INTRUSION DETECTION ARCHITECTURE FOR GRIDS

Cătălin LEORDEANU[1], Valentin CRISTEA[2]

Acest articol prezintă o soluție completă pentru detecția intruziunilor pentru sisteme de tip Grid. La nivelul rețelei metodele prezentate în acest articol sunt create pentru a opri atacuri directe cum ar fi cele de Denial of Service sau încălcarea unor politici de securitate locale. În acest sens prezentăm un sistem nou pentru detecția intruziunilor care foloseşte o abordare hibridă bazată pe detecția de patternuri în combinație cu o rețea neurală pentru îmbunătățirea detecției.Folosim feedback-ul primit de la aceste sisteme locale de detecție a intruziunilor pentru a proteja întregul mediu Grid. La acest nivel putem vizualiza întregul sistem distribuit din punctul de vedere al intruziunilor şi putem detecta atacuri complexe îndreptate spre anumite aplicații sau grupuri de resurse. Această abordare corelează informația primită de la nivelul rețelei, împreună cu date de monitorizare şi identifică atacurile ce nu pot fi detectate de la nivelul rețelei.

This paper presents a complete intrusion detection solution for Grid Systems. On the network level the methods presented in this paper are intended to stop attacks such as Denial of Service or local policy violations. To achieve this goal we present a novel Intrusion Detection System which uses a hybrid approach based on a pattern matching engine and a neural network functioning in parallel to improve the detection efficiency. We use the feedback from these low-level intrusion detection systems to protect the entire Grid environment. At this level we have access to a global view of the Grid and we can also detect complex attacks focused on certain running applications ore groups of resources. This approach correlates the information received from the network level, as well as monitoring data from the Grid System and identifies attacks that cannot be detected at a local level.

Keywords: Grid systems, Security, Intrusion Detection

1. Introduction

Distributed Systems and Grids[2] in particular represent important advances in computing, enabling resource sharing, communication and problem solving in large scale, dynamic virtual organizations. However, these systems must be secured in order to offer a safe environment to run large scale applications. This is no easy task due to the dynamic nature of Grid Environments, the existence of different administrative domains and the large

[1] Assist., Dept.of Computer Science, University POLITEHNICA of Bucharest, Romania, e-mail: catalin.leordeanu@cs.pub.ro

[2] Prof., Dept.of Computer Science, University POLITEHNICA of Bucharest, Romania, e-mail: valentin.cristea@cs.pub.ro

number of users and resources. Security is one of the critical concerns that directly impacts the adoption rate of the Grid paradigm. This motivates a large number of research efforts and collaborative projects on this subject.

Some basic security mechanisms are already in place. The Globus Security Infrastructure(GSI)[1] is widely-used on Grid infrastructures. The typical usage of GSI in Grids exploits credential delegation to manage access rights. The foundations for these mechanisms can be also found in the WS-Security and WS-Trust which address the security concerns for web services[3] which are widespread in Grid Environments. GSI also ensures message integrity and authentication of the communicating entities using X.509 certificates. Transport Level Security (TLS) can also be employed to encrypt the entire communication, including the SOAP messages used when communicating with web services.

Intrusion Detection Systems(IDS) [5] have a very important role in Grid security management. For the execution of large scale applications there is clearly a need to detect intrusions and any other kind of dangerous events. This goal can be achieved through integration with lower level or Network IDSs [6] [7], each providing intrusion detection for a domain or subdomain of the Grid system. There are also a number of additional constraints like the ease of integration with external IDS.

The lower level or local IDS monitoring the resources where a part of the application is currently running must be able to send alerts to the Grid level Intrusion Detection System (G-IDS) when an intrusion is detected. The local IDS must employ a number of advanced methods and algorithms in order to efficiently detect a wide array of possible intrusions. On a higher level G-IDS also provides mechanisms to correlate separate alerts from different local IDS in the case of possibly related intrusions, which is another contribution of this paper.

This paper introduces a novel approach to Grid Intrusion Detection, based on the correlation of alerts received from lower level IDSs. The design of this system is modular and can be easily integrated with existing network intrusion detection system.

The rest of this paper is organized as follows. Section 2 presents the possible types of attack on a Grid System and provides solutions to efficiently detect and block them. Section 3 focuses on the network level of the environment and describes solutions to the detection of intrusions using network traffic analysis. Section 4 switches to the Grid level of the architecture and analyzes the intrusion detection challenges from the point of view of the entire Grid. Chapter 5 concludes this paper and presents directions for future research.

2. Types of attacks

Let us consider the following situation. Once mutual authentication is performed, a potential threat is that authenticated users may behave in a malicious way. The focus of our research is the detection of such intruders that may be performing Denial of Service attacks, crawling or other similar actions. These may be operations that a user has the appropriate access rights to do, but used in such a way that they would harm the system, or just consume bandwidth and decrease the overall performance. An attacker may also attempt to damage the Grid System from the outside, to disrupt its normal state or to compromise the running applications.

A typical threat for Grid systems in general consists in Denial of Service or Flooding attacks [4]. These attacks are very dangerous and involve sending large amounts of requests or malformed data to services inside the Grid. Grid Systems[8] must be monitored to detect such successful and unsuccessful attempts to breach security. This research report discusses the major concerns for intrusion detection in Grid and proposes a novel solution to effectively detect intrusions, building a Grid-Based Intrusion Detection System (G-IDS).

A possible intruder might also be interested in altering the entire execution of a complex application which is running on a large number of geographically distributed resources. Because of this the G-IDS must be aware of the running applications and should correctly identify a single intruder when receiving multiple alerts from lower level IDSs which are related to different resources belonging to the same application. To achieve this goal the G-IDS also receives real-time information from Grid monitoring systems which offers a global view of the context in which an attack might take place.

The scope of G-IDS is not to detect a node restriction violation and therefore it only works with threats generated by other applications. The distributed nature of G-IDS offers a global view of the entire Grid and so it is useful for decision making level.

As stated in our previous work in [11], we can distinguish between two different types of large scale attacks based on the intent of the attacker:

- *Application based attack.* This type of attack may happen when an attacker is actively attempting to compromise the resources where a certain application is running.
- *Resource based attack.* In this case an attacker may focus on a certain group of resources, either as part of a certain VO or which are in the same administrative domain.

Based on these attacker profiles and the information gathered from the Grid we can easily detect these kinds of attacks using a simple threshold. For example, if a certain percentage of the resources used to run an application are under attack then we can conclude it is an Application Based attack and raise the degree of threat for the rest of the resources used for that application. The same steps are valid in the case of a Resource Based Attack.

Another action which can be taken in case such an attack is detected, besides attempting to block the attacker, is the rescheduling of the running jobs. In case of an Application Based attack the running application could be moved to a different set of resources which are considered more secure or if a Resource Based Attack some of the jobs running on those resources could be moved to minimize the damage that the attacker can do.

3. Low level intrusion detection

Network administrators are interested in how their networks are used at a given time and the current traffic for a given segment of the network. We used the Netpy[28] project to develop an efficient Intrusion Detection System which would serve as a basis for our Grid Intrusion Detection approach and detect the threats which may appear at the network level. Since a Grid System is composed of large numbers of individual networks, each of them should be monitored and the IDS should detect as many attacks as possible, with a small degree of false positives.

Netpy is a network traffic analysis tool using Netflow data. We used its existing features to develop a complex IDS based on pattern matching, as well as anomaly detection. The IDS solution we propose is therefore capable of efficiently detecting known attacks as well as signaling unknown threats which do not follow a predefined pattern.

The first step in developing the IDS was to design a novel architecture which would take advantage of the Netpy network monitoring data and also incorporate the two intrusion detection mechanisms. Thus, the resulting architecture for the hybrid Netpy IDS is presented in Fig. 1.

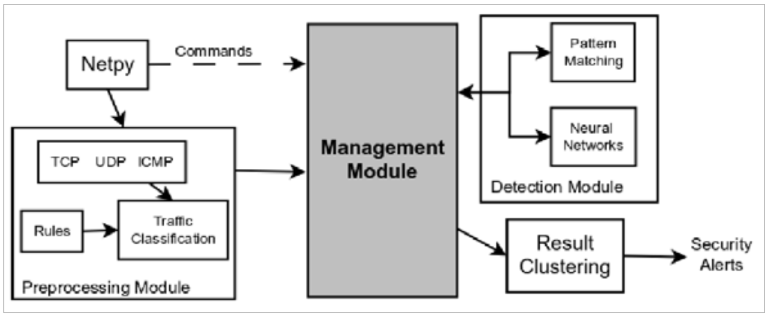


Fig. 1. Netpy Intrusion Detection Architecture

The monitored traffic is sent through Netpy to the Management Module which acts as a broker between the three phases of the IDS: preprocessing, detection and result clustering. Our approach for a hybrid network IDS is based on four separate modules: The management module, the preprocessing module, the detection module and result clustering.

The Management Module is a critical component of our architecture. It ensures the communication and data transfer between the preprocessing and detection phases. It also delivers the training data to the neural network component which is essential to the accuracy of the entire system by allowing the Detection Module to differentiate between normal and potentially malicious traffic.

The Preprocessing Module module links the network traffic monitoring tool to our detection architecture. It can perform basic operations such as filtering the raw traffic input to decrease the amount of data which needs to be analyzed. Furthermore, this module is very configurable with the use of dynamic rules written in an XML format. Through this module we can configure data source locations, logging information, various parameters of the monitored network or if we wish the IDS to ignore certain protocols or traffic sources. Moreover, the Preproceesing Module has the function of collecting the normal traffic data which is used to train the Neural Network detector.

The Detection Module is the central part of the architecture which receives the traffic data, analyzes it and outputs the alerts which were detected. It incorporates the two intrusion detection methods which complement each other. The detection methods function in the following way:

- Pattern Matching. This component implements a simple pattern matching algorithm which accepts patterns in an XML format. It continually matches the incoming network traffic against the existing attack patterns. The patterns can be loaded dynamically while the matching engine is running, which adds a degree of flexibility to the module.
- Anomaly Detection. In parallel to the pattern matching component an anomaly detection engine is also running and detecting possible attacks.

The two intrusion detection mechanisms function independently when analyzing traffic and the alerts which are detected are sent back to the Management Module, even though they may refer to the same malicious events.

The purpose of the *Result Clustering Module* this module is to attempt to reduce the number of false positives and also to provide the network administrator a clearer picture of the monitored network. Since the pattern matching algorithm is considered to have a lower false positive rate by design the security alerts which are generated by it are considered to have a high priority while the ones generated by the anomaly detection engine have a low priority. The security alerts are also grouped by timestamp. In the case of heavy network traffic any of the presented intrusion detection mechanisms may send multiple alerts which may belong to the same actual attack. We defined a time threshold which is used to determine if two security alerts are close in time or not. If the clustering module detects multiple alerts which are similar in nature, belong to the same traffic segments and the distance between them is less then the threshold then they will be considered as part of the same attack and only a single notification will be sent to the network administrator.

4. Grid-level intrusion detection

Grid Systems usually have a large number geographically dispersed resources, with many active users running complex applications. These characteristics add greatly to the difficulty of offering a high security level for Grid Systems. Because of the heterogeneous characteristic of Grids, the concept of intrusion detection becomes of interest in an environment where data security is essential and there is clearly a need to detect any attacks and other dangerous events. Providing a reliable Intrusion Detection System(IDS) [5] is the focus of our work and in this paper we will present our approach for a fault tolerant gossip based IDS. The proposed solution is also capable of detecting complex attacks specific to Grid Systems in an efficient and timely fashion.

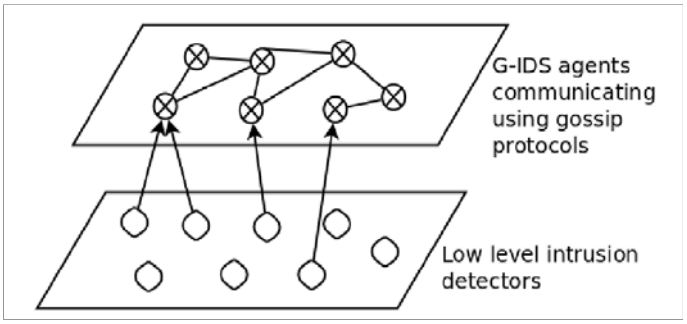

G-IDS agents communicating using gossip protocols
Low level intrusion detectors

Fig. 2. Communication model

To be able to detect Grid-based threats we needed a new IDS approach and to create new algorithms to match possible attacks. The layered approach for our Grid Intrusion Detection solution ensures the fact that local threats are detected at the lower levels by IDSs which also send information to the G-IDS and that global threat. Such threats are composed of attacks on different resources in different administrative domains, are detected directly by the G-IDS. This type of organization can be seen in Fig. 2. The large scale threats can only be detected by an entity which is able to have a global overview of the entire Grid System.

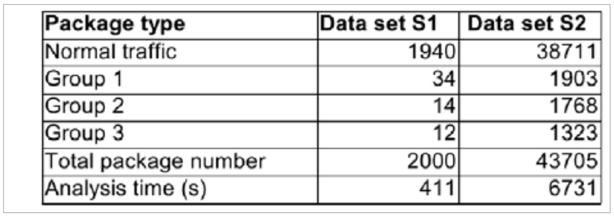
G-IDS is an agent based intrusion detection system focused on the detection of complex attacks in Grid Systems. As its name suggests the application is designed to work in a Grid Environment. Our approach uses an agent platform to ensure scalability, fault tolerance and load balancing. With these goals in mind we present in the following paragraphs the basic components of our application with some insights where it is necessary. Since this is an agent based architecture, there are a number of different entities, each of them receiving alerts from lower level IDSs scattered around the Grid, as represented in Fig. 2.

The basic operating principle is that each agent should only communicate with a small part of other agents, called neighbors. This behavior of the algorithm ensures that all unnecessary communication will not take place and that the number of messages passed between the agents will be kept to a minimum. We assume that an agent can deal with a lot of tasks independently and can support network or communication errors as well.

The most important task an agent needs to accomplish is to setup its neighbor view which uses gossip algorithms. To achieve this goal we used a variant of the T-Man algorithm [12] which is suitable for our problem. Since each agent only communicates with its neighbors we divide the entire Grid between the available agents. Using the gossip algorithms the system reconfigures itself every time a new agent is started or one fails. This division of resources among agents is developed to take into consideration geographical distribution of the nodes, Virtual Organizations, and related resources that are under surveillance.

5. Test cases

To validate our approach we tested the network level Intrusion Detection module using large sets of data. We proved that it is an efficient module, capable of detecting a large number of possible attacks by analyzing the network traffic monitored by Netpy.



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. Structure of the experimental data sets

To test the intrusion detection capabilities we used two data sets for the experiments. 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 presented an equal distribution of malicious events in the 3 possible groups. The exact structure of the data sets is shown in Fig. 3.

The proposed IDS detected the following number of alerts:

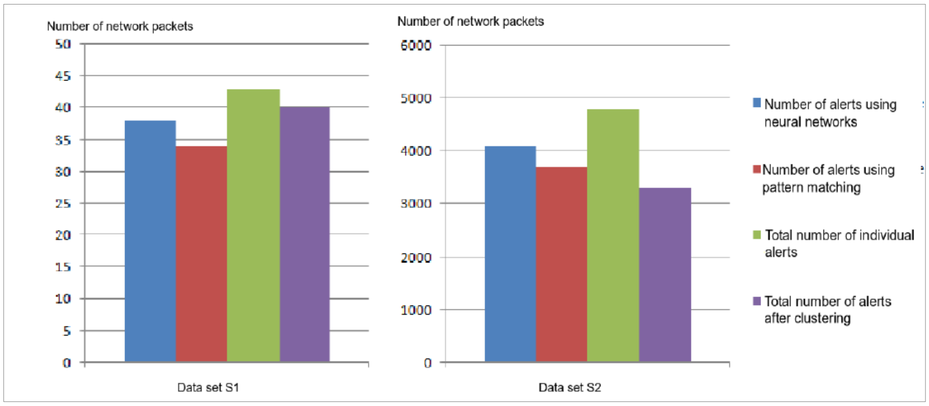


Fig. 4. Exchanged messages with peer sampling

Fig. 5 shows that using both detection methods we are able to detect more intrusions than with any single one. We can conclude that the neural network improves the overall detection capability of the IDS by detecting attacks which are ignored by the pattern matching component. Its efficiency is also due to the fact that the training data for the neural network was first classified using the pattern matching algorithm. This ensures a high quality training data which greatly influences the accuracy of the detection process. Furthermore, the clustering of the alerts eliminates duplicate alerts by approximately 22%, thus reducing the load for the network administrator.

The efficiency of the detection process can be assessed using two metrics. First of all, the false positive rate needs to be very low to ensure the fact that the IDS will send as few false alarms as possible. Using our approach the false positive rate for the two data sets is below 0,03% which proves the usability of the Netpy IDS. Another important metric is the real security alert detection rate, meaning the number of real security alerts which were detected compared to the total number of attacks in the data sets. We achieved a detection rate of 99,3% for both the data sets.

For the entire Grid we defined a threshold t, which was given an initial value. If we received alerts from all the resources where some of the application's nodes are running, it was very easy to determine a set of resources with a high risk. If the percentage of attacked nodes from the total nodes of a list of resources used by an application was higher than the threshold then we assumed that the rest of the resources were also at risk, according to the policies of Application Based Attacks.

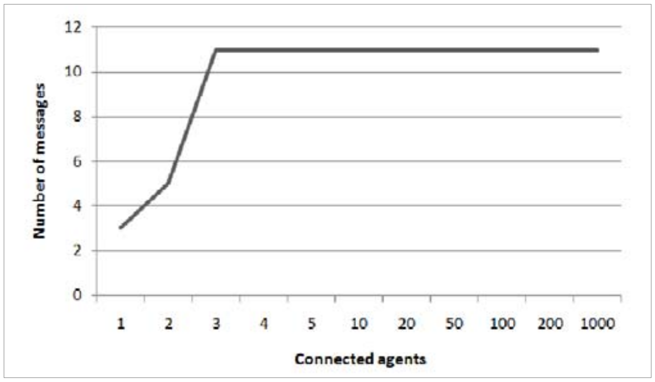


Fig. 5. Exchanged messages with peer sampling

In Fig. 5 we can see that the number of exchanged messages increases only when there are a small number of agents in the system. When the number of agents increases beyond the size of the neighbors view the number of messages exchanged between agents in this stage is constant due to the fact that an agent only has to communicate with a fixed number of agents. The implementation of a peer sampling service is programmed to be developed in a later version of the G-IDS application.

For these tests we used a simple attack pattern established a threshold of 60%. Out of 10 resources under surveillance, generating alerts for 6 of them led the system to correctly identify the remaining 4 as being in danger and correctly generate actions to protect them.

6. Conclusions and future work

This paper presents a complete Grid Intrusion Detection solution. 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.

On the network level we proved that using a hybrid intrusion detection system provides increased detection capabilities. We combined a basic pattern matching engine with a neural network detection component to detect anomalies in the network traffic. This novel approach is able to efficiently detect known classes of attacks, as well as unknown ones. Since the two detection solutions run in parallel we also provide a method to filter and group the security alerts to minimize the number of notifications which will be sent to the network administrator. The proposed IDS was developed as a module for the Netpy traffic monitoring an analysis tool.

We intend in the future to refine the detection capabilities of the proposed Netpy IDS to enable the detection of a larger array of attacks. Furthermore. we will continue to evaluate the intrusion detection capabilities of the proposed solution using larger and more diverse data sets. Such an IDS solution would also benefit from the addition of a module to continually adapt the existing attack patterns and add new ones based on the feedback received from the detection module.

Since our approach is based on a multi-agent system we also intend to protect it against malicious activity. There is always a need to create a robust system, capable of functioning correctly in the case of external attacks. An agent could attempt to inject false data into the system create the illusion of an attack or to simply disturb the Grid system. The requirement of a consensus between a number of agents before the intrusion detection information is considered accurate will solve this problem.

REFERENCES

[1]. *V. Welch, F. Siebenlist, I. Foster, J. Bresnahan, K. Czajkowski, J. Gawor, C. Kesselman, S. Meder, L. Pearlman, S. Tuecke,* Security for grid services, International Symposium on High Performance Distributed Computing (HPDC-12), 2003.

[2]. *I. Foster, C. Kesselman, S. Tuecke,* The Anatomy of the Grid: Enabling Scalable Virtual Organizations, International Journal of Supercomputer Applications, 2001.

[3]. *S. Weerawarana, F. Curbera, F. Leymann, T. Storey, D.F. Ferguson,* 2005 Web Ser- vices Platform Architecture: Soap, Wsdl, Ws-Policy, Ws-Addressing, Ws-Bpel, Ws-Reliable Messaging and More. Prentice Hall PTR.

[4]. *D. Moore, C Shannon, D.J. Brown, G.M. Voelker*, Inferring Internet denial-of-service activity,ACM Transactions on Computer Systems (TOCS), 2006.

[5]. *Yu, Zhenwei, Tsai, J.P. Jeffrey, Th. Weigert*, An adaptive automatically tuning intrusion detection system, ACM Transactions on Autonomous and Adaptive Systems (TAAS), 2008.

[6]. *B. Mukherjee, L. Heberlein, K. Levitt*, Network intrusion detection, IEEE Network, May/June 1994.

[7]. *H. Debar, M. Dacier, A. Wespi*, Towards a taxonomy of intrusion-detection systems, Int. J. Computer and Telecommunications Networking, 1999.

[8]. *I. Foster, C. Kesselman, S. Tuecke*, The Anatomy of the Grid: Enabling Scalable Virtual Organizations, International Journal of Supercomputer Applications, 2001.

[9]. *C. Leordeanu, L. Arif, V. Cristea*, Correlation of Intrusion Detection Information in Grid Environments, Proceedings of the Fourth International Workshop on P2P, Parallel, Grid and Internet Computing (3PGIC-2010)

[10]. *M. Jelasity, O. Babaoglu, T-Man*: Gossip-based Overlay Topology Management, Engineering Self-Organising Systems, **Vol. 3910**/2006

[11] *C. Leordeanu, L. Arif, V. 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

[12] *M. Jelasity. O. Babaoglu, T-Man*: Gossip-based Overlay Topology Management, Engineering Self-Organising Systems, **Vol. 3910**/2006