Intrusion Detection For Computational Grids

Alexandre Schulter, Kleber Vieira, Carols Westphal, Carla Westphal,Sekkaki Abderrahim

Abstract—**Current intrusion detection technology is limited in providing protection against the intrusions that may violate the security of computational grids. We present the problem of grid intrusion detection, describe the requirements of a system to detect them, propose a grid intrusion detection method, and show how it overcomes the limitations by integrating the detection of the typical host computer and network attacks with the detection of grid-specific attacks and user behavior anomalies. This integration is evaluated with a case study that makes use of simulations and a prototype implementation.**

I. INTRODUCTION *(HEADING 1)*

Computational grids are emerging as tools to facilitate the secure sharing of resources in heterogeneous environments (Foster and Kesselman, 2003). Security is one of the most challenging aspects of grid computing (Humphrey et al., 2005) and Intrusion Detection Systems (IDS) have an important role in grid security management. IDSs are responsible for the detection of intrusions in information systems and the responses to them, usually alert notifications sent to the security managers (Allen et al., 2000). The intrusions can be characterized as unauthorized use by external parties or abuse of the system by insiders. Typical host-based IDSs and network-based IDSs (Debar et al., 1999) can be deployed in a grid environment to improve its security. However, they cannot properly detect grid intrusions. The detection of these intrusions poses new challenges and current intrusion detection technology is limited in providing protection against them. This paper describes a grid intrusion detection method that overcomes the limitations. This paper is organized as follows: Section 2 provides the background of the problem. Section 3 analyzes the shortcomings of current technology, what requirements need to be satisfied, and describes a grid intrusion detection method proposed to solve the problem. Section 4 describes the mechanisms that are necessary to make the method practicable. Section 5 describes a case study performed to evaluate the mechanisms. The paper concludes with Section 6.

A. *Grid Intrusions*

A grid is subject to several accidental and intentional security threats, including threats to the integrity, confidentiality and availability of its resources, data and infrastructure. Also, when a grid with large computing power and storage capacity is misused by an ill-intentioned party for malicious purposes, the grid itself is a threat against society (Navqi and Riguidel, 2005). Intentional threats are imposed by insiders and external intruders. Insiders are legitimate grid users who abuse their privileges by using the grid for unintended purposes and we consider this intrusive behavior to be detected. An intrusion consists of an attack exploiting a security flaw and a consequent breach which is the resulting violation of the explicit or implicit security policy of the system (Lindqvist and Jonsson, 1997). Although an intrusion connotes a successful attack, IDSs also try to identify attacks that don't lead to compromises (Allen et al., 2000). “Attacks” and “intrusions” are commonly considered synonyms in the intrusion detection context. The underlying network infrastructure of a grid, being an important component of the computing environment, can be the object of an attack. Grid applications running on compromised hosts are also a security concern. We consider attacks against any network or host participating in a grid as attacks against that grid, since they may directly or indirectly affect its security aspects. Grid systems are susceptible to all typical network and computer security attacks, plus specific means of attack because of their new protocols and services. The targets that are possibly vulnerable are the protocol stack; network devices; processes running in kernel space, such as operating system daemons; and processes running outside kernel space, such as grid middleware, grid applications, and any non-grid applications running with either root or user privileges.

Our classification of grid intrusions is given as follows:
(a) Unauthorized access: A break-in committed by an intruder that masquerades as a legitimate grid user. It is made possible by obtaining the user’s password through stealing, brute-force cracking, guessing, or the careless exposure by the user himself. Attacking the authentication service is another possibility, and this may result in attack trails left at the service location.
(b) Misuse: This may be a consequence of an (a) unauthorized access or the abuse of privileges by a legitimate user (insider) and generally results in an observable user behavior anomaly. What is a misuse of grid resources depends on the defined policies, and those should consider aggressive utilization, user mistakes and malicious usage.
(c) Grid Attack: Attacks performed with the help of tools or exploit scripts that target vulnerabilities existent in grid protocols, services and applications. They may appear in the form of denial-of-service attacks, probes, and worms, and may leave their trails at several locations of a grid’s infrastructure.
(d) Host or Network-specific intrusion: whereas (a), (b), and (c) are grid-specific intrusions, these are any of the typical computer host and network attacks (Kendall, 1999).

A difference between intrusions in grids and in non-grid distributed systems is the potentially greater speed, consequences, and damages, as the massive resource aggregation, wide user access, efficient and automated

978-2-9532443-0-4 © 2008 ESRGroups France

resource allocation provided by grids can be used by intruders for their advantage.

B. *Motivation and Related Works*

In the previous sub-section we described four kinds of intrusions that may violate grid security: (a) unauthorized access, (b) misuse, (c) grid attack, and (d) host/network intrusion. To avoid unwanted consequences of these intrusions, typical host-based and network-based IDSs (Section 2.2) can be deployed in a grid environment and provide protection against attacks that explore vulnerabilities in its nodes (hosts) and networks. This solution is not complete, as it provides protection against (d) host and network-specific intrusions but not against (a, b, c) grid-specific intrusions. The signature database of typical IDSs can be updated to identify trails of (a) unauthorized accesses and (c) grid attacks left at hosts and network packets. This is not a complete solution either, because (c) grid attacks may leave trails at more than one location and they may become evident only by correlating the trails identified by the IDSs. Furthermore, a HIDS is unable to properly detect grid users committing (b) misuse, because they analyze the behavior of users in their local contexts and since grid users are allowed to use multiple resources from different domains at the same time or consecutively, the analysis must be done in the scope of the grid as a hole. Therefore, a different approach to the problem is needed to overcome the deficiencies. The need for grid-based intrusion detection systems (GIDS) was first mentioned in (Liabotis et al., 2003), (Talnar et al., 2003), and (Leite et al., 2003), although solutions to the problem were not described. An efficient and scalable solution for storing and accessing audit data collected from grid nodes was proposed by Kenny & Coghlan (Kenny and Coghlan, 2005), but there was no mention on how to use the data to identify intrusions. Choon and Samsudin (2003) described a grid-based IDS architecture that consists of agents located at grid nodes responsible for collecting and sending host audit data to storage and analysis servers, but since IDSs are known to consume considerable processing time and storage space (Allen et al., 2000), their centralized solution is not scalable with the number of nodes under analysis. The Grid Intrusion Detection Architecture (GIDA) proposed by Tolba et al. (Tolba et al., 2005) solves the scalability problem by distributing the intrusion detection problem among several analysis servers. Both (Choon and Samsudin, 2003) and (Tolba et al., 2005) concentrate on the detection of anomalies in the interaction of grid users with resources, which is the result of (b) misuse. But they lack proper detection of (a) unauthorized accesses and (c) grid attacks, as these may not cause anomalies and are more accurately detected by signature-based systems (Section 2.2). Furthermore, none of the architectures aim to provide protection against (d) host and network-specific intrusions. Fang-Yie et al. (2005) proposed an IDS called Performance-based Grid Intrusion Detection System (PGIDS) in which grid nodes are allocated through load balancing to analyze collected network traffic and search for network denial-of-service attacks. The system uses a grid's abundant resources to detect intrusion packets, but it does not detect attacks to the grid itself and it only looks for (d) network attacks, therefore it acts as a NIDS, rather than a GIDS. The shortcomings of the available solutions motivate us to propose new approach. The problem is further analyzed in the next section.

II. PROPOSED APPROACH

A. *Problem Analysis*

Grid intrusion detection is a process that involves the gathering of information available at its networks and nodes (host computers), and the identification, based on the evaluation and correlation of the gathered data, of attacks against all the possible vulnerable targets, as well as anomalies in the interaction of grid users with resources (Section 2.3). As discussed in the Section 2.4, current intrusion detection technology fails to provide protection against all the intrusions that may violate grid security. The deployment of typical NIDS and HIDS in a grid environment improves the security, but it is not a complete solution because they are not able to properly detect grid-specific intrusions. A node may not be dedicated to the grid, and its resources may be shared by local users and external users allocated through the grid to resources belonging to that node. Therefore, user interaction must be examined in the scope of individual nodes, in the case of local users, and in the scope of the grid as a hole, in the case of grid users. Also, as discussed in Section 2.4, the previously published GIDS architectures (Choon and Samsudin, 2003; Tolba et al., 2005) are designed to properly detect user behavior anomalies, but are deficient in detecting host attacks, network attacks and grid-specific attacks that don't cause anomalies. We believe the following three basic requirements need to be satisfied by a grid-based intrusion detection system:

(x) Coverage: must provide detection of (a) unauthorized access, (b) misuse, (c) grid attack, and (d) host/network-specific intrusion (Section 2.3);

(y) Scalability: must be scalable with the number of grid resources and users;

(z) Grid compatibility: must suit and benefit from the grid environment.

While current solutions to the grid intrusion detection problem aim to satisfy the requirements of (y) scalability (Kenny and Coghlan, 2005; Tolba et al., 2005) and (z) grid compatibility (Choon and Samsudin, 2003; Tolba et al., 2005), they lack in (x) coverage. Next sub-section describes our proposed solution which aims to satisfy these three requirements.

B. *An Intrusion Detection Method for Computational Grids*

For intrusion detection in computational grids we recommend a method in which GIDS is a high-level component that utilizes functionality of lower-level HIDS and NIDS (Figure 1) provided through inter-IDS communication. This makes possible the reuse of intrusion detection software already available, avoiding re-implementation of functionality. GIDS integration with the lower-level is the method's core and is illustrated in Figure 1. In this method, to achieve the desired security level for the grid, HIDS and/or NIDS are installed at certain grid nodes and network domains and work integrated with GIDS sending relevant information for the detection of intrusions. To achieve the maximum security level, each grid node and grid network domain must have a lower-level IDS installed. In this case, the several NIDS located in each grid network domain capture network audit data and look for protocol anomalies and attack trails existent in network packets. Also, each grid node has a HIDS installed that collects and examines host audit data to identify evidence left by attacks and resource usage anomalies caused by local users. GIDS uses the audit data (i) shared by the lower-level IDSs to identify grid attacks and to compare the behavior of grid users with their previously built historical profiles. The grid security manager is (ii) alerted whenever an intrusion is detected by GIDS or an alert is (iii) sent by the lower-level IDSs.

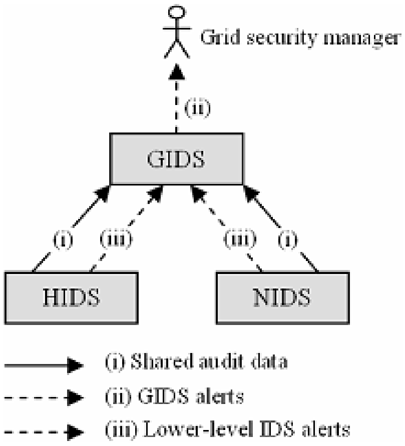

Fig. 1 - Integration of GIDS with lower-level IDSs

Next section describes the mechanisms needed for integrating the IDSs. Being this integration feasible, we must then know how can a GIDS that is integrated with lower-level IDSs satisfy the requirements listed in Section 3.1. In Figure 1, HIDS and NIDS are depicted in an abstract manner, since their architectures vary. Figure 2 shows the architecture of a GIDS example that is closer to reality. In this example, GIDS is composed of Agents, Analyzers, and a Scheduler. The organization of HIDS and NIDS components is illustrative and the audit information they (i) share with GIDS Agents is (iv) stored in Grid Information Databases. Every time a user accesses the grid, GIDS Schedulers (v) consult the user profile stored in a database and, depending on the demanded computing power for audit data analysis, (vi) submit one or more Analyzer jobs to nodes with available computing resources. The jobs (vii) exchange data with the databases in order to analyze user behavior and update the profiles. The Analyzers are also responsible for (viii) correlating the (iv) stored audit data to identify grid attacks.

To show how the GIDS example satisfies the (x) coverage requirement, consider a scenario where a grid is protected by it and an intruder that follows these steps:

(1) The intruder launches a buffer overflow attack (Kendall, 1999) against an operating system (OS) process running on a grid node. The attack is successful and he is then able to execute arbitrary code.

(2) Now with OS root privileges, he runs an exploit script and impersonates (Kendall, 1999) a user with grid privileges, gaining facilitated access to several nodes.

(3) Continuing the malicious activity, he uses several grid nodes to run a distributed application.

(4) The application launches a coordinated network denial-of-service (DoS) attack (Kendall, 1999) against an external target.

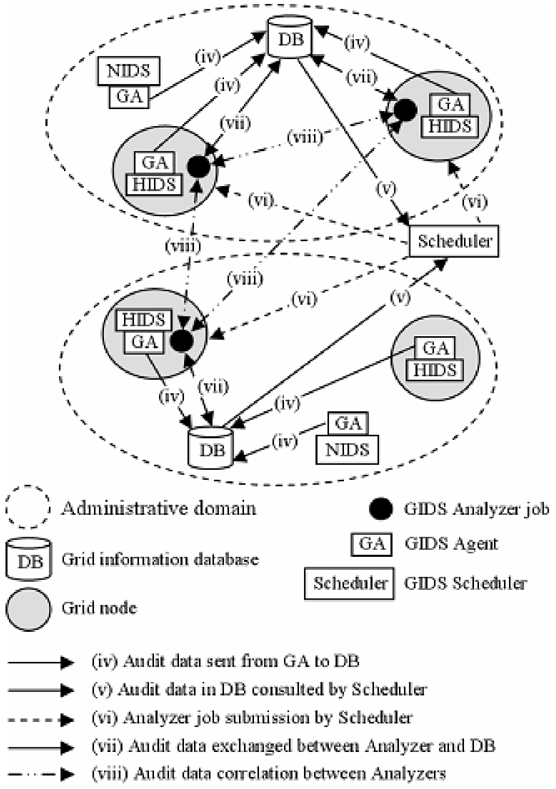
DB
NIDS
GA
(iv)
(iv)
(iv)
(vii)
GA
HIDS
(vii)
(viii)
GA
HIDS
(vi)
(v)
(vi)
(viii)
Scheduler
(viii)
(vi)
(v)
HIDS
GA
GA
HIDS
(iv)
(vii)
(iv)
GA
NIDS
(iv)
DB
Administrative domain
Grid information database
Grid node
GIDS Analyzer job
GIDS Agent
GIDS Scheduler
(iv) Audit data sent from GA to DB
(v) Audit data in DB consulted by Scheduler
(vi) Analyzer job submission by Scheduler
(vii) Audit data exchanged between Analyzer and DB
(viii) Audit data correlation between Analyzers

Fig. 2 - Architecture of a GIDS example

The first step characterizes a (d) host intrusion detectable by HIDS. Supposing it's not detected, the intruder proceeds to the second step, which characterizes a (c) grid attack and a consequent (a) unauthorized access, both detectable by GIDS. If not stopped at that point, the intruder gets to the third step, where GIDS compares his behavior with the historical profile of the user he impersonated to identify (b) misuse. If somehow GIDS fails to identify a behavior anomaly, in the fourth step NIDS is responsible to detect the (d) DoS attack trails. In conclusion, in this scenario the GIDS example covers (a), (b), (c), and (d) intrusions, satisfying the requirement of (x) coverage. The system example is designed to distribute the detection problem among its components in order to achieve (y) scalability and, since it benefits from the grid by consuming its computing resources, it achieves (z) grid compatibility.

III. IDS INTEGRATION

The integration of a GIDS with lower-level IDSs (HIDS and NIDS) can be achieved, as required by the grid intrusion detection method described in the previous section, if the following requirements are supported: (a) the sending of alerts from lower-level IDSs to warn GIDS about intrusions detected locally; (b) the sending of alerts from lower-level IDSs to warn GIDS about grid attack trails; (c) the sending of audit data from HIDS to GIDS; (d) standard communication.

The lower-level IDSs must (a) alert GIDS about any intrusions detected locally in their acting domain. Alerts of any attack trails that can be correlated by GIDS with other trails to detect grid-specific attacks must also be (b) sent by the IDSs. To identify misuse committed by grid users, GIDS must analyze their behavior and this is done with resource usage data. Host-based audit data sources are the only way to retrieve information about user activities in a grid and, therefore, HIDS are responsible for (c) sending audit data to GIDS. As lower-level IDSs might be heterogeneous and write their alerts in different formats, it is hard to aggregate precise information for automatic processing without the use of communication standards. The interoperability between distinct intrusion detection systems demands that they speak the same language (Brandão, 2006). For these reasons, we defined the requirement of (d) standard communication between HIDS/NIDS and GIDS. The IDMEF message format (Debar et al., 2006) and the IDXP protocol (Feinstein et al., 2002) being developed by the IETF Intrusion Detection Working Group (IDWG) were chosen to address the (a), (b), (c), and (d) requirements. The interactions between users, resources, lower-level IDSs, and GIDS are depicted in Figure 3. The user and his application (i) interact with the Grid Resource Broker (GRB) to negotiate the requirements to process the application. The GRB communicates with Grid Service Providers (GSP) to find adequate nodes where the application can be executed and, then, (ii) submits application jobs for processing at nodes of the chosen GSP.

The nodes provide the GSP service by executing the jobs and the Grid Resource Meter (GRM) measures resource consumption (iii) extracting operating system (OS) data through the grid middleware. The data is filtered by the GRM and is used to generate Resource Usage Records (RUR) (Lim, 2005), which are OS-independent standardized records that represent the utilization of heterogeneous resources.The RURs are sent to (iv) HIDS and to (v) other services that are possibly offered by the GSP, such as monitoring and accounting services. Besides receiving data from the GRM, HIDS also (vi) receives audit data from the OS and is able to generate alerts warning about all kinds of intrusions that leave trails at hosts. Each operating system creates audit records in a different manner, and a standard and popular mechanism to send them is the syslog protocol (Debar et al., 1999). GIDS, on the other hand, receives and analyzes (vii) IDMEF alerts sent through the IDXP protocol by the various HIDS and NIDS installed at the GSP nodes and networks. The content of these alerts refers to typical host and network attacks, trails of possible grid attacks, and user RURs used to identify the occurrence of misuse. The flexibility of the IDMEF message format allows all the information contained in a RUR to be mapped to a message of this kind.

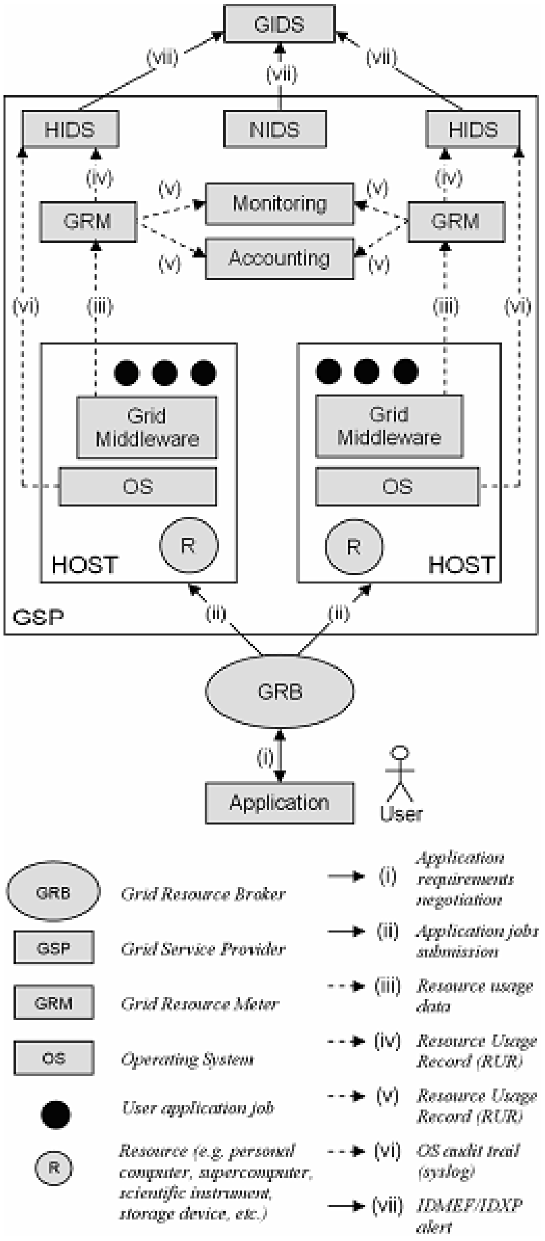

Fig. 3 - Interactions between users, resources, and IDSs

IV. CASE STUDY

The mechanisms described in the previous section serve to satisfy the (a), (b), (c), and (d) equirements, as explained. Evaluating the mechanism proposed to satisfy requirement (a) is not necessary, since that mechanism consists only of sending alert messages from one IDS to another, and most available lower-level IDSs support this (OSSEC-HIDS, 2006; Snort, 2006). Evaluating the mechanisms for (b), on the other hand, can not be done realistically, since to date no signature databases of grid attacks were found available for testing purposes. To evaluate the mechanisms for (c) and (d), a case study of a simulated grid environment was performed with the GridSim tool (Sulistio et al., 2005). The simulated grid consisted of several computational hosts distributed in a few administrative domains.

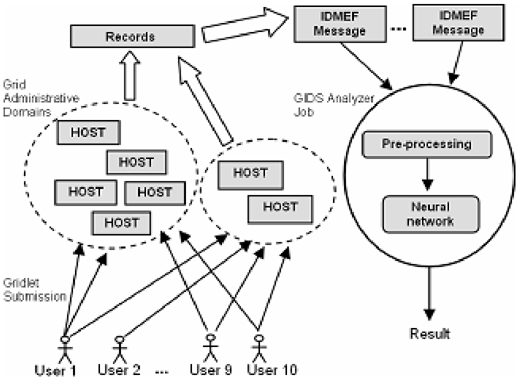

Fig. 4 - Simulation environment

Ten simulated users submitted applications that were scheduled for processing at the grid nodes. These applications are called gridlets in GridSim (Figure 4). Records describing the resource usage (e.g. CPU time, memory usage) and the cost demanded by each gridlet were captured and converted to IDMEF messages. These messages were sent to a prototype GIDS analyzer job, simulating the HIDS interactions with GRM, grid nodes, and GIDS, as illustrated in Figure 3 in the previous section. The GIDS prototype is centralized, simpler than the distributed GIDS example presented in Section 3.2. For this case study, the program was implemented as a Matlab script (MathWorks, 2006) that applied a feed forward neural network (Mehrotra et al., 2000) to detect user behavior anomalies. Table 1 sums up the technologies utilized in this case study.

Table 1 - Technologies utilized

Components	Technologies
Simulation toolkit	GridSim 3.3 (GridSim, 2006)
IDMEF messaging	Java, JavaIDMEF v0.94 beta (JavaIDMEF, 2006), Java BEEP Core 0.9.08 (BEEP Core, 2006)
GIDS prototype	Matlab 6.5 (MathWorks, 2006)

The neural network was trained to detect significant deviations of behavior. To achieve this, a training phase was executed in which the network was taught to identify users that suddenly started to submit gridlets that demanded computational resources greatly different from what were demanded by their past gridlets. Experiments were performed to verify if the net really learned to detect great modifications in user behavior. In each experiment, among the ten users a random number of them had anomalous behavior and the neural net did identify most of them as possible intruders.

In the training and experimentation phases the net was fed with a certain volume of input data obtained from the last gridlets submitted by the users. A lower number of gridlets resulted in a lower volume of data analyzed by the net and a greater difficulty for it to learn and to identify anomalies caused by the possible intruders. Figure 5 shows that increasing the number of gridlets under analysis results in a lower percentage of detection errors, that is, a lower percentage of false alarms and intruders not identified. Besides a neural network, GIDS can apply other types of detection techniques. But, although it seems to be able to detect anomalies well, it is not an objective of this work to identify the best type of neural network or any other technique. This case study, though, shows that integrating lower-level IDSs such as HIDS with a GIDS using IDMEF messaging is useful to detect grid intrusions.

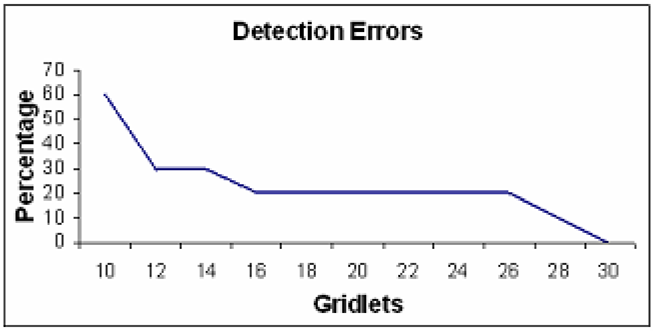

Fig. 5 – Percentage of detection errors

V. CONCLUSIONS

The purpose of this work is to describe an approach that overcomes the weaknesses of the available solutions to the grid intrusion detection problem. As discussed in Section 2, current technology is limited in detecting all the kinds of attacks that may violate the security of a grid. Typical IDSs cannot properly identify grid-specific attacks and grid users misusing resources. The available GIDS architectures also lack protection against grid attacks and typical computer host and network attacks. In Section 3 we listed basic requirements that need to be satisfied by a GIDS: coverage, scalability, and grid compatibility. Related works on the subject of grid intrusion detection describe solutions which try to achieve scalability and grid compatibility, but lack in achieving complete coverage protection against the possible grid intrusions. We described a grid intrusion detection method in which GIDS is a high-level component that works in an integrated manner with lower-level IDSs (NIDS and HIDS). Then, assuming that integrating the IDSs was feasible, we showed with an example that this method can be used to satisfy the basic GIDS requirements. In Section 4 we presented mechanisms to integrate GIDS with lower-level IDSs. The use of standard protocols and formats was focused: IDMEF, IDXP, RUR, and syslog. In Section 5 the case study performed with a simulated grid environment to evaluate the mechanisms was described. It demonstrated that the integration of lower-level IDSs with a GIDS using IDMEF messaging is possible and useful to detect the grid intrusions, although a complete case study involving all the types of grid intrusions was not performed, since signature databases of grid-specific attacks have not been made available to the scientific community yet. Research topics to be considered for future work are the distributed GIDS architecture that was left as an idea in Section 3.2, the impact that a grid-wide intrusion detection service has on a grid's performance, and grid-specific attacks, including languages and tools for their manipulation. Also, other requirements could be considered for GIDS, such as accuracy, fault tolerance, timeliness, and performance.

References

CHOON, Ong Tian; SAMSUDIN, Azman. Grid-based Intrusion Detection System. In: ASIA-PACIFIC CONFERENCE ON COMMUNICATIONS, 9., 2003, Penang, Malaysia. Proceedings… [S.l.: s.n.], 2003. v. 3, p. 1028-1032.

DEBAR, Hervé; DACIER, Marc; WESPI, Andréas. Towards a Taxonomy of Intrusion-Detection Systems. International Journal of Computer and Telecommunications Networking, v. 31, n. 9, p. 805-822, April 1999, Special Issue on Computer Network Security.

DENNING, Dorothy E.. An Intrusion-Detection Model. IEEE Transactions on Software Engineering, USA, v. 13, n. 2, p. 222-232, February 1987, Special Issue on Computer Security and Privacy.

FANG-YIE, Leu et al. A Performance-based Grid Intrusion Detection System. In: IEEE INTERNATIONAL COMPUTER SOFTWARE AND APPLICATIONS CONFERENCE (COMPSAC), 29., 2005, Edinburgh, Scotland, UK. Proceedings… [S.l.]: IEEE Computer Society, 2005. p. 525-530.

FEINSTEIN, Benjamin; MATTHEWS, Gregory; WHITE, John. The Intrusion Detection Exchange Protocol. Intrusion Detection Working Group (IDWG). IETF Internet-Draft, 2002. Available: <http://www.ietf.org/internet-drafts/draft-ietf-idwg-beep-idxp-07.txt>.

FOSTER, Ian; KESSELMAN, Carl. The Grid 2: Blueprint for a New Computing Infrastructure. 2. ed., San Francisco: Morgan Kaufmann, 2003. 748 p.

GRIDSIM, A Grid Simulation Toolkit for Resource Modelling and Application Scheduling for Parallel and Distributed Computing (2006). Available: <http://www.gridbus.org/gridsim/>.

HUMPHREY, Marty; THOMPSON, Mary; JACKSON, Keith R. Security for Grids. Proceedings of the IEEE, v. 93, n. 3, p. 644-652, March 2005, Special Issue on Grid Computing.

JAVAIDMEF, JavaIDMEF Library (2006). Available: <http://sourceforge.net/projects/javaidmef/>.

KENDALL, Kristopher. A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems. Cambridge: MIT, 1999. Master Thesis.

KENNY, Stuart; COGHLAN, Brian. Towards a Grid-wide Intrusion Detection System. In: EUROPEAN GRID CONFERENCE (EGC), 2005, Amsterdam, The Netherlands. Proceedings… [S.l.]: Springer, 2005. p. 275-284.

LINDQVIST, Ulf; JONSSON, Erland. How to Systematically Classify Computer Security Intrusions. In: IEEE SYMPOSIUM ON SECURITY AND PRIVACY, 1997, Oakland, CA, USA. Proceedings… Los Alamitos: IEEE Computer Society, 1997. p. 154-163.

MATHWORKS MATLAB, The Language for Technical Computing (2006). Available: <http://www.mathworks.com/products/matlab/>.

MEHROTRA, K.; MOHAN, C.; RANKA, S.; Elements of Artificial Neural Networks. MIT Press, 2000.

MUKHERJEE, Biswanath; HEBERLEIN, L. Todd; LEVITT, Karl N. Network Intrusion Detection. IEEE Network, v. 8, n. 3, p. 26-41, May/June 1994.

NAQVI, Syed; RIGUIDEL, Michel. Threat Model for Grid Security Services. In: EUROPEAN GRID CONFENCE (EGC), 2005, Amsterdam, The Netherlands. Proceedings… [S.l.: s.n.], 2005.

TALNAR, Vanish; BASU, Sujoy; KUMAR, Raj. An Environment for Enabling Interactive Grids. In: IEEE INTERNATIONAL SYMPOSIUM ON HIGH PERFORMANCE DISTRIBUTED COMPUTING, 12., 2003, Seattle, Washington, USA. Proceedings… Washington: IEEE Computer Society, 2003. p. 184-195.

TOLBA, M. et al. GIDA: Toward Enabling Grid ntrusion Detection Systems. In: IEEE/ACM NTERNATIONAL SYMPOSIUM ON CLUSTER COMPUTING AND THE GRID (CCGrid), 5., 2005, Cardiff, UK. Proceedings… [S.l.:s.n.], 2005.