A Comprehensive Approach to Intrusion Detection Alert Correlation

***Nalini.L**
Lecturer, Dept of Computer Science,
BMS College for Women,
Basavanagudi, Bangalore-04

Alert correlation is a process that analyzes the alerts produced by one or more intrusion detection systems and provides a more succinct and high-level view of occurring or attempted intrusions. Even though the correlation process is often presented as a single step, the analysis is actually carried out by a number of components, each of which has a specific goal. Unfortunately, most approaches to correlation concentrate on just a few components of the process, providing formalisms and techniques that address only specific correlation issues. This paper presents a general correlation model that includes a comprehensive set of components and a framework based on this model. A tool using the framework has been applied to a number of well-known intrusion detection data sets to identify how each component contributes to the overall goals of correlation. The results of these experiments show that the correlation components are effective in achieving alert reduction and abstraction. They also show that the effectiveness of a component depends heavily on the nature of the data set analyzed.

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.

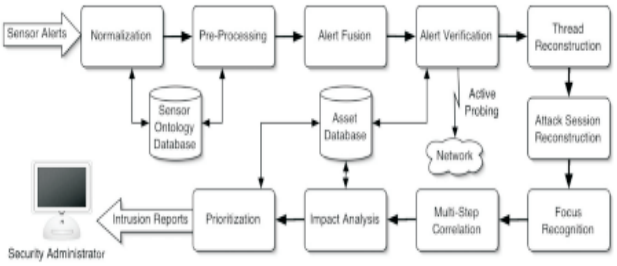
The surveillance and security monitoring of the network infrastructure is mostly performed using Intrusion Detection Systems (IDSs). These systems analyze information about the activities performed in computer systems and networks, looking for evidence of malicious behavior. Attacks against a system manifest themselves in terms of events, which can be of differing nature and level of granularity. For example, events may be represented by network packets, operating system calls, audit records produced by operating system auditing facilities, or log messages produced by applications. The goal of intrusion detection is to analyze one or more event streams and identify manifestations of attacks. When an attack is detected, an alert that describes the type of the attack and the entities involved (e.g., hosts, processes, users) is produced.

Event streams are used by intrusion detection systems in two different ways, according to two different paradigms: anomaly detection and misuse detection. In anomaly detection systems, historical data about a system's activity and/or specifications of the intended behavior of users

and applications are used to build a profile of the "normal" operation of the system being monitored. The intrusion detection system then tries to identify patterns of activity that deviate from the defined profile. Misuse detection systems take a complementary approach. These systems are equipped with a number of attack descriptions (or "signatures") that are matched against the stream of audit data looking for evidence that one of the modeled attacks is occurring.

Misuse and anomaly detection both have advantages and disadvantages. Misuse detection systems can perform focused analysis of the audit data, and they usually produce only a few false positives, which are erroneous detections. However, these systems can detect only those attacks that have been modeled. Anomaly detection systems have the advantage of being able to detect abnormal behavior, which may reveal previously unknown attacks. This advantage is paid for in terms of a large number of false positives. Another disadvantage of anomaly detection systems is the difficulty of training a system with respect to a very dynamic environment.

The intrusion detection community has developed a number of different intrusion detection systems that perform intrusion detection in particular domains (e.g., hosts or networks), in specific environments (e.g., Windows NT or Solaris), and at different levels of abstraction (e.g., kernel level tools or application-level tools). As more IDSs are developed, network security administrators are faced with the task of analyzing an increasing number of alerts resulting from the analysis of different event streams. In addition, IDSs are far from perfect and may produce both false positives and non-relevant positives. Non-relevant positives are alerts that correctly identify an attack, but



the attack fails to meet its objective. For instance, the attack may be exercising vulnerability in a service that is not provided by the victim host. As an example, consider a "Code Red" worm that attacks a Linux Apache server. Although an actual attack is seen on the network, this attack has to fail because Apache is not vulnerable to the exploit utilized by the worm. Clearly, there is a need for tools and techniques that allow the administrator to aggregate and combine the outputs of multiple IDSs, filter out spurious or incorrect alerts (such as "Code Red" attacks against Apache installations), and provide a succinct, high-level view of the security state of the protected network.

To address this issue, researchers and vendors have proposed alert correlation, an analysis process that takes the alerts produced by intrusion detection systems and produces compact reports on the security status of the network under surveillance. Although a number of correlation approaches have been suggested, there is no consensus on what this process is or how it should be implemented and evaluated. In particular, existing correlation approaches operate on only a few aspects of the correlation process, such as the fusion of alerts that are generated by different intrusion detection systems in response to a single attack, or the identification of multistep attacks that represent

a sequence of actions performed by the same attacker. In addition, correlation tools that do cover multiple aspects of the correlation process are evaluated "as a whole," without an assessment of the effectiveness of each component of the analysis process. As a result, it is not clear if and how the different parts of the correlation process contribute to the overall goals of correlation.

This paper presents a comprehensive correlation approach and an analysis of its components. The correlation approach has been designed to be as complete as possible and to include all the aspects of correlation discussed in previous research efforts. We built a framework that implements the correlation process and used a tool based on the framework to analyze a number of data sets. Instead of just focusing on the overall effectiveness of correlation, this paper focuses on each aspect of the process separately, to provide insights into how each component of the correlation process contributes to alert reduction and abstraction and what properties of the data being analyzed make one component more effective than another. The results show that the effectiveness of correlation is highly dependent on the nature of the data sets. That is, different parts of the process contribute to correlation in different ways, depending on the nature of the data being analyzed.

An extensive explanation of this paper will be structured as each section providing detailed information about overall architecture of the correlation process, the data sets that will be analyzed, a detailed description of the components of the correlation process and the results of applying each component to the sample data sets, related work on alert correlation and finally conclusions and outlines future work.