Intrusion Detection for Grid and Cloud Computing

Vanoir G. Zacaron
Tiago Souza Cândido
Tiago Pereira Coelho
Vicente Silveira Inácio

ADMINISTRATION AND MANAGEMENT OF COMPUTER NETWORKS - INE5619
Federal University of Santa Catarina

NOVEMBER 6TH, BRAZIL
Outline

1. ABSTRACT
2. INTRODUCTION
3. RELATED WORKS
4. OUR PROPOSED SERVICE
   4.1 ARCHITECTURE
Outline

5. GRID NODE
   5.1. IDS Service
   5.2. Event Auditor
   5.3. Behavior Analysis
   5.4. Knowledge Analysis
   5.5. Increasing Attack Coverage

6. RESULTS

7. CONCLUSIONS

8. REFERENCES

NOVEMBER 6TH, BRAZIL
1. ABSTRACT

This presentation describes:

- Grid and Cloud Computing Intrusion Detection System using knowledge and behavior analysis
2. INTRODUCTION

The intent of this presentation is to:

• Provide insight about new methods to provide security in grid and cloud computing environments by identifying typical malicious behavior.

• Propose a model of Grid and Cloud Computing Intrusion Detection System (GCCIDS), which covers a wider range of attacks than previous systems.
3. RELATED WORKS

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

<table>
<thead>
<tr>
<th>Author</th>
<th>Host-based IDS</th>
<th>Network-based IDS</th>
<th>Data from a grid</th>
<th>Knowledge-based technique</th>
<th>Behavior-based technique</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tolba</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Schulten</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Choon</td>
<td>No</td>
<td>Yes</td>
<td>N/A</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Kenny</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Leu</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Feng</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Leu's and Kenny's solutions are **based on analyzing data from a grid’s network** therefore it can’t detect grid-specific attacks, because they don’t capture any high-level data.
Feng's solution integrates a host-based IDS into a grid environment, providing protection against typical operating system attacks, but not the ones that might target middleware vulnerabilities.
3. RELATED WORKS

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

<table>
<thead>
<tr>
<th>Author</th>
<th>Host-based IDS</th>
<th>Network-based IDS</th>
<th>Data from a grid</th>
<th>Knowledge-based technique</th>
<th>Behavior-based technique</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tolba</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Schulter</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Choon</td>
<td>No</td>
<td>Yes</td>
<td>N/A</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Kenny</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Leu</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Feng</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Tolbas' and Schulter's solution collect audit data from the OS as in typical host-based IDSs and focus on analyzing high-level information regarding grid usage by its users and applies behavior-based techniques in the analysis.
An IDS service offered at the middleware layer

Looking to solve the problems found with former solutions such as network-based IDSs (communication is usually encrypted) and host-based IDSs (cloud-specific attacks don't necessarily leave traces in a node's OS).
4.1 ARCHITECTURE

- **Node**: Contains the resources (accessed through the middleware), each node identifies local events and alerts the other nodes.
- **Service**: The service itself, functioning through the middleware.
- **Event Auditor**: Captures data from various sources, such as the log system, service, and node messages.
- **IDS**: Analyzes this data and applies the detection techniques.
- **Storage**: Holds the data that the IDS analyzes.
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.
5.1. IDS Service

Intrusion-Detection Systems offer additional security measures for Grid and Cloud environments by investigating configurations, logs, network traffic, and user actions to identify typical attack behavior.
Intrusion-Detection Systems increases a cloud's security level by applying two methods of intrusion detection:

- **Behavior-based**: Dictates how to compare recent user actions to the usual behavior.
- **Knowledge-based**: Detects known trails left by attacks or certain sequences of actions from a user who might represent an attack.
5.1. IDS Service

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.
5.1. 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.
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.
5.2 Event Auditor

The first component (Event Auditor) 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.
5.2 Event Auditor

The second component (Analyzer) 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.
5.3. Behavior Analysis

Numerous methods exist for behavior-based intrusion detection, such as data mining, artificial neural networks, and artificial immunological systems.
5.3. Behavior Analysis

Feed-forward artificial neural network was used, 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.
5.3. Behavior Analysis

By using this method, a recognition of expected behavior (legitimate use) or a severe behavior deviation is needed. Training plays a key role in the pattern recognition that feed-forward networks perform. The network must be correctly trained to efficiently detect intrusions.
5.3. Behavior Analysis

For a given intrusion sample set, the network learns to identify the intrusions using its retropropagation algorithm. The focus are on identifying user behavioral patterns and deviations from such patterns. With this strategy, a wider range of unknown attacks is covered.
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. It uses rules (signatures), creating a rule consists of defining a set of conditions that represents an attack.
5.5. Increasing Attack Coverage

The knowledge-based intrusion detection is deficient in detecting new attacks. Behavior-based technique can discover deviations from acceptable use and thus help identify privilege abuse.
6. RESULTS

Was developed a prototype to evaluate the proposed architecture using Grid-M. Was created data tables to perform the experiments with audit elements coming from both the log system and from data captured during node communications.
6. RESULTS

Evaluating the Event Auditor

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.
6. RESULTS

Evaluating the Event Auditor

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.
6. RESULTS

Evaluating the Behavior-Based System

• To measure IDS efficiency, we considered accuracy in terms of the system’s ability to detect attacks and avoid false alarms.

• 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).
6. RESULTS

Evaluating the Behavior-Based System

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

Evaluating the Behavior-Based System

- Identify a convenient period of days for determining the profile of a legitimate user.
- The neural network tended to avoid identifying legitimate actions as attacks.
- The quantity of false positives varied, again representing the nondeterministic nature of neural networks.
6. RESULTS

Evaluating the Knowledge-Based System
- We created a series of rules to illustrate security policies that the IDS should monitor.
- Our goal was to evaluate our solution’s functionality and the prototype’s performance.
6. RESULTS

Evaluating the Knowledge-Based System

The analyzer performed two primary functions: it searched for improper content, and it compared numerical intervals. Comparing 10,000 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.

NOVEMBER 6TH, BRAZIL
7. CONCLUSION

-Low processing cost while still providing a satisfactory performance for real-time implementation. Sending data to other nodes for processing didn’t seem necessary.

-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.
7. CONCLUSION

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.
References indicated in this presentation: