# GIDA: Toward Enabling Grid Intrusion Detection Systems

M. F. Tolba M. S. Abdel-Wahab I. A. Taha A. M. Al-Shishtawy

alshishtawy@yahoo.com, Tel. +20105690130
Scientific Computing Department
Faculty of Computer and Information Sciences
Ain Shams University
Cairo, Egypt

## Abstract:

*Applying intrusion detection to the fast growing computational Grid environments improves the security which is considered to be the heart of this new field. Flexible cooperative distributed intrusion detection architecture is introduced that suits and benefits from the underlying Grid environment. The proposed architecture was tested using homogeneous distributed intrusion detection servers that use learning vector quantization neural network to detect the intrusion if occurred.*

## Keywords:

*Computational Grids, Grid Security Architecture, Intrusion Detection.*

## 1. Introduction

Security is a very important issue that must exist to enable the creation of Grid environments. The Grid needs intrusion detection as a second line of defense. It is very important because the current Grid security mechanisms can be penetrated and to provide protection from insiders. Intrusion detection systems are based on the assumption that normal use of the system is different from malicious use [5]. Due to the special characteristics [2] and requirements [1] of Computational Grids, detecting such difference in behavior imposed some new unique challenges that did not exist in traditional intrusion detection systems.

## 2. The Proposed Grid Intrusion Detection Architecture (GIDA)

GIDA was designed with the Grid characteristics in mind. GIDA has two main parts (Figure 1). The first is the Intrusion Detection Agent (IDA), that is responsible for gathering information. The second part is the Intrusion Detection Server (IDS), responsible for analyzing the gathered information and cooperating with other IDSs to detect intrusion.

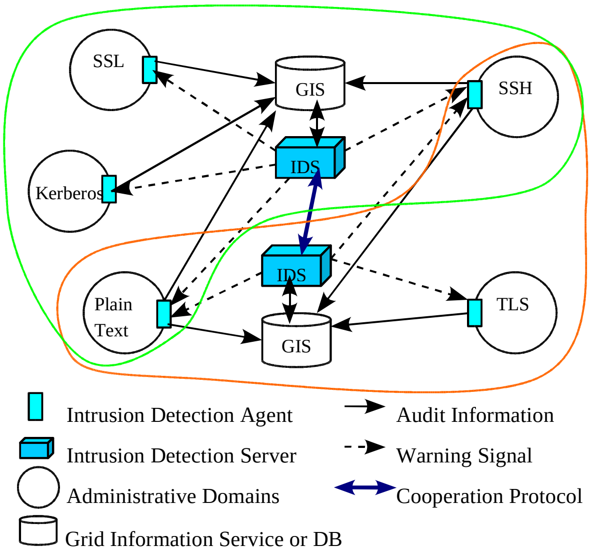

**Figure 1.** Proposed Grid intrusion detection architecture

The circles represent the administrative domains (resources) in a Grid environment. Each administrative domain will have an IDA to collect data and the IDA will register with one or more IDSs which will analyze the gathered data.

The IDAs will be designed for each class of resources to handle heterogeneity. The IDSs may use different techniques for data analysis. GIDA compatibility with the Grid is summarize in Table 1.

## 3. An Implementation of GIDA

We used two stages to test the proposed GIDA. The first stage simulates the IDA and the Grid environment. Most of the available Grid simulation toolkits are

**Table 1.** GIDA compatibility with computational Grid characteristic and requirements.

| **Characteristics / Requirements** | **GIDA Compatibility** |
|---|---|
| Heterogeneity | IDA deals with heterogeneity |
| Scalability | All components are distributed |
| Dynamicity or adaptability | Registration with multiple IDSs so if one fails others provide protection |
| No centralized control | Decision is made through cooperation between IDSs |
| Standard protocols | Build on top of GSI and Grid protocols |
| Nontrivial QoS | Different ID algorithms and trust relationships |

designed for resource management and scheduling problems. For this reason we developed a grid simulation toolkit based on GridSim [3] to satisfy our needs.

The simulation environment simulates users with different behaviors, resources with associated IDAs, and IDAs registration with IDSs. This allow us to perform the required experiments. Each experiment will generate a dataset consisting of one or more log file. Figure 2 shows the simulation environment with dummy IDSs that only generate log files reflecting the data they should analyze.

The next stage implements the IDS modules and test them with the data generated from the simulation stage (Figure 3). In this initial implementation we choose to use homogeneous IDSs for simplification. We believe that currently the best intrusion detection technique to use in this case is host-based anomaly intrusion detection [4].

The host in this case is the administrative domain with all its resources. The assigned IDA will gather information about the users interactions with this domain.

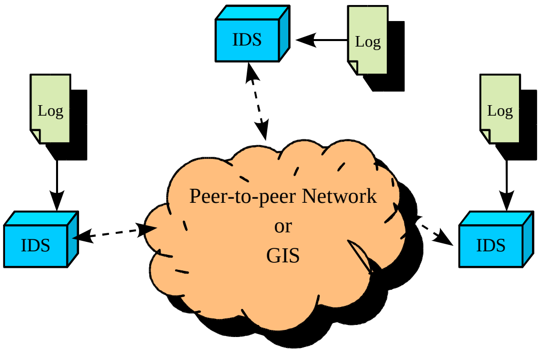

**Figure 3.** The implementation of IDSs based on the data generated from simulation.

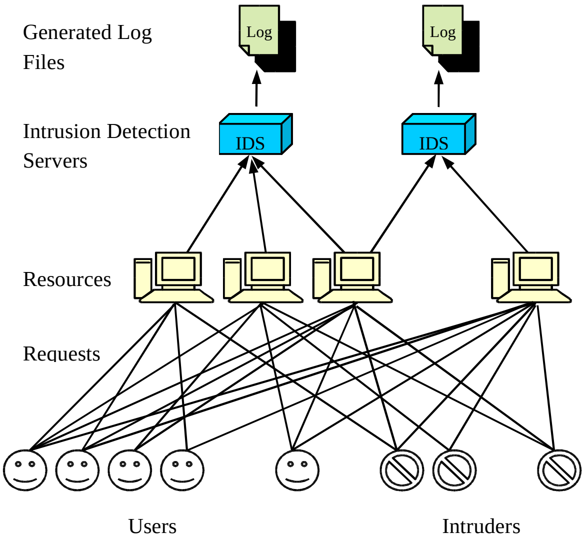

**Figure 2.** The simulated Grid and data gathering modules.

The anomaly detection is implemented using LVQ [6] neural network. The LVQ will try to learn the user behavior through interactions with different resources and then detect deviation from normal behavior. So an intruder in this case is a user whose current behavior deviates from the learned historical profile. The system takes advantage from the fact that each user in the Grid has a unique Global name. The decision module will analyze the LVQ result then, with information from the cooperation module, will decide wither a user is normal or intruder (Figure 4). The cooperation module helps in sharing the results. Each IDS analyze the user behavior in its scope and then shares these results with other IDSs in a way similar to P2P networks where the IDSs are the peers.

## 4. Testing of GIDA

The number of IDSs is an important issue that shows the scalability of the system and that it is possible to

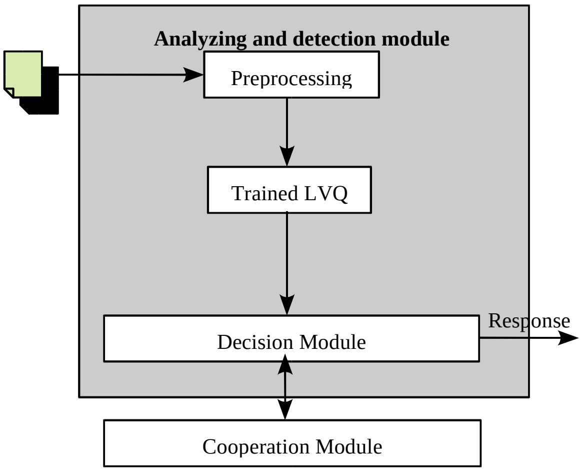

**Figure 4.** The Intrusion Detection Server.

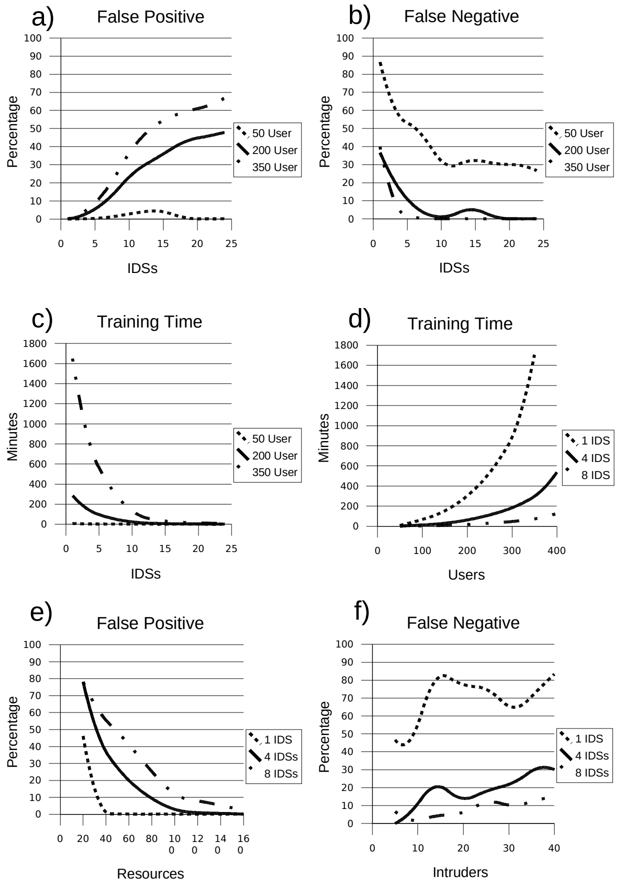

**Figure 5.** Some experimental results.

distribute the intrusion detection problem among multiple IDSs. Increasing the number of IDSs increased the percentage of false positive (Figure 5.a). This is because fewer information is available to each IDS about the user behavior. Meanwhile it decreased the percentage of false negative (Figure 5.b) because among the few user actions monitored at an IDS detecting deviation form them is easier. Increasing the number of IDSs has a great effect on reducing the training time (Figure 5.c).

On the other hand, increasing the number of users increased the training time (Figure 5.d). This shows that centralized systems with one IDS are not scalable as training time increased exponentially, multiple IDSs kept training time low.

Increasing resources reduced the false positive percentage (Figure 5.e). This is because users have wider variety of resources to choose from and this gives them better distinct behavior. Increasing the number of intruders has only slightly increased the percentage of the false negative (Figure 5.f). More detailed information can be found in [4].

## 5. Conclusions and future work

The proposed GIDA is an open and flexible architecture that addresses the special requirements of the Grid. The main issues affecting the system have been discussed to help in deciding the value of different parameters to increase the performance of the system in different Grid environments. The distribution of the intrusion detection problem among multiple IDSs made GIDA suitable for the Grid and improved performance compared with centralized systems. This work helps to understand the problem of intrusion detection in Grid environments and to build future systems.

The effect of trust relationships between different resource owners and the use of heterogeneous IDSs should be further investigated. Also these two issues will raise a question about their effects on different QoSs and how these QoSs can be selected and measured. With Heterogeneous IDSs and trust relationships more complex algorithms will be needed for the cooperation module that will need further investigations. The application of the Grid in real problems will help in building a knowledge base of attack signatures that will enable the use of misuse intrusion detection with the Grid.

## References

[1] I. Foster, Grid Today. Daily News and Information for the Global Grid Community. July 22, 2002: VOL. 1 NO. 6. http://news.tgc.com/msgget.jsp?mid=286185&xsl=story.xsl

[2] M. Baker, R. Buyya, and D. Laforenza, “Grids and Grid technologies for wide-area distributed computing”. Software Practice and Experience, 2002

[3] M. Murshed, R. Buyya, and D. Abramson, "GridSim: A Grid Simulation Toolkit for Resource Management and Scheduling in Large-Scale Grid Computing Environments". 17th IEEE International Symposium on Parallel and Distributed Processing (IPDPS 2002), April 15-19, 2002, Fort Lauderdale, FL, USA.

[4] M. Tolba, M. Abdel-Wahab, I. Taha, and A. Al-Shishtawy, "Distributed Intrusion Detection System for Computational Grids". Second International Conference on Intelligent Computing and Information Systems, March 2005.

[5] P. Anderson, "Computer Security Threat Monitoring and Surveillance". Technical report, James P. Anderson Company, Fort Washington, Pennsylvania, April 1980.

[6] T. Kohonen, "Learning Vector Quantization". In M. Arbib, editor, The Handbook of Brain Theory and Neural Networks. pages 537--540. MIT Press, 1995.