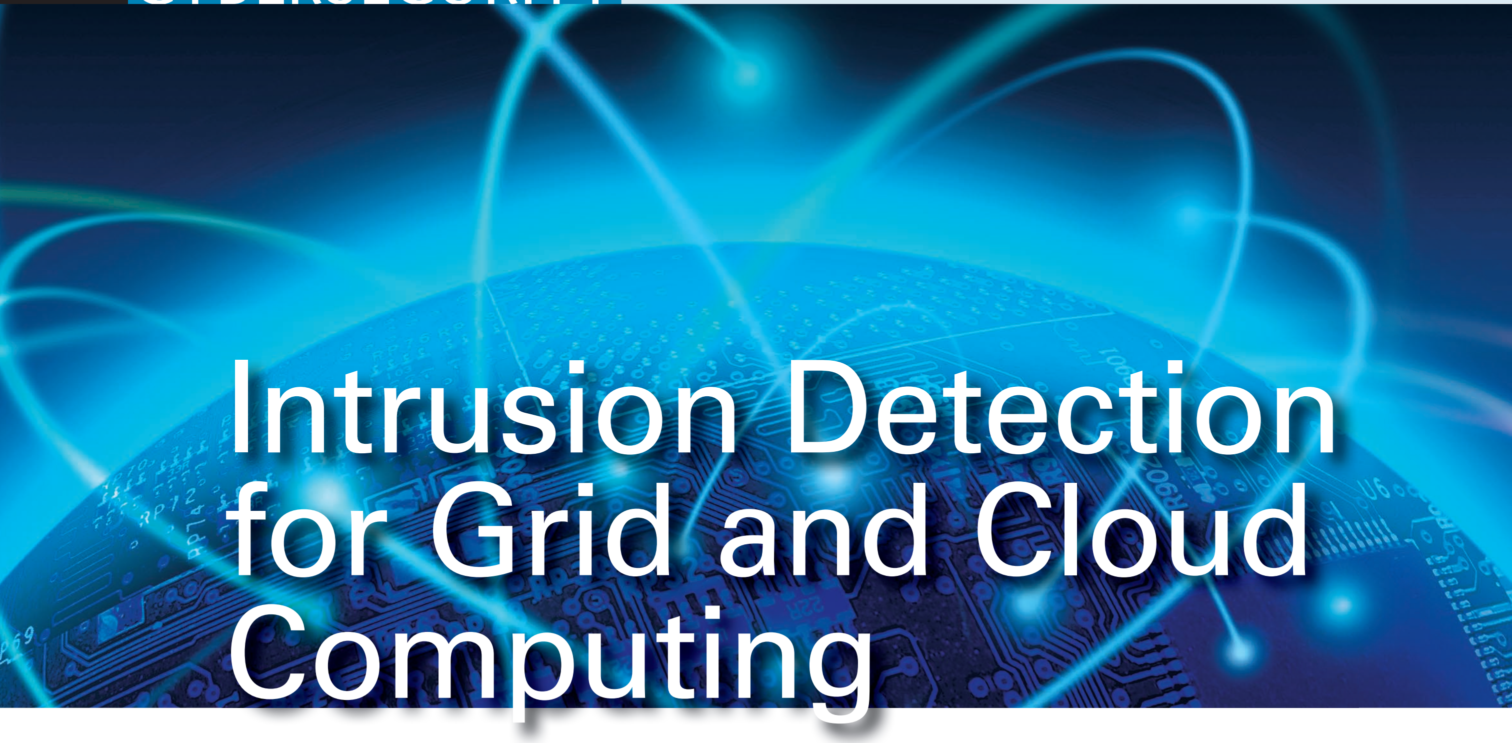

Intrusion Detection for Grid and Cloud Computing

Kleber Vieira, Alexandre Schulter, Carlos Becker Westphall, and Carla Merkle Westphall,
Federal University of Santa Catarina, Brazil

Providing security in a distributed system requires more than user authentication with passwords or digital certificates and confidentiality in data transmission. The Grid and Cloud Computing Intrusion Detection System integrates knowledge and behavior analysis to detect intrusions.

Because of their distributed nature, grid and cloud computing environments are easy targets for intruders looking for possible vulnerabilities to exploit. By impersonating legitimate users, the intruders can use a service's abundant resources maliciously.

To combat attackers, intrusion-detection systems (IDSs) can offer additional security measures for these environments by investigating configurations, logs, network traffic, and user actions to identify typical attack behavior.[1] However, an IDS must be distributed to work in a grid and cloud computing environment. It must monitor each node and, when an attack occurs, alert other nodes in the environment. This kind of communication requires compatibility between heterogeneous hosts, various communication mechanisms, and permission control over system maintenance and updates—typical features in grid and cloud environments.[2] Cloud middleware usually provides these features, so we propose an IDS service offered at the middleware layer (as opposed to the infrastructure or software layers).

An attack against a cloud computing system can be silent for a network-based IDS deployed in its environment, because node communication is usually encrypted. Attacks can also be invisible to host-based IDSs, because cloud-specific attacks don't necessarily leave traces in a node's operating system, where the host-based IDS resides. In this way, traditional IDSs can't appropriately identify suspicious activities in a grid and cloud environment[3] (see the "Related Work in Intrusion Detection" sidebar).

Here, we take a careful look at the cloud case in particular. We propose the Grid and Cloud Computing Intrusion Detection System (GCCIDS), which has an audit system designed to cover attacks that network- and host-based systems can't detect. GCCIDS integrates knowledge and behavior analysis to detect specific intrusions.

Published by the IEEE Computer Society

Related Work in Intrusion Detection

Here we present some of the relevant research on intrusion detection for grids, discussing in particular the techniques they apply and the source of the data they analyze.

Table A classifies related work according to the audit data source (host, network, or grid), the analysis technique (knowledge- or behavior-based), and if there was a proper evaluation. Fang-Yie Leu, Jia-Chun Lin, Ming-Chang Li, Chao-Tung Yang, and Po-Chi Shih's work,[1] along with Stuart Kenny and Brian Coghlan's[2] solutions, are based on analyzing data from a grid's network, although these approaches can't detect grid-specific attacks, because they don't capture any high-level data. Guofu Feng, Xiaoshe Dong, Weizhe Liu, Ying Chu, and Junyang Li integrate a host-based intrusion-detection system (IDS) into a grid environment, providing protection against typical operating system attacks, but not the ones that might target middleware vulnerabilities.[3]

Mohamed Tolba[4] and Alexandre Schulter[5] and their colleagues view a computational grid as one big host of resources, and the audit data is collected from the operating systems as in typical host-based IDSs. Their solutions focus on analyzing high-level information regarding grid usage by its users, and they apply behavior-based techniques in the analysis. In comparison, we conclude that the available solutions approach the problem in a different way, especially in regards to the threats we try to defend against by combining two distinct auditing techniques.

References

1. F-Y. Leu et al., "Integrating Grid with Intrusion Detection," *Proc. Int'l Conf. Advanced Information Networking and Applications* (AINA 05), vol. 1, IEEE CS Press, 2005, pp. 304–309.
2. S. Kenny and B. Coghlan, "Towards a Grid-Wide Intrusion Detection System," *Proc. European Grid Conf.* (EGC 05), Springer, 2005, pp. 275–284.
3. G. Feng et al., "GHIDS: Defending Computational Grids against Misusing of Shared Resource," *Proc. Asia-Pacific Conf. Services Computing* (APSCC 06), IEEE CS Press, 2006, pp. 526–533.
4. M. Tolba et al., "Distributed Intrusion Detection System for Computational Grids," *Proc. 2nd Int'l Conf. Intelligent Computing and Information Systems* (ICICIS 05), 2005.
5. A. Schulter et al., "Intrusion Detection for Computational Grids," *Proc. 2nd Int'l Conf. New Technologies, Mobility, and Security*, IEEE Press, 2008, pp. 1–5.

Table A. Features of related works concerning intrusion detection for grids.

Author	Host-based IDS	Network-based IDS	Data from a grid	Knowledge-based technique	Behavior-based technique	Validation
Tolba	Yes	No	Yes	No	Yes	Yes
Schulter	Yes	Yes	No	No	Yes	Yes
Choon	No	Yes	N/A	No	No	No
Kenny	No	Yes	No	Yes	No	Yes
Leu	No	Yes	No	Yes	No	Yes
Feng	Yes	No	No	Yes	No	Yes

Our Proposed Service

In our solution, each node identifies local events that could represent security violations and alerts the other nodes. Each individual IDS cooperatively participates in intrusion detection. Figure 1 depicts the sharing of information between the IDS service and the other elements participating in the architecture: the node, service, event auditor, and storage service.

The *node* contains the resources, which are accessed homogeneously through the middleware. The middleware sets the access-control policies and supports a service-oriented environment.

The *service* provides its functionality in the environment through the middleware, which facilitates communication.

The *event auditor* is the key piece in the system. It captures data from various sources, such as the log system, service, and node messages. The IDS service analyzes this data and applies detection techniques based on user behavior and knowledge of previous attacks. If it detects an intrusion, it uses the middleware's

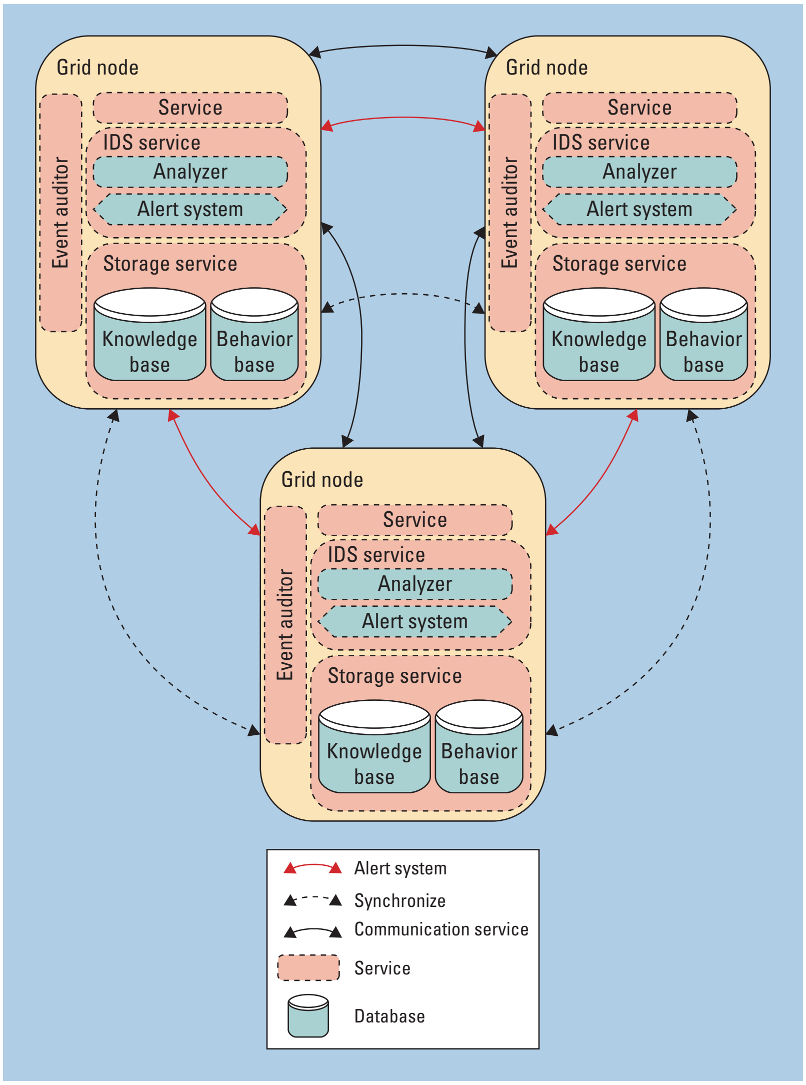

Figure 1. **The architecture of grid and cloud computing intrusion detection. Each node identifies local events that could represent security violations and sends an alert to the other nodes.**

communication mechanisms to send alerts to the other nodes. The middleware synchronizes the known-attacks and user-behavior databases.

The *storage service* holds the data that the IDS service must analyze. It's important for all nodes to have access to the same data, so the middleware must transparently create a virtualization of the homogeneous environment.

IDS Service

The IDS service increases a cloud's security level by applying two methods of intrusion detection. The behavior-based method dictates how to compare recent user actions to the usual behavior. The knowledge-based method detects known trails left by attacks or certain sequences of actions from a user who might represent an attack.

The audited data is sent to the IDS service core, which analyzes the behavior using artificial intelligence to detect deviations. The analyzer uses a profile history database to determine the distance between a typical user behavior and the suspect behavior and communicates this to the IDS service.

The rules analyzer receives audit packages and determines whether a rule in the database is being broken. It returns the result to the IDS service core. With these responses, the IDS calculates the probability that the action represents an attack and alerts the other nodes if the probability is sufficiently high.

Event Auditor

To detect an intrusion, we need audit data describing the environment's state and the messages being exchanged. The event auditor can monitor the data that the analyzers are accessing. The first component monitors message exchange between nodes. Although audit information about the communication between nodes is being captured, no network data is taken into account—only node information.

The second component monitors the middleware logging system. For each action occurring in a node, a log entry is created containing the action's type (such as error, alert, or warning), the event that generated it, and the message. With this kind of data, it's possible to identify an ongoing intrusion.

Behavior Analysis

Numerous methods exist for behavior-based intrusion detection, such as data mining, artificial neural networks, and artificial immunological systems. We use a feed-forward artificial neural network, because—in contrast to traditional methods—this type of network can quickly process information, has self-learning capabilities, and can tolerate small behavior deviations. These features help overcome some IDS limitations.[4]

Using this method, we need to recognize expected behavior (legitimate use) or a severe behavior deviation. Training plays a key role in the pattern recognition that feed-forward networks perform. The network must be correctly trained to efficiently detect intrusions. For a given intrusion sample set, the network learns to identify the intrusions using its retropropagation algorithm. However, we focus on identifying user behavioral patterns and deviations from such patterns. With this strategy, we can cover a wider range of unknown attacks.

Knowledge Analysis

Knowledge-based intrusion detection is the most often applied technique in the field because it results in a low false-alarm rate and high positive rates, although it can't detect unknown attack patterns. It uses rules (also called signatures) and monitors a stream of events to find malicious characteristics.

Using an expert system, we can describe a malicious behavior with a rule. One advantage of using this kind of intrusion detection is that we can add new rules without modifying existing ones.

In contrast, behavior-based analysis is performed on learned behavior that can't be modified without losing the previous learning. Generating rules is the key element in this technique—it helps the expert system recognize newly discovered attacks. Creating a rule consists of defining the set of conditions that represent the attack.

Increasing Attack Coverage

The two intrusion detection techniques are distinct. The knowledge-based intrusion detection is characterized by a high hit rate of known attacks, but it's deficient in detecting new attacks. We therefore complemented it with the behavior-based technique, which can discover deviations from acceptable use and thus help identify privilege abuse.

The volume of data in a cloud computing environment can be high, so administrators don't observe each user's actions—they observe only alerts from the IDS.

Results

We developed a prototype to evaluate the proposed architecture using Grid-M, a middleware of our research group developed at the Federal University of Santa Catarina.[5]

We created data tables to perform the experiments with audit elements coming from both the log system and from data captured during node communications. We prepared three types of simulation data to test.

First, we created data representing legitimate action by executing a set of known services simulating a regular behavior.

Then, we created data representing behavior anomalies. To represent anomalous sequences of actions, we altered the services and their usage frequency. For example, for a teaching department that posts grades electronically, if two out of every 100 grades are typically corrected later because of a mistake, then an anomalous behavior would be correcting 10 consecutive grades. This action would deserve special attention to determine whether it constituted an abuse of privileges.

Finally, we created data representing policy violation. This was prepared with a set of audit packages containing a series of elements violating base rules.

Evaluating the Event Auditor

The event auditor captures all requests received by a node and the corresponding responses, which is fundamental for behavior analysis.

For each action a node performs, a log entry is generated to register the methods and parameters invoked during the action.

In the experiments with the behavior-based IDS, we considered using audit data from both a log and a communication system. Unfortunately, data from a log system—with the exception of the message element—has a limited set of values with little variation. This made it difficult to find attack patterns, so we opted to explore communication elements to evaluate this technique.

We evaluated the behavior-based technique using artificial intelligence enabled by a feed-forward neural network.[6] In the simulation environment, we monitored five intruders and five legitimate users.

We initiated the neural-network training with a data set representing 10 days of usage simulation. Using this data resulted in a high number

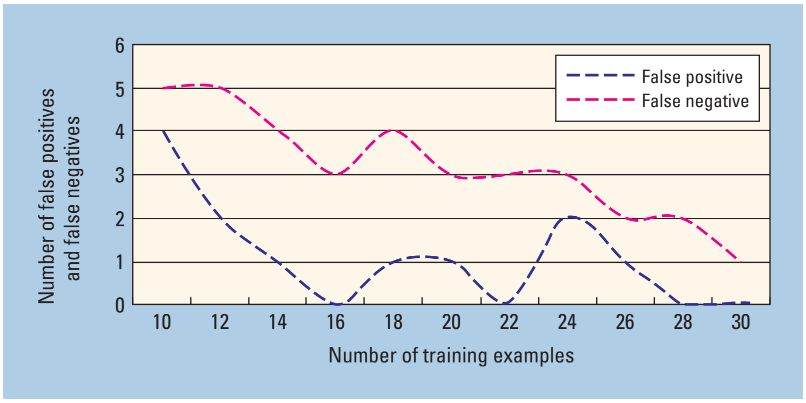

Figure 2. The behavior score results. The algorithm had the lowest number of false positives for input periods with 28–30 days.

of false negatives and a high level of uncertainty. Increasing the sample period for the learning phase improved the results.

Evaluating the Behavior-Based System

To measure IDS efficiency,[1] we considered accuracy in terms of the system's ability to detect attacks and avoid false alarms. A system is imperfect if it accuses a legitimate action of being malicious. So, we measured accuracy using the number of false positives (legitimate actions marked as attacks) and false negatives (the absence of an alert when an attack has occurred).

The performance test we designed also evaluated the analysis technique's cost. We performed a load test where the program analyzed 1 to 100,000 actions. The simulation involving 100,000 actions is hypothetical. It surpasses the usual data volume and served as a base for understanding system behavior in an overloading condition. An action took approximately 0.000271 seconds to be processed with our setup.

The training time for an input of 30 days of sample behavior took 1.993 seconds. However, the training was sporadic—we had to plan updates to the behavior profile database according to a routine in the execution environment (since a user's behavior tends to change with time). This helped us identify a convenient period of days for determining the profile of a legitimate user. Artificial neural networks aren't deterministic, so the number of false positives and false negatives didn't represent a linear decreasing progression.

Figure 2 shows the results. The neural network tended to avoid identifying legitimate actions as attacks—there were always more false negatives than false positives when using the same quantity of input data.

No false alarms occurred when we started the training with 16 days of simulation, although the uncertainty level was still high, with several outputs near zero. With input periods of 28, 29, and 30 days, the algorithm showed a low number of false positives, but after several repetitions, the quantity of false positives varied, again representing the nondeterministic nature of neural networks.

Evaluating the Knowledge-Based System

In contrast to the behavior-based system, we used audit data from both a log system and the communication system to evaluate the knowledge-based system. We created a series of rules to illustrate security policies that the IDS should monitor.

We collected audit data referring to a route-discovery service, service discovery, and service request and response. The series of policies we created tested the system's performance, although our scope didn't include discovering new kinds of attacks or creating an attack database. Our goal was to evaluate our solution's functionality and the prototype's performance.

The rule below characterizes an attack in any message related to the storage service. The functions of the rule are as follows:

1. At start-up, the rules stored in an XML file are loaded into a data structure.
2. The auditor starts to capture data from the log and communication systems.
3. The data is preprocessed to create a data structure dividing log data from communication data to provide easy access to each element.
4. The corresponding policy for the audit package is verified.
5. An alert is generated if an attack or violation occurred.

We performed a load test for this algorithm simulating the analysis of 10 to 1,000,000 rules for an action. We verified the textual or numerical field in comparison to the rules. The analyzer performed two primary functions: it searched for improper content, and it compared numerical intervals. Comparing 100,000 rules for an action consumed 0.361 seconds; comparing a million rules consumed 2.7 seconds. This suggests that real-time analysis is possible up until a certain limit in the number of rules.

In testing our prototype, we learned that it has a low processing cost while still providing a satisfactory performance for real-time implementation. Sending data to other nodes for processing didn't seem necessary.[7] The individual analysis performed in each node reduces the complexity and the volume of data in comparison to previous solutions, where the audit data is concentrated in single points.

In the future, we'll implement our IDS, helping to improve green (energy-efficient), white (using wireless networks), and cognitive (using cognitive networks) cloud computing environments. We also intend to research and improve cloud computing security.

References

1. H. Debar, M. Dacier, and A. Wespi, "Towards a Taxonomy of Intrusion Detection Systems," *Int'l J. Computer and Telecommunications Networking*, vol. 31, no. 9, 1999, pp. 805–822.
2. I. Foster et al., "A Security Architecture for Computational Grids," *Proc. 5th ACM Conf. Computer and Communications Security*, ACM Press, 1998, pp. 83–92.
3. S. Axelsson, *Research in Intrusion-Detection Systems: A Survey*, tech. report TR-98-17, Dept. Computer Eng., Chalmers Univ. of Technology, 1999.
4. A. Schulter et al., "Intrusion Detection for Computational Grids," *Proc. 2nd Int'l Conf. New Technologies, Mobility, and Security*, IEEE Press, 2008, pp. 1–5.
5. H. Franke et al., "Grid-M: Middleware to Integrate Mobile Devices, Sensors and Grid Computing," *Proc. 3rd Int'l Conf. Wireless and Mobile Comm.* (ICWMC 07), IEEE CS Press, 2007, p. 19.
6. N.B. Idris and B. Shanmugam, "Artificial Intelligence Techniques Applied to Intrusion Detection," *Proc. 2005 IEEE India Conf. (Indicon) 2005 Conf.*, IEEE Press, 2005, pp. 52–55.
7. P.F. da Silva and C.B. Westphall, "Improvements in the Model for Interoperability of Intrusion Detection Responses Compatible with the IDWG Model," *Int'l J. Network Management*, vol. 17, no. 4, 2007, pp. 287–294.

***Kleber Vieira** is a team leader for a software development company in Brazil and is a member of the Networks and Management Laboratory at the Federal University of Santa Catarina, Brazil. His research interests include information systems, software engineering, distributed systems, and security. Vieira received his MSc in computer science from the Federal University of Santa Cataria. Contact him at kleber@inf.ufsc.br.*

***Alexandre Schulter** is an IT analyst for a Brazilian government company. Previously, he was a researcher and software developer at several laboratories in the Technological Centre at the Federal University of Santa Catarina, Brazil. His research interests include information systems, component-based systems, software engineering, distributed systems, and security. Schulter received his MSc in computer science from the Federal University of Santa Cataria. Contact him at schulter@inf.ufsc.br.*

***Carlos Becker Westphall** is a full professor in the Department of Informatics and Statistics at the Federal University of Santa Catarina, Brazil, where he is the leader of the Networks and Management Laboratory. His research interests include network management, security, and grid and cloud computing. Westphall received his DSc in computer science from the Paul Sabatier University, France. Contact him at westphal@inf.ufsc.br.*

***Carla Merkle Westphall** is a professor in the Department of Informatics and Statistics at the Federal University of Santa Catarina, Brazil. Her research interests include distributed security, identity management, and grid and cloud security. Westphall received her PhD in electrical engineering from the Federal University of Santa Cataria. Contact her at carlamw@inf.ufsc.br.*