THE PUBLISHING HOUSE
OF THE ROMANIAN ACADEMY

PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A,
Volume 13, Number 2/2012, pp. 149–156

E-SERVICE SECURITY[*]

Valentin CRISTEA, Catalin LEORDEANU, Florin POP, Ciprian DOBRE

"Politehnica" University of Bucharest, Faculty of Automatic Control and Computers
Splaiul Independentei 313, Sector 6, Bucharest 060042, Romania
E-mail: florin.pop@cs.pub.ro

Service Oriented Distributed Systems have important advantages in computing by enabling resource sharing and communication among services in environments that support large scale applications. E-Services are exposed to various kinds of threats in the Internet. Ensuring intrusion detection in such systems is not an easy task due to their heterogeneity, dynamic nature, and large scale. We present a novel Intrusion Detection System (IDS) with two layers. The low layer includes local IDSs to detect attacks to individual resources / services while the high layer is a global IDS that detects coordinated attacks to collections of resources / services of a Distributed System. For the local IDS we propose a hybrid solution that combines two intrusion detection methods (pattern matching and anomaly detection) to increase detection's precision. The global IDS efficiently detects application-based and resource-based attacks that cannot be detected at the local level.

Key words: security, e-Services, intrusion detection, distributed system.

1. INTRODUCTION TO E-SERVICES AND SECURITY ISSUES

The actual knowledge-based society is governed by the necessity to offer various functionalities in the form of e-Services, mainly hosted by distributed systems, platforms, and environments. E-Services have several advantages over previous paradigms used in building large scale Distributed Systems (DSs) such as files, documents, objects, etc. E-Services are easy to use, can dynamically discover each other by using the publish-subscribe-notify mechanisms, and can communicate in timely and consistent information. They are at the aim of many Web-based systems (e-business, e-government, e-health, etc.), of Grid systems, and of the actual Cloud systems that expose the known paradigms Infrastructure as a Service, Platform as a Service, and Software as a Service.

E-Services are vulnerable to various kinds of threats from the Internet, and are asked to offer confidentiality, integrity, and availability to users and other e-Services. Distributed systems are more vulnerable than other kinds of systems due to the decentralized control, distribution on large geographic areas, remote user access, use in more critical applications (such as Internet banking), extension over multiple administrative domains with different security policies, use of local and global operations, and sharing by user groups. Solutions for problems such as global identity management, cross boarding identification, privacy, prevention of attacks are essential for the wide public acceptance of e-Services.

In this paper we present new solutions for a large spectrum of attacks, from simple network attacks to high-level coordinated attacks against Large Scale Distributed Systems. At the network level the methods presented here are intended to stop attacks such as Denial of Service and detect local policy violations. We propose a novel IDS which uses a hybrid approach based on a pattern matching engine and a neural network functioning in parallel to improve the detection efficiency. Moreover, we use the data collected from these low-level IDSs to protect an entire Grid environment. At this high level we have access to a global view of the Grid and we can also detect complex attacks focused on certain running applications or on groups of resources. This approach correlates the information received from the network level and monitoring data from the Grid System and identifies attacks that cannot be detected at a local level.

[*] This paper was presented in part at the 1st Conference *Romanian Cryptology Days* – RCD2011, October 11–12 2011, Bucharest, Romania.

The rest of the paper is organized as follows. Section 2 presents related works and their limitations. Section 3 describes separately the salient features of the two-level intrusion detection solution. For each level, the experimental results are presented that demonstrate the high of the proposed solutions in terms of detection precision and efficiency.

This material was in part presented at the international conference Romanian Cryptology Days, RCD-2011.

2. RELATED WORK

Network IDSs are software and / or hardware tools that monitor the events which may appear in a computer network and try to identify potential security breaches. IDSs are present in almost all environments. They are very useful in detecting Denial of Service (DoS) or Distributed Denial of Service (DDoS) attacks, which try to damage computing systems by decreasing their performance and possibly by impairing real user requests. In network intrusion detection we distinguish two major approaches: pattern matching and anomaly detection.

Many IDSs rely on *pattern matching*. In this method, IDSs try to identify a specific pattern which is known to be malicious. The advantage is that IDSs are efficient, and easy to implement and install. They have a low rate of false positives. One model used to detect the series of actions which are part of an attack pattern is the finite state machine. While the matching process is taking place the IDS advances from state to state until an attack pattern is identified. The model is efficient and can represent very complex attacks. Unfortunately, pattern matching cannot detect new attacks for which a pattern was not defined. Moreover, since two attacks are rarely identical, it may be difficult to detect variations of the same type of attack if the patterns have not been carefully designed.

The most popular network IDS is Snort [9]. It has capabilities to detect every attack known to system administrators so far. A lot of options are available to configure Snort as a Network IDS. We can raise or lower the alert available conditions; we can make it log a lot of data, or just the significant parts. The application can be used even on Gigabit Ethernet to detect possibly malicious traffic. Another application for monitoring the network traffic is Nagios [10]. It is not strictly oriented on intrusion detection, but a system administrator might find it useful for the large number of things it can monitor. But the intrusion decisions are mostly based on the administrator ability to spot wrong behaviours.

To detect anomalies we can use statistic methods [14, 15]. This approach defines the expected normal behaviour and then applies statistic tests on the monitored traffic to determine whether it is normal or abnormal. This approach helps detecting previously unknown attacks and also collects information about the network traffic and generates a behaviour model for further use. It is very accurate but usually such algorithms are very complex and difficult to implement and adapt.

Neural networks are also used for anomaly detection [14]. They learn how to predict the next action based on the previous ones. To use this method for an IDS, someone needs to train the neural network on large amounts of traffic. An anomaly is perceived as a discrepancy between the normal behaviour and the monitored one. However, method's efficiency depends heavily on the quality of the initial training data. Anomalies detection methods are good choices for IDSs which are continually confronted with unknown events that represent possible threats.

While intrusion detection for services in Large Scale Distributed Systems is of great concern [8], there are also considerable challenges which must be faced. First of all a good IDS must be scalable and must offer an acceptable performance level. Also, due to the heterogeneous nature of the Grid and the fact that the resources have a high geographic distribution and are part of different administrative domains, it becomes extremely difficult to monitor and detect the potential attackers by using a single intrusion detection entity. We mention here two works on distributed IDSs.

The Distributed IDS presented in [9] combines the abilities of a simple system monitor with intrusion detection monitoring of individual hosts. It provides basic intrusion detection capabilities for a general distributed system. The IDS uses a centralized analysis engine (DIDS director) and agents for monitoring systems and network traffic. The agents scan logs of events of interest and report the events to the DIDS director, which invokes a rule-based expert system for data analysis. The results are passed to the user interface, which displays them in a simple, easy-to-grasp manner for the system security officer.

Another approach to intrusion detection is described in [8]. It offers fault tolerance by using autonomous agents. Each agent can have its own model of behavior. When it detects a deviation from the expected behavior, a match with a particular rule, or a violation of a specification, it notifies other agents. The agents would jointly determine whether the set of notifications are sufficient for a reportable intrusion. The main advantage of this solution is the absence of a single point of failure. If one agent is compromised, the others could continue to function. Furthermore, when an attacker breaks one agent it learns nothing about the other agents or about the network. Our solution also has these advantages. In addition it provides scalability and awareness of different administrative domains and of the running applications, which are useful in large scale distributed systems.

Intrusion detection in multi-agent systems was considered in [16] where is presented the design of such system for a distributed environment dedicated to developing and monitoring agents. The idea is to endow the agent platform with a high level of immunity by integrating an intrusion detection system based on bio-inspired techniques (genetic algorithm for intrusion detection which simulates the natural model of recombination for evolving new generations of chromosomes which represent sets of actions to be applied in order to increase system's performances). The agent platform itself is improved with advanced mechanisms for monitoring, analyzing, discovering, learning, but its most important role is to detect and reject intruders. In this approach is introduced a monitoring process of agents activities based on the physical resources usage which is the key element in detecting possible intruders. In an agent platform, as in any other distributed system, the need of monitoring and limiting the access to certain physical resources proves to be critical in order to guarantee a proper level of performance for developing different activities.

3. SOLUTION FOR E-SERVICE SECURITY

We propose an IDS with a two level configuration: the lower level or local IDSs that monitor the resources used by the currently running application must be able to send alerts to the Grid level IDS (G-IDS) when an intrusion is detected. The local IDS can employ a number of advanced methods and algorithms in order to efficiently detect a wide range of possible intrusions. On the higher level, G-IDS provides mechanisms to correlate separate alerts from different local IDS in the case of possibly related intrusions.

3.1. Local IDS

Our solution uses Netpy [1], a network traffic analysis and visualization package. The result is a complex IDS able to efficiently detect known attacks and signal unknown threats which do not follow a pre-defined pattern [5].

The novel IDS architecture takes advantage of the Netpy network monitoring functionality and incorporates two intrusion detection mechanisms [4], namely the pattern matching and neural networks. The resulting architecture for the hybrid Netpy IDS is presented in Fig. 1.

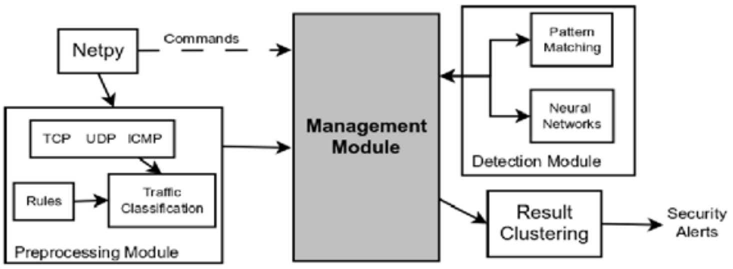


Fig. 1 – Netpy intrusion detection architecture.

The Preprocessing Module filters the traffic meta-data to decrease the volume of data to be analyzed, and supports the configuration of data source locations, logging information, various parameters of the monitored network, etc. It collects the normal traffic meta-data which is used to train the Neural Network detector.

The Detection Module has two components. The first one implements a simple pattern matching algorithm that accepts attack patterns (in XML format) and continuously tries to match them to the incoming network traffic. The attack patterns can be loaded dynamically while the matching engine is running, which adds flexibility to the module. The second component is an anomaly detection engine that uses a "back propagation" neural network for training. The two mechanisms analyze the traffic and independently send detected alerts back to the Management Module (even though they send similar alerts).

Since our architecture incorporates two independent intrusion detection solutions, we define a uniform and comprehensible alert format and introduce a Result Clustering module for analyzing the alerts. The solution helps reducing the number of false positives and provides the network administrator a clearer picture of the monitored network. Since pattern matching has a lower false positive rate, the security alerts it generates have a higher priority than the ones generated by the anomaly detection engine.

3.1.1. The Hybrid Intrusion Detection Algorithm

By implementing the two Intrusion Detection mechanisms we increased the overall efficiency of the local IDSs and brought several other advantages. The system can detect known attacks with a high degree of accuracy and can notify any unknown events which may represent possible attacks.

The Pattern Matching component of the Intrusion Detection Module can identify PortScan attacks, Denial of Service (DoS) and many others through the use of carefully designed attack patterns which are then compared to the NetFlow records received from Netpy. DoS attacks force the targeted computers to reset or consume their network resources so they can no longer provide the intended services. A pattern specification for a possible DoS intrusion attack is shown in Fig. 2, considering a 20-second window of NetFlow data.

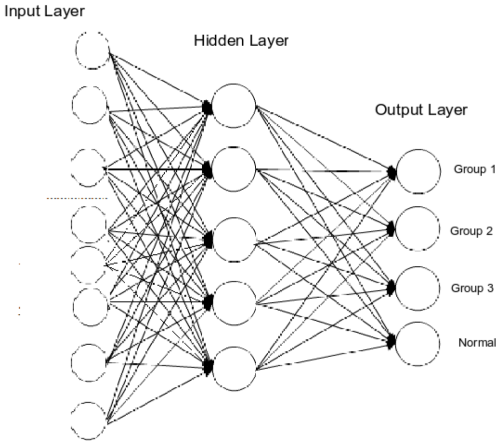
```
<patterns>
        <rule scope="20s">
                <timeconstraints>
                        <begin>11:00:00</begin>
                        <end>11:30:00</end>
                        <repeat>true</repeat>
                </timeconstraints>
                <precondition>
                        <alert type="A">
                                <sourceIP>0</sourceIP>
                                <destIP>127.0.0.1</destIP>
                                <sourcePort>0</sourcePort>
                                <destPort>80</destPort>
                                <protocol>TCP</protocol>
                                <connections>400</connections>
                        </alert>
                </precondition>
                <action type="sendAlert">
                        <alertLevel>1</alertLevel>
                        <actDestIP>192.168.1.1</actDestIP>
                </action>
        </rule>
</patterns>
```

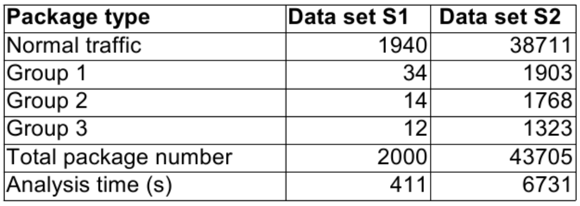
Fig. 2 – Pattern specification of a DoS attack.

This module allows the administrator to define his own patterns for malicious behavior, according to local policies. The anomaly detection component uses a "back-propagation" neural network that has a single hidden layer (see Fig. 3) and an output layer with 4 units: 3 of them correspond to Denial of Service attacks, PortScan attacks, and user-defined attacks, while the fourth is for normal traffic. Each attack type has an associated XML pattern. The neural network is built and exploited in 3 phases: collection the training data, neural network training, and the detection of anomalies. The collection of training data is achieved by the Preprocessing Module.

The output of the neural network component is assigned to one of the four groups according to the following rules:

1. If the output of the neural network belongs to one of the groups with accuracy larger than 50\% then it is placed in that group.
2. If the output of the neural network belongs to several groups it will be placed in the one with the highest accuracy.





Package type	Data set S1	Data set S2
Normal traffic	1940	38711
Group 1	34	1903
Group 2	14	1768
Group 3	12	1323
Total package number	2000	43705
Analysis time (s)	411	6731

Fig. 3 – Neural Network of the Detection Module.

Fig. 4 – Description of the experimental data sets.

This approach can detect unknown attacks if they can be classified in one or more of the first three types previously mentioned.

3.1.2. Experimental results

For validation, the Intrusion Detection module was tested with large sets of data. We proved that the module is efficient and capable of detecting a large number of attacks. The full set of experiments proving these claims can be found in [4]. Here we show that an IDS using two detection methods detects more intrusions than those based on one method (Figure 3). We used two data sets for the experiments (see Fig. 4). The first one (S1) contained synthetic data with a total of 2000 packets. The second data set (S2) was much larger and contained random traffic. Each of the data sets presents an equal distribution of malicious events.

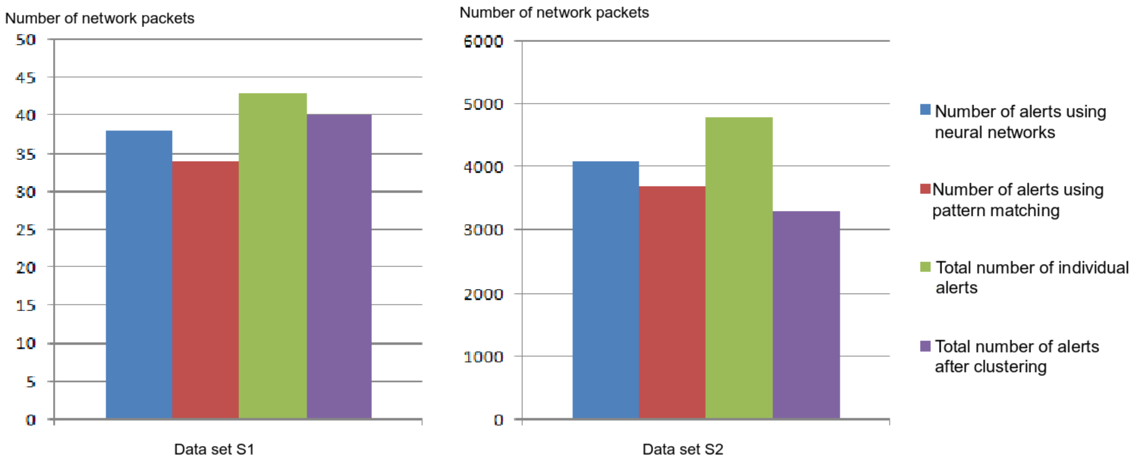


Fig. 5 – Number of security alerts using the Netpy IDS.

The false positive rate for the two data sets was below 0,03% which proves the efficiency of the detection process (see Fig. 5). Also, we achieved a detection rate of 99,3% for both data sets. With these results we proved that the hybrid approach for IDS offers good performance and accuracy. This approach has a higher performance than a simple pattern matching algorithm or a neural network.

3.2. Global IDS

Global threats are coordinated attacks on different resources possible situated in different administrative domains. Grid-based threats can only be detected by an entity with a global view on the entire

Grid System. In the layered approach of our Grid Intrusion Detection [2, 3] solution, local threats are detected at the lower levels by IDSs and the information is passed to the G-IDS, which detects Grid-based threats. The two-layer organization is depicted in Figure 6.

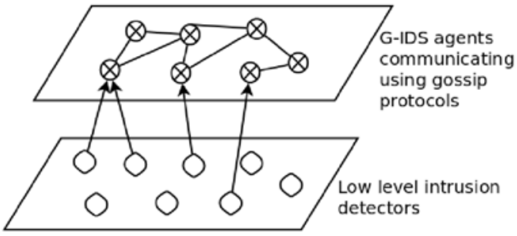

G-IDS agents communicating using gossip protocols
Low level intrusion detectors

Fig. 6 – Two layer architecture of G-IDS.

We can distinguish between two different types of global attacks according to the intent of attackers. In *Application based* attacks an intruder is interested in altering the execution of a complex application which is running on a large number of geographically distributed resources. The Resource based attacks focus on a group of resources that belong to a specific VO or share the same administrative domain. In these cases, the G-IDS must know the running applications and should correctly correlate alerts received from lower level IDSs. To do this the G-IDS also receives real-time information from Grid monitoring systems, which offers a global view of the context in which attacks take place.

The distributed nature of G-IDS offers a view of the entire Grid. It is designed as a network of inter-cooperating agents. The architecture of a G-IDS agent includes four modules (Fig. 7).

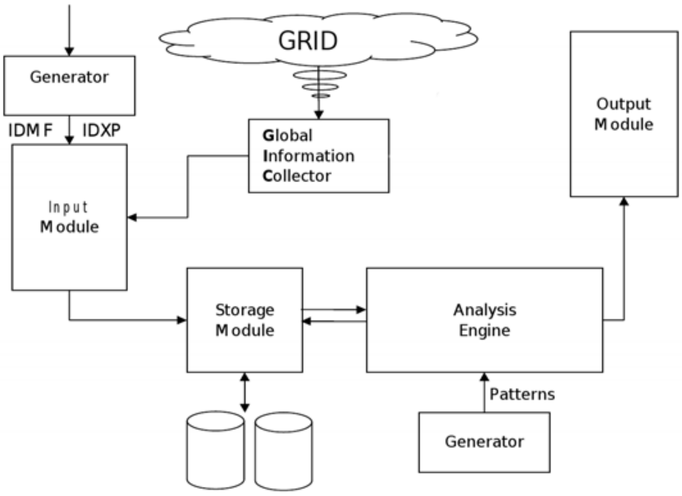


Fig. 7 –Architecture of a G-IDS Agent.

The Input Module receives intrusion detection data relevant to the system and supplies it to the Storage Module which stores it and provides it when necessary. The Analysis Engine tries to match data in the Storage Module and information about the real-time status of the Grid System which is being monitored. It sends any identified alerts to the Output Module which makes sure the relevant information is sent to the administrator and the appropriate action (run a script, send an alert, create a log entry) is taken. The most important task an agent needs to accomplish is to setup its neighbor view by the use of gossip algorithms. A variant of the T-Man algorithm [30] has been selected due to the compliance with this problem, and it is used for this purpose.

The system is efficient and avoids excessive network traffic by gathering information from other IDSs. This is facilitated by the use of the Intrusion Detection Exchange Protocol IDXP [14] to integrate the lower level IDS with G-IDS. IDXP is used by a large number of existing intrusion detection systems, including Snort. This makes the system more extensible, allowing different types of IDS to send data to the higher level detector through a common format without requiring any significant changes to the low level IDS.

Since a Grid System is composed of multiple heterogeneous administrative domains this is a very important advantage.

In case of attack detection an attempt to block the attacker is made and the rescheduling of the running jobs is done. In case of an Application Based attack, the running application could be moved to a different set of resources, which are considered more secure. For a Resource Based Attack some of the jobs running on those resources could be moved to minimize the damage that the attacker can cause.

We performed an analysis of system efficiency and scalability. We measured the time required for an agent to distribute alerts throughout its neighbors. Figure 8 shows the spread time when each agent transmits a specific number of alerts to all other agents. For a number of four agents and under 200 alerts transmitted by each agent, the spread time of alerts is lower than 300ms. It is worth mentioning that 200 hundred alerts correspond to a number of 400 agents, while no more than 100 agents are usually needed.

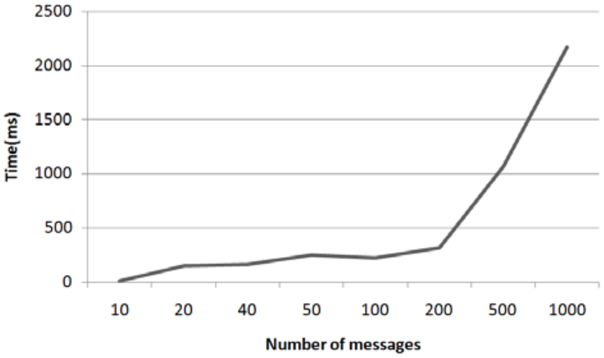


Fig. 8 – Message spread duration.

Other aspects regarding the quality, performance and behavior of the solution described in this section can be found in the papers [1, 2, 3, 4, and 5].

4. CONCLUSIONS

This paper presents a complete Grid Intrusion Detection solution developed at the NCIT of UPB. It describes novel mechanisms for the detection of complex attacks from the network level to the design of an IDS which protects the entire Grid.

Communications and information technology was declared national priorities in the beginning of 2001, and is considered an engine for the development of the Romanian economy. One of the most important issues that the Ministry of Communications and Information Society in Romania has concentrated on in the last years is the regulatory framework for the development of the Information Society services in Romania, which was established. The laws concerning the protection of people with regard to their personal data, the electronic signature, the cybercrime, the electronic commerce, e-procurement and e-tax are already in force. Nevertheless, new methods, mechanisms, platforms and services are needed to sustain the increasing number of users and features of e-Services.

ACKNOWLEDGEMENTS

The research presented in this paper is supported by national projects: “*SORMSYS – Resource Management Optimization in Self-Organizing Large Scale Distributes Systems*”, Contract No. 5/28.07.2010, Project CNCSIS-PN-II-RU-PD ID: 201, ant "*TRANSYS – Models and Techniques for Traffic Optimizing in Urban Environments*", Contract No. 4/28.07.2010, Project CNCSIS-PN-II-RU-PD ID: 238. The work has been co-funded by the Sectorial Operational Programme Human Resources Development 2007-2013 of the Romanian Ministry of Labour, Family and Social Protection through the Financial Agreement POSDRU/89/1.5/S/62557.

REFERENCES

1. ANDREEA CIRNECI, STEFAN BOBOC, CATALIN LEORDEANU, VALENTIN CRISTEA, CRISTIAN ESTAN, *Netpy: Advanced Network Traffic Monitoring*, Proceedings of INCOS 2009, International Conference on *Intelligent Networking and Collaborative Systems*, November 4–6, 2009, pp. 253–254.
2. CATALIN LEORDEANU, LEVNI ARIF AND VALENTIN CRISTEA, *Correlation of Intrusion Detection Information in Grid Environments*, Proceedings of the *Fourth International Workshop on P2P, Parallel, Grid and Internet Computing* (3PGIC-2010), held in conjunction with the CISIS-2010 Conference http://www.lsi.upc.edu/ fatos/3PGIC-2010/, pp. 463–368.
3. IONUT UNGUREANU, CATALIN LEORDEANU AND VALENTIN CRISTEA, *Grid-Aware Intrusion Detection System using Gossip Algorithms*, Proc. of 12th *International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, 2010.
4. CRISTINA AMZA, CATALIN LEORDEANU AND VALENTIN CRISTEA, *Hybrid Network Intrusion Detection*, IEEE International Conference on *Computer Communication and Processing* (ICCP 2011), 2011.
5. CATALIN RADU, CATALIN LEORDEANU, VALENTIN CRISTEA, *Using Cell Processors for Intrusion Detection through Regular Expression Matching with Speculation*, Proceedings of The Fifth International Conference on *Complex, Intelligent, and Software Intensive Systems* (CISIS-2011), 2011.
6. M. HUMPHREY, M.R. THOMPSON, AND K.R. JACKSON. *Security for grids*, Proceedings of the IEEE, **93**, *3*, pp. 644–652, march 2005.
7. STEVEN R. SNAPP, JAMES BRENTANO, GIHAN V. DIAS, TERRANCE L. GOAN, TODD L. HEBERLEIN, CHE L. HO, KARL N. LEVITT, BISWANATH MUKHERJEE, STEPHEN E. SMAHA, TIM GRANCE, DANIEL M. TEAL, DOUG MANSUR. *DIDS (Distributed Intrusion Detection System) – Motivation, Architecture, and An Early Prototype*, Proceedings of the 14th *National Computer Security Conference*, pp. 167–176, Washington, DC, October 1991.
8. EUGENE H. SPAFFORD AND DIEGO ZAMBONI. *Intrusion detection using autonomous agents*, Comput. Netw., **34**, pp. 547–570, October 2000.
9. NATHAN CAREY, ANDREW CLARK, AND GEORGE M. MOHAY. *IDS interoperability and correlation using IDMEF and commodity systems*, Proceedings of the 4th International Conference on *Information and Communications Security* (ICICS '02), pp. 252–264, London, UK, 2002, Springer-Verlag.
10. WOLFGANG BARTH. *Nagios: System and Network Monitoring*, No Starch Press, San Francisco, CA, USA, 2nd edition, 2008.
11. MARK JELASITY, ALBERTO MONTRESOR, AND OZALP BABAOGLU. *T-man: Gossip-based fast overlay topology construction*, Comput. Netw., **53**, pp. 2321–2339, August 2009.
12. T. BUCHHEIM, M. ERLINGER, B. FEINSTEIN, G. MATTHEWS, R. POLLOCK, J. BETSER, AND A. WALTHER. *Implementing the intrusion detection exchange protocol*, Proceedings of the 17th *Annual Computer Security Applications Conference*, Washington, DC, USA, 2001, p. 32, IEEE Computer Society.
13. B. NICOLAE, G. ANTONIU, AND L. BOUGE., *BlobSeer: How to enable efficient versioning for large object storage under heavy access concurrency*, in *Data Management in Peer-to-Peer Systems*, St-Petersburg, Russia, 2009.
14. V. CHANDOLA, A. BANERJEE, AND V. KUMAR, *Anomaly Detection: A Survey*, ACM Computing Surveys, **41**, *3*, 2009.
15. V. GOWADIA, C. FARKAS, AND M. VALTORTA, *Paid: A probabilistic agent-based intrusion detection system*, in *Computers & Security*, pp. 529–545, 2005.
16. BOGDAN GHIT, FLORIN POP, VALENTIN CRISTEA, *Intrusion Detection in Multi-Agent Systems, Intelligent Networking*, Collaborative Systems and Applications, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 235–256.

Received March 11, 2012