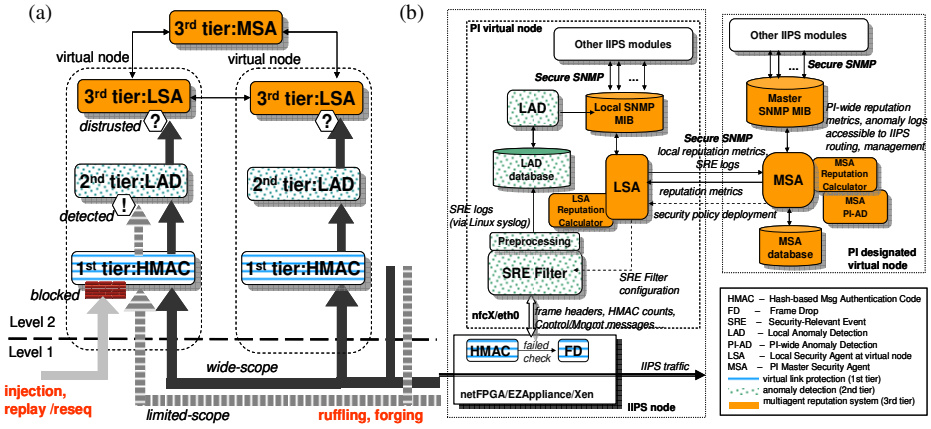


Fig. 2. IIPS Level 2 security architecture; (a) defense tiers (thick arrows visualize attack origin and impact), (b) placement of security modules.

2nd tier. Ruffling or forging attacks cannot be stopped by HMAC, yet anomalous behavior they cause can be detected as *security-relevant events* (SREs) defined by an SRE Filter e.g., HMAC ordered IIPS-PDU drops, illegal control or management messages, suspicious traffic statistics, abnormal resource usage etc. SREs are stored in a local anomaly detection (LAD) database and subjected to analysis by a LAD module implemented within a virtual node's VM code. Two complementary algorithms are used: times series analysis [19] to detect suspicious traffic and resource usage, and data mining via frequent sets [20] to detect specific patterns in suspicious IIPS-PDUs. Anomalies indicative of attacks are reported to the 3rd tier.

3rd tier. At a compromised virtual node, LAD cannot be trusted for proper detection and honest reporting of local anomalies. Moreover, certain wide-scope attacks would be missed if local-scope SRE and anomaly logs were only analyzed. This calls for inter-node cooperation. A Local Security Agent (LSA) at an IIPS virtual node translates the detected anomalies into local reputation metrics to derive the current level of trust that node deserves. LSA reports these metrics, along with the SRE logs, to the PI's Master Security Agent (MSA) via a PI-wide multi-agent reputation system using SNMPv3, a cryptographically protected version of the Simple Network Management Protocol. MSA uses a data fusion algorithm to calculate PI-wide reputation and trust metrics of the virtual nodes and its PI-wide anomaly detection (PI-AD) module captures anomalies of a larger scope. Results are made accessible to other IIPS modules, such as routing or management, via an SNMP database. They are also fed back to the nodal LSAs, which can then suitably redefine SRE Filters.

5 Local Anomaly Detection

Suspicious traffic able to penetrate the 1st tier can be detected by LAD through checking the semantics of received traffic against the security policy, or through observation of temporal traffic and virtual node behavior. The former uses data mining methods and is suitable against forging attacks targeting specific management and higher-level functionalities; the latter uses time series analysis and protects against ruffling attacks and all-purpose forging attacks such as malicious redirecting of IIPS traffic between the PIs.

Data Mining. Target specific forging attacks e.g., scanning, DoS or malware/spam outbreaks, produce repetitive patterns in observed sequences of SREs. The allowed sources of SREs are local firewall or HMAC ordered IIPS-PDU drop data and errors reported by the local SNMP agent e.g., unauthorized resource access attempts.

An SRE is represented as a set of relevant SRE features e.g., offending IPv6 address, used protocol or port identifiers. An SRE related to a reported SNMP error can have the form (sourceIPv6 = 2001:db8:201::3, user name = management, SNMP action = denied, OID = 1.3, …), where OID is the SNMP database object identifier. SRE logs are analyzed in successive *time windows* in search of *frequent sets* [20] i.e., subsets of features found abnormally frequent, as dictated by the *minimal support* parameter. E.g., a time window of 10 s and the minimal support of 4 mean that any pattern of features repeated at least four times over 10 s raises an anomaly alarm. If a discovered frequent set contains, say, a source address then a report sent to LSA via a secure SNMP message indicates the culprit node. E.g., the *nmap scanning* attack, used to discover services running on a victim machine, can manifest itself as a frequent set {type = security policy violation, sourceIPv6 = 2001:db8:201::3, destination IPv6 = 2001:db8:201::3, packet length = 1080, used protocol = TCP}. The lack of features related to source and destination ports indicates that in the current time window they have taken diverse values, as expected in a scanning attack. An anomaly has two attributes on a scale from 0 to 1: *severity*, a measure of the anomaly's adverse impact, and *probability*, a measure of conviction that the anomaly indeed indicates a security threat. (Note that the term "probability" is used here in its axiomatic sense.) For preliminary experiments described in Section 7, arbitrary severity and probability values reflect typical threat occurrences e.g., 0.05 and 0.5 for low-impact anomalies, 0.3 and 0.7 for probable software configuration errors, and 1 and 1 for (D)DoS attacks. Future research is expected to fine-tune such assignments.

Time Series Analysis. LAD also checks for anomalies in an IIPS node's behavior regarding memory and CPU usage, per-PI received and transmitted traffic volume etc. Abnormally high CPU usage or received traffic volume are typical of DoS (in particular, traffic injection) or all-purpose forging attacks, whereas traffic ruffling creates abnormal statistics of traffic bursts. First, relevant behavioral features are selected. Next, historical (training) selected feature values are compared with the current ones to learn how indicative the latter are of possible anomalies. Upon iterating the above steps, anomalies are indicated by discrepancies between the statistics of the current feature values and those derived from training data.

The anomaly detection algorithm takes as input the time series of feature values in successive time windows, denoted $x_1,\ldots, x_l,,\ldots$ Following [19], the severity of an anomaly accompanying the observation of x_l is calculated as

$$c_l = \min\{1, \sqrt{[|x_l - x_{P,l}|/(3\sigma_{P,l})]^2 + [|x_l - x_{T,P,l}|/(3\sigma_{T,P,l})]^2}\}, \quad (1)$$

where $x_{P,l}$ and $\sigma_{P,l}$ ($x_{T,P,l}$ and $\sigma_{T,P,l}$) are, respectively, the current exponential moving average and standard deviation of the time subseries consisting of arithmetic averages $\sum_{k=0}^{P-1} x_{l-k}/P$ ($\sum_{k=0}^{T-1} x_{l-kP}/T$), P and T being predefined integers. Note that for a current feature value close to both averages the severity is close to 0, whereas deviations treble the corresponding standard deviations indicate a maximum severity. The anomaly probability is the proportion of observations yielding distinctly positive severities. (Note that this time the term "probability" is used in its empirical sense.)

As an illustration, artificially generated received traffic of 5000-byte IIPS-PDUs, with Pareto distributed of interarrival times with mean 15 ms and standard deviation 75 ms, is regarded as typical and produces zero-severity observations. After 20 LAD time windows, an extra stream of 500-byte IIPS-PDUs with normally distributed interarrival times with mean 20 ms and standard deviation 5 ms is superimposed. As a result, the severity and probability values increase (Fig. 3). If the change of the traffic pattern is permanent, LAD eventually learns it and returns to zero severities, as is the case in Fig. 3. Were the extra traffic to vanish after another 30 LAD time windows, modeling a short-term ruffling attack, the period of nonzero severity and probability values would roughly double in length.

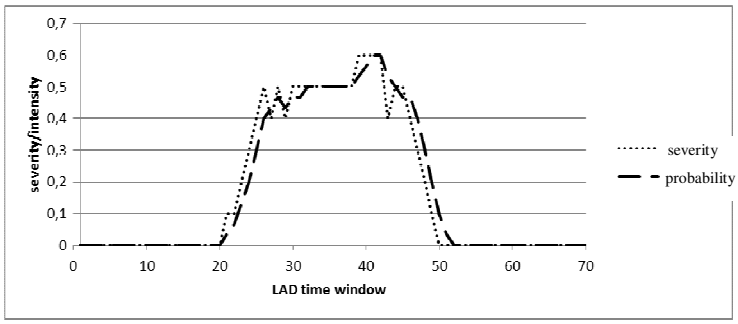


Fig. 3. Severity and probability values during a change in the traffic pattern.

6 Proposed Reputation System

The proposed PI-wide multi-agent reputation system has LSAs communicate with MSA in successive *reporting intervals*. Local trust metrics derived by LSAs are reported to MSA, which converts them into global (PI-wide) trust and reputation metrics. These are accessed by other IIPS modules e.g., routing or management, and

used to identify and/or ostracize ill-reputed nodes. They are also fed back to LSAs to modify local SRE Filters i.e., update the security policy, and to enable LSAs to act as backup MSA in case the original MSA itself is disrupted by an attack.

In the n^{th} reporting interval, LSA at node i records the severity and probability values, $c_l \in [0, 1]$ and $p_l \in [0, 1]$, of anomalies detected by LAD as caused by node j. The involved *risk* values are taken to be $r_l = c_l p_l$. Then local trust placed in node j is calculated as $T_n^{i,j} = \alpha^h (1 - r_{\max})$, where $r_{\max} = \max_l c_l p_l$, $h = \left| \{ l \mid r_{\max} / 2 \le r_l < r_{\max} \} \right|$ is the number of other high-risk anomalies, and $\alpha \in [0, 1]$ is selected experimentally. Thus threats arising from both a single maximum-risk attack and repeated high-risk ones are accounted for. Having collected the $T_n^{i,j}$ from LSAs, MSA calculates the global trust placed in node j as a combination of local trust metrics from other nodes, weighted by their respective reputation metrics

$$T_n^j = \sum_i T_n^{i,j} R_n^i \Big/ \sum_i R_n^i , \tag{2}$$

where $R_n^i \in [0, 1]$ is the current reputation metric of virtual node i (initially set to 1). MSA then calculates new reputation metrics for the $(n + 1)^{\text{th}}$ reporting interval:

$$R_{n+1}^i = \begin{cases} T_n^i, \text{ if } T_n^i \le R_n^i, \\ (1 - \beta) R_n^i + \beta T_n^i, \text{otherwise}, \end{cases} \tag{3}$$

where $\beta \in [0, 1]$. Note the conservative approach––reputation decreases immediately as dictated by diminished trust, but increases somewhat more reluctantly.

7 Preliminary Test Results

Since HMAC protects against injected traffic, we present sample proof-of-concept tests of the 2nd- and 3rd-tier security modules i.e., foiling forging and ruffling attacks. A small-scale PI testbed is shown in Fig. 4. Three IIPS virtual nodes named Node1, Node2 and Node3 host LSAs; a fourth one hosts MSA. They are controlled by a Xen virtualization engine [21] and communicate over IPv6 using SNMPv3.

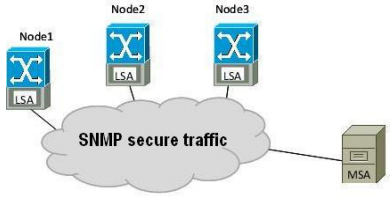


Fig. 4. IIPS testbed emulating a small-scale PI.

In the experiments, the focus was on the cooperation of the security mechanisms rather than on the feasibility of specific attacks. Therefore, SREs were only derived from `ip6table` firewall logs, which were checked for symptoms of forging attacks,

and from HMAC data and received traffic volume, which were checked for symptoms of injection and ruffling attacks. In the former case, severity and probability values were adjusted to produce adequate response to the attacks, whereas in the latter, they were calculated based according to Section 5. The reporting interval was 15 s.

In the first scenario (Fig. 5), Node2 starts attacking Node3 at time T_0. LAD at Node3 classifies this as an attack with probability 0.8 and severity 0.9. LSA at Node3 then reports to MSA the trust metric of Node2. As a result, Node2's global trust metric decreases instantly to 55% of the maximum, and remains so until time T_1, when Node2 also attacks Node1. This is detected by LAD at Node1 and causes Node2's global trust metric to drop to about 10%. Since Node2 keeps attacking, its trust metric remains low. Meanwhile, Node1 starts a short-term attack. At time T_2, the now distrusted Node2 reports to MSA an attack from Node1, but this report has minimal influence upon Node1's trust metric. However, when both Node2 and Node3 report an attack from Node1 at time T_3, Node1's trust metric decreases rapidly. This is because Node3's trust (hence, also reputation) metric remains at the maximum and MSA now weights reports about Node1's attacks much higher. When Node1's short-term attack is over, its trust and reputation metrics start increasing.

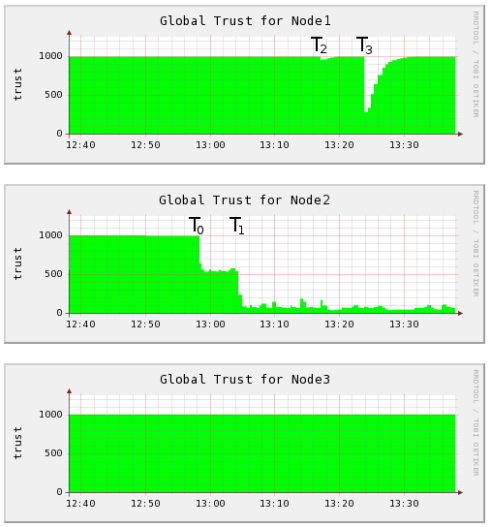


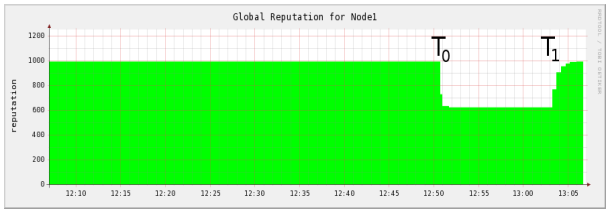
Fig. 5. Global trust for Node1..3 vs. time under Node1 and Node 2's attacks.

In Fig. 6a, between times T_0 and T_1, Node1 floods Node3 with IIPS-PDUs performing an `nmap` scanning attack (with symptoms similar to a DoS attack). LAD at Node3 correctly classifies this as an attack with severity 1 and probability 1. MSA receives repeated reports from Node3, hence Node1's reputation metric decreases instantly to about 62%.

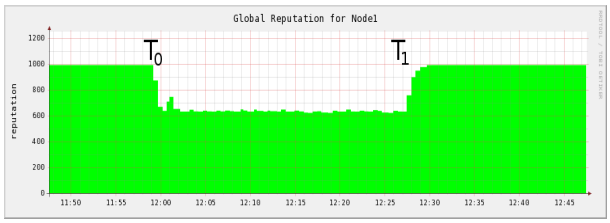
In Fig. 6b, Node1 performs a *sophisticated* scanning attack: after each connection termination it pauses for 1000 ms by executing `nmap -6 --scan-delay 1000 Node3`. LAD at Node3 reports the attack severity 0.1 and probability 0.05. MSA again receives repeated reports, so Node1's reputation and trust metrics decrease, throughout the attack averaging 64% and ranging from 62% to (momentarily) 92%.

In Fig. 6c, Node1 tries a pause duration of 1500 ms (a *moderate* scanning attack). LAD at Node3 detects an attack of severity 0.1 and probability 0.5. The global trust metrics throughout the attack average 71% and range from 65% to 93%, behaving steadily most of the time. (A *lazy* scanning attack with pause durations of 3000 ms does not change the chart visibly; in this case, Node1's global trust metric average rises to 85%, reflecting the limited impact of the attack.)

(a)



(b)



(c)

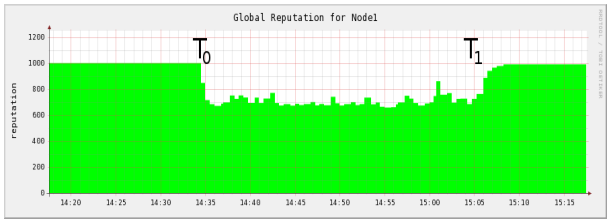


Fig. 6. Global trust for Node1 vs. time under scanning; (a) ordinary, (b) sophisticated, (c) moderate.

8 Conclusion

We have proposed a low-level security architecture for a Future Internet system called IIPS, where several Parallel Internets share a common physical transmission infrastructure via link and node virtualization. Security threats have been pointed out that arise from physical link sharing by IIPS and non-IIPS users, as well as from traffic possibly originated at compromised virtual nodes. This calls for security measures different in nature from those addressed by classical security attributes such as data confidentiality, authentication or non-repudiation. A case has been made for a three-tier security architecture featuring HMAC, anomaly detection, and virtual node reputation and trust evaluation mechanisms. Sample test results, obtained from a

small-scale Parallel Internet with a repertoire of nodal misbehavior, have been discussed and demonstrated to create adequate response to Level 2 security threats. Future work will focus on fine-tuning of the key LAD and MSA parameters and testing the proposed architecture against a broader scope of low-level attacks.

Acknowledgments. Work supported in part by the European Regional Development Fund Grant POIG.01.01.02-00-045/09-00 Future Internet Engineering. The work of J. Konorski is supported in part by the Air Force Office of Scientific Research, Air Force Material Command, USAF, under Grant FA 8655-11-1-3076.

References

1. Burakowski W., Tarasiuk H., Beben A.: 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. Anderson T., Peterson L., S. Shenker J. Turner J.: Overcoming the Internet impasse through virtualization. IEEE Computer, 38(4):34–41, 2005
3. Fernandes N.C.: Virtual networks: isolation, performance, and trends. Annales des Telecomm., 66, 5-6, June 2011, pp. 339-355
4. Campanella M., Maglaris V., Potts M.: Virtual Infrastructures in Future Internet. In: Tselentis G. et al. (Eds.), Towards the Future Internet, IOS Press, 2010
5. Gavras A. *et al*.: Future Internet Research and Experimentation: The FIRE Initiative. ACM SIGCOMM Computer Communication Review, 37, 3, July 2007
6. Future Internet-Strategic Research Agenda, ver. 1.1, Future Internet X-ETP Group, 2010.
7. http://www.syssec-project.eu/
8. www.4ward-project.eu
9. http://www.sail-project.eu/wp-content/uploads/2011/08/SAIL_DB1_v1_0_final-Public.pdf
10. http://www.effectsplus.eu
11. initiative.future-internet.eu
12. European Future Internet Portal, www. future-internet.eu
13. http://fipedia.org/fipedia/index.php?title=Category:Security
14. Flizikowski A., Majewski M., Hołubowicz M., Kowalczyk Z., Romano S.P.: The INTERSECTION Framework: Applied Security for Heterogeneous Networks. J. of Telecomm. and Information Technology, 1/2011
15. New Generation Network Architecture: AKARI Conceptual Design, http://akari-project.nict.go.jp/eng/index2.htm
16. Castrucci M., Delli Priscoli F., Pietrabissa A., Suraci V.: A Cognitive Future Internet Architecture. In: Domingue J. *et al*. (Eds.): Future Internet Assembly. LNCS 6656, pp. 91-102, 2011, Springer, Heidelberg
17. Kelly S., Frankel S.: Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec. Proposed standard (with errata), Internet Engineering Task Force, May 2007.
18. http://www.netfpga.org/
19. Burgess M.: Two dimensional time-series for anomaly detection and regulation in adaptive systems. Proc. IFIP/IEEE 13th Int. DSOM Workshop, 2002, pp. 169-185
20. Agrawal R., Srikant R.: Fast algorithm for mining association rules. In: Bocca J.B., Jarke M., Zaniolo C. (Eds.), Proc. 20th Int. Conf. on Very Large Databases, pp. 487-499, (1994)
21. Egi N. *et al*.: Evaluating Xen for router virtualization. Proc. Int. Conf. on Computer Communications and Networks ICCCN 2007, August 2007, pp. 1256–1261