An Efficient Alert Correlation Engine for Intrusion Detection

Sanoop Mallissery, Shahana Sathar, Vijay Kumar Chitturi
TIFAC Core Cyber Security Department
Amrita School of Engineering, Ettimadai
Coimbatore, India
{sanoopmallissery, shahanasathar, vijaykumar515}@gmail.com

Abstract- Alert Correlation is a process that analyses the alerts produced by one or more Intrusion Detection Sensors and provides a more succinct and high-level view of occurring or attempted intrusions. The objective of this study is to implement an Alert Correlation technique by using an improved cluster algorithm for an optimized correlation in the process of attack detection. Here, using the Quantum-Behaved Particle Swarm Optimization to optimize the cluster parameters and by using these optimized parameters (weights and threshold) to improve the performance of cluster algorithm. In our proposed system, before alert clustering we have to perform normalization, pre-processing, and alert fusion. So it will give a more accurate alert data set. Then the optimized parameters are given to the cluster algorithm. After this, we can use the concepts of Pre-requisites and consequences to correlate alerts in the middle of the attack chain, even if the start of the chain is missed.

Keywords- Alert correlation, QPSO, Alert management

I. INTRODUCTION

Recently, networks have evolved into a ubiquitous infrastructure. High-speed backbones and local area networks provide the end user with bandwidths that are orders of magnitude larger than those available a few years ago. In addition, wireless technology is bringing connectivity to a number of devices, from laptops to cell phones and PDAs, creating a complex, highly dynamic network of systems. Most notably, the Internet has become a mission-critical infrastructure for governments, companies, institutions, and millions of everyday users. Because of this increased reliance on networked computers, security has become a primary concern.

To alleviate some of the problems of intrusion detection systems, alert correlation systems have been proposed. Correlation systems collect the alerts from a number of sensors and process these alerts in order to generate a high-level view of the current security status of the network. The main goal of a correlation system is to reduce the number of alerts a system administrator has to manually process. The correlation system achieves this by identifying and suppressing false alerts, grouping alerts that refer to the same incident together, and prioritizing the alerts.

II. RELATED WORK

Most of the previous works on alert correlation has mostly focused on one of three types of correlation techniques: multi-step, fusion-based, and filter-based. Multi-step correlation relies on the fact that complex attacks are usually executed in several phases or steps, where the first steps prepare for the attacks executed in the later steps. Therefore, the multi-step correlation approach tries to link alerts that are part of different steps of the same complex attack scenario. Fusion-based correlation looks for alerts that are similar in some way. This approach is usually based on a function that calculates the similarity between any two pairs of alerts. If this similarity score exceeds some threshold, the alerts are correlated. Filter-based approaches either identify alerts that are irrelevant or assign a priority to each alert. For instance, an alert could be classified as irrelevant if it represents an attack against a non-existent service. Priorities are usually assigned to alerts depending on how important the attacked assets are.

III. CORRELATION COMPONENTS

Alert correlation is a multi-component process that receives as input a stream of alerts from multiple intrusion detection systems. In each component of the process, alerts are merged into high-level intrusion reports or tagged as irrelevant if they do not represent successful attacks. Finally, the alerts are prioritized according to the site's security policy, and eventually the results are reported.

A. The Correlation Process

The main objective of the correlation process is to produce a succinct overview of the security-related
activity on the network. This process consists of a collection of components that transform intrusion detection sensor alerts into intrusion reports. Because alerts can refer to different kinds of attacks at different levels of granularity, the correlation process cannot treat all alerts equally. Instead, it is necessary to provide a set of components that focus on different aspects of the overall correlation task.

1) Alert Base Management: The alerts from different sensors are collected by an alert base management. It contains all the alerts from each and every sensor connected to our correlation engine. Each sensor alerts represented is of different format, depends on which sensor drops the corresponding alert.

2) Alert Normalization: The goal of the alert normalization component is to translate all attributes of each sensor alert into a common format [3]. This translation requires that the syntax and semantics of a sensor alert are recognized. That is transforms alerts into a standardized format (different sensors use different formats). So we will use the IDMEF (Intrusion Detection Message Exchange format). For this purpose use an adapter module to act as an interface with sensors.

3) Alert Database: We are using a database to store all of these normalized alerts. From this database we are fetching the alerts for our next step.

4) Alert Pre-Processing: Normalized alerts have a standardized name and are in a format that is understood by the other components of the correlation process [3]. However, additional pre-processing may be required, since some sensors omit fields that are necessary for other components of the correlation process (i.e., time, source, and target). The goal of the pre-processing component is to supply, as accurately as possible, missing alert attributes that are important for later correlation components.

5) Alert Fusion: The goal of the alert fusion component is to combine alerts that represent the independent detection of the same attack occurrence by different intrusion detection systems [3]. The decision to fuse two alerts is based on the temporal difference between these alerts and the information that they contain. If no match is found after searching through the whole queue, the alert is inserted into the queue, to be considered for matching with future alerts. If duplicates are there, then it is removed.

6) Alert Cluster: In this session we converting alerts into different cluster, based on

a) Computing the similarity value
   - Alert type comparison
   - IP address comparison
   - Port Comparison
   - Protocol Comparison
   - Time Comparison

b) Using QPSO to optimize cluster parameters
   The Quantum-behaved Particle Swarm Optimization algorithm [1] is a new Particle Swarm Optimization (PSO) algorithm model that Sun and others presented from the standpoint of Quantum Mechanics. In the algorithm, the states of particles in quantum space no longer are denoted to the position vector and velocity vector, but are described by wave functions. And the position of mobile particle X is denoted by probability density function, but is not limited to the established track function. So the algorithm can search the global optimal solution in the entire feasible solution space.
   We are using this QPSO to optimize the cluster parameters. Then use the optimized parameters (weights and threshold) to improve the performance of cluster algorithm.

c) Cluster Algorithm
   Alert cluster is to group alerts so that alerts in a given class are similar to each other and dissimilar to each of other classes. And alerts in the same class are more likely
Alert cluster is to group alerts so that alerts in a given class are similar to each other and dissimilar to each of other classes. And alerts in the same class are more likely to belong to the same scenario. Hence we can use the optimized parameters (weights and threshold) to a cluster algorithm.

7) Alert Correlation: By Using the concept of prerequisites and consequences [2] its possible to correlate the alerts in the middle of the attack chain, even if start of the chain is missed. That is by using this concept we can correlate the second and third attack even if the pre-conditions of the second alert were not met.

8) Impact Analysis: This determines the impact of the detected attacks on the operation of the network being monitored and on the assets that are targeted by the malicious activity [3]. The impact analysis process combines the alerts from the previous correlation components with data from an asset database. The asset database stores information about installed network services, dependencies among these services, and their importance to the overall operation of a network installation. An example of a dependency between two services is a mail server that requires an operational domain name server (DNS) to work properly or the NFS server that requires the RPC services.

9) Alert Prioritization: Priorities are important to classify alerts and quickly discard information that is irrelevant or of less importance to a particular site. The goal of the alert prioritization component is to assign appropriate priorities to alerts. This component has to take into account the security policy and the security requirements of the site where the correlation system is deployed. Therefore, there is no absolute priority for an attack [3].

The alert prioritization component can use the information from the impact analysis and the asset database to determine the importance of network services to the overall operation of the network. For each network resource, the asset database contains values that characterize its secrecy, integrity, and availability properties.

IV. PROPOSED CRITERIA FOR ALERT CORRELATION PROCESS

Mainly Four important intrusion alert correlation techniques has been reviewed and analyzed. To objective of this study is to analyze the current alert correlation technique and identify the significant criteria within each technique which can improve the IDS problem.

The main problem that normally happened in IDS is the capability to do alert reduction, alert clustering, multi-step attack, reduce false alert, detect known attack, and detect unknown attack. Based on the analysis, all of the techniques have the same capability to reduce and cluster the alert, and detect known attack.

The analysis also shown that, most of the researchers identified that all of these techniques are incapable to reduce false alert. This has given an implication that there is a room for improvement in detecting known and unknown attack, and multi-step attack as these capability criteria shall overcome large number of false alert problem. Based on the study, we proposed an improved solution for alert correlation.

V. CONCLUSION

In this work, we have reviewed and analyzed the existing alert correlation technique to overcome the IDS’s problems discussed. From the analysis we are propose an improved solution for alert correlation technique based on six capability criteria identified which are capability to do alert reduction, capability to do alert clustering, capability to identify multi-step attack, capability to reduce false alert, capability to detect known attack and capability to detect unknown attack.

REFERENCES


