Journal of Information Systems and Communication
ISSN: 0976-8742 & E-ISSN: 0976-8750, Volume 3, Issue 1, 2012, pp.-30-34
Available online at http://www.bioinfo.in/contents.php?id=45

USE OF EVENT CORRELATION ANALYSIS TO REDUCE FALSE POSITION ALARM RATE IN NETWORK INTRUSION DETECTION SYSTEM

ROHINI BHOSALE*, NARESH D. HARALE AND ASHOK CHALAK

Department of Computer Engineering, MGMCET, Navi Mumbai, India.
*Corresponding Author Email: bhosalerohini123@gmail.com

Received: January 12, 2012; Accepted: February 15, 2012

Abstract- As the network based computer system plays an important role in modern society they have become target of our enemies and criminals. Therefore we need to find the best possible ways to protect our IT System. Different methods and algorithms are developed and proposed in recent years to improve intrusion detection systems. The most important issue in current systems is False Positive alarm rate. This is because current systems are poor at detecting novel anomaly attacks. These kinds of attacks refer to any action that significantly deviates from the normal behaviour which is considered intrusion. Many NIDSs are signature based which consider only one device log, and conclude whether intrusion happened or not and internet attacks are increasing exponentially and there have been various attacks methods, consequently. False Positive alarm affects the effectiveness of NIDS and increase load on network administrator which can frustrate admin. To minimise this False Positives, we can use Data mining using Event Correlation Technique (ECT) for Network Intrusion Detection such that by correlating events at different component of network security NIDS can identify whether actually intrusion occurred or not. In this paper we aim to discuss our proposed system in that we are using Event correlation analysis to reduce False Positive alarm rate. This paper also clarifies important issues concerning Network Intrusion Detection System, Data mining and Event Correlation and proposed system that increase the effectiveness of NIDS by using event log analysis and correlation.

Keywords- Network Intrusion Detection System (NIDS), Data mining, Event Correlation, False Positive Alarm

Citation: Rohini Bhosale, Naresh D. Harale and Ashok Chalak. (2012) Use of Event Correlation Analysis to Reduce False Position Alarm Rate in Network Instruction Detection System. Journal of Information Systems and Communication, ISSN: 0976-8742 & E-ISSN: 0976-8750, Volume 3, Issue 1, pp.-30-34.

Introduction

In many industries computer network play an important role for information exchange, example tender quotations or for sending confidential information computer networks re most preferred. And so they have become the targets of our enemies and criminals. Therefore, we need to find the best ways possible to protect our systems. When Intrusion occurs security of system compromised. An intrusion can thus be defined as “any set of actions that attempt to compromise the integrity, confidentiality, or availability of a resource”. Intrusion prevention techniques, such as user authentication (e.g., using passwords or biometrics), avoiding programming errors, and information protection (e.g., encryption) have been used to protect computer systems as a first line of defense. Intrusion prevention alone is not sufficient because as systems become ever more complex, there are always exploitable weaknesses in the systems due to design and programming errors, or various “socially engineered” penetration techniques [1]. Also prevention is more expensive than detection so using detection is more feasible than prevention. Firewall is one way to detect any malaises activity where the packets are analyses and discarded on the basis of the policies that are defined by the Network Administrator .But use of only firewall to defend is not good solution so the Intrusion Detection Systems IDS are now used for Intrusion detection rather than only firewall. Detection don’t do anything to avoid or take any action to remove that problem but simple detect an intrusion and generate an alarm or sometime take first aid like block the IP address of the system from where any malicious packet came from. In section II introduction about IDS, different types of IDS are explained .Section III section IV covers current IDS and problems with those IDSs. Different techniques of Data Mining and Event correlation are explained in section V. Section VI briefly describe our proposed system.

Intrusion Detection System (IDS)

Intrusion Detection Systems (IDS) are security tools that, like other measures such as antivirus software, firewalls and access control schemes, are intended to strengthen the security of information and communication systems .It is a device or software application that monitors network and/or system activities for malicious activities or policy violations and produces reports to a Management Station [2]. Although intrusion detection technology is immature and should not be considered as a complete defence, but at the same time it can play a significant role in overall security architecture. If an organization chooses to deploy an IDS, a range of commercial and public domain products are available that offer varying deployment costs and potential to be effective. Because any deployment will incur ongoing operation and maintenance costs, the organization should consider the full IDS life cycle before making its choice. When an IDS is properly deployed, it can provide warnings indicating that a system is under attack, even if the system is not vulnerable to the specific attack [3]. These warnings can help users alter their installation's defensive posture to increase resistance to attack. In addition, IDS can serve to confirm secure configuration and operation of other security mechanisms such as firewalls. Within its limitations, it is useful as one portion of a defensive posture, but should not be relied upon as a sole means of protection. As e-commerce sites become attractive targets and the emphasis turns from break-ins to denials of service, the situation will likely worsen.

Several types of IDS technologies exist due to the variance of network configurations. Different intrusion detection systems use different sets of features (i.e., measures on audit data) and different analytical models to determine whether the system activities are intrusive [4] .Each type has advantages and disadvantage in detection, configuration, and cost. In the case of detecting data target, intrusion detecting system can be classified as host-based and network-based

Host-based IDS (HIDS)

Its data come from the records of various host activities, including audit record of operation system, system logs, application programs information, and so on.

Network-based IDS

A Network Intrusion Detection System (NIDS) is an intrusion detection system that tries to detect malicious activity such as denial of service attacks; port scans or even attempts to crack into computers by monitoring network traffic. A NIDS reads all the incoming packets and tries to find suspicious patterns known as signatures or rules. If, for example, a large number of TCP connection requests to a very large number of different ports are observed, one could assume that there is someone conducting a port scan of some or all of the computer(s) in the network. Its data is mainly collected network generic stream going through network segments, such as: Internet packets [5]

Another classification of intrusion detection is,

Anomaly-Based IDS

Anomaly detection consists of first establishing the normal behaviour profiles for users, programs, or other resources of interest in a system, and observing the actual activities as reported in the audit data to ultimately detect any significant deviations from these profiles. Most anomaly detection approaches are statistical in nature. Anomaly detection systems can detect unknown intrusion since they require no a priori knowledge about specific intrusions. Statistical-based approaches also have the added advantage of being adaptive to evolving user and system behaviour since updating the statistical measures is relatively easy. Shortcomings of this type are , The selection of the right set of system (usage) features to be measured can vary greatly among different computing environments; The fine tuning of the deviation threshold is very ad hoc; User behavior can change dynamically and can be very inconsistent.

Misuse Detection IDS

Misuse detection IDS models function in very much the same sense as high-end computer anti-virus applications. That is, misuse detection IDS models analyse the system or network environment and compare the activity against signatures (or patterns) of known intrusive computer and network behaviour. These signatures must be updated over time to include the latest attack patterns, much like computer anti-virus applications. Main advantage of this type IDS is, if the target deployment is only a few computer systems, then a misuse-based IDS is easy to implement, update and deploy. Shortcoming of this system is unknown attack goes undetected so system has to be timely updated [6].

Comparison of Misuse Detection and Anomaly based system is given below.

Table 1- A brief comparison among Misuse-based and Anomaly-based IDS.

Misuse-Detection	Anomaly-based
The attacks uncovered under this are assumed to be true positives.	The normal packets separated under this are assumed to be true negatives.
It risks high porosity towards new and undefined attacks.	It risks the chances of normal but undefined packets to be tagged as abnormal data.
It has a chance of failure to capture many attacks.	It has a tendency to show greater number of false positives.

In our proposed system paper we have consider all types of IDS more focused on NIDS.

Existing NIDS

First major work in the area of intrusion detection was discussed by J.P Anderson. Concept that is introduced by that was as certain types of threats to the security of computer systems could be identified through a review of information contained in the system's log. This system log is available in many types of operating systems, particularly the various “flavors” of UNIX; automatically create a report which details the activity occurring on the system. Anderson identified three threats which could be identified from a concentrated review of the audit data:

- External Penetrations- Unauthorized users of the system
- Internal Penetrations- Authorized system users who utilize the system in an unauthorized manner.
- Misfeasors - Authorized user who mislead their access privileges [6]

Numbers of IDS are available in market, it's best to use a Web search to locate current products, reviews, and so forth. Commercial product literature is generally weighted towards marketing, which often makes it difficult to determine the product's functionality and detection approach. Virtually no commercial literature addresses issues such as the frequencies of false alarms, missed detections, or the system's sensitivity to traffic loads.

Real Secure

from Internet Security Systems (www.iss.net) is a real-time IDS that uses a three-part architecture consisting of a network-based recognition engine, a host-based recognition engine, and an administrator's module. The network recognition engine runs on dedicated workstations to provide network intrusion detection and response. Each network-recognition engine monitors a network segment looking for packets that match attack signatures. When a network-recognition engine detects intrusive activity, it can respond by terminating the connection, sending alerts, recording the session, reconfiguring firewalls, and so forth. The host-based engines analyse log data to recognize attacks. Each host engine examines its system's logs for evidence of intrusions and security breaches. Log data can contain information that is difficult or impossible to infer from network packet data. The host engine can prevent further incursions by terminating user processes or suspending user accounts. An administrative module manages multiple-recognition engines. The result is comprehensive protection, easily configured and administered from a single location. The administrative module is supplied with both recognition engines and is also available as a plug-in module for a variety of network and systems management environments [7].

Tripwire

is a file integrity assessment tool (www.tripwire. com) that is useful for detecting the effects of an intrusion. Tripwire creates a database of critical system file information that includes file lengths and cryptographic checksums based on each file's contents. Tripwire compares current information with a previously generated baseline and identifies changed files. Tripwire will report modified files, but the user must decide whether the modifications resulted from an intrusion. Because most monitored files are not expected to change except when new software versions are installed, changes usually indicate an unexpected or unauthorized activity. For reliable Tripwire results, users must protect the database and program from tampering, either by maintaining them offline or online using read-only storage media

Shadow and Snort

two public-domain ID tools, are unlikely to have the same level of support as commercial systems, so users will need a higher level of technical expertise to install and manage them. The effort involved is likely to pay off with a better understanding of ID and its strengths and limitations. Sensors usually reside at key monitoring points in the network, such as outside a firewall, while the analysis station resides inside the firewall. The sensor is based on public domain packet-capture software and does not pre-process the data, thus preventing an intruder from determining the detection objectives by capturing an unprotected sensor. Sensors extract packet headers and save them to a file that the analysis station reads periodically. The analysis station uses a Web-based interface to display filtering results as well as raw data. Shadow runs on many UNIX systems and Linux. Snort is a recent open-source public-domain effort to build a lightweight, efficient, ID tool that can be deployed on a wide variety of UNIX platforms. According to the Snort Web site (www.snort.org), views are quickly outdated. The "Technology" sidebar describes a sample of commercial, research, and public domain tools. Snort is a lightweight network intrusion detection system, capable of performing real-time traffic analysis and packet logging on IP networks. It can perform protocol analysis, content searching/ matching and can be used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts, and much more. Snort uses a flexible rules language to describe traffic that it should collect or pass, as well as a detection engine that utilizes a modular plug-in architecture. Snort is currently undergoing rapid development. The user community is contributing auxiliary tools for analysing and summarizing snort logs, providing additional capabilities. More importantly, there is a large group of users who contribute new signatures. As a result, new attacks are quickly represented in the signature database [3].

Problems In Current Intrusion Detection System

We measure the quality of IDS by its effectiveness, adaptability and extensibility. An IDS is effective if it has both high intrusion detection (i.e., true positive) rate and low false alarm (i.e., false positive) rate. System is adaptable if it can detect slight variations of the known intrusions, and can be quickly updated to detect new intrusions soon after they are invented. It is extensible if it can incorporate new detection modules or can be customized according to (changed) network system configurations. Current IDSs lack effectiveness. The hand-crafted rules and patterns, and the statistical measures on selected system measures are the codified "expert knowledge" in security, system design, and the particular intrusion detection approaches in use. Expert knowledge is usually incomplete and imprecise due to the complexities of the network systems. Current IDSs also lack adaptability. Experts tend to focus on analysing "current" (i.e., "known") intrusion methods and system vulnerabilities. As a result, IDSs may not be able to detect "future" (i.e., "unknown") attacks. Developing and incorporating new detection modules is slow because of the inherent "learning curve". Current IDSs lack extensibility. Reuse or customization of IDS in a new computing environment is difficult because the expert rules and statistical measures are usually ad hoc and environment-specific. Since most current intrusion detection systems are monolithic, it is also hard to add new and complementary detection modules to existing IDS.

Current IDS have a number of significant drawbacks:

Current IDS are usually tuned to detect known service level network attacks. This leaves them vulnerable to original and novel malicious attacks.

Data overload: Another aspect which does not relate directly to misuse detection but is extremely important is how much data an analyst can efficiently analyse. That amount of data he needs to look at seems to be growing rapidly. Depending on the intrusion detection tools employed by a company and its size there is the possibility for logs to reach millions of records per day [8].

False positives: is the amount of false positives IDS will generate. A false positive occurs when normal attack is mistakenly classified as malicious and treated accordingly.

False negatives: This is the case where an IDS does not generate an alert when an intrusion is actually taking place. (Classification of malicious traffic as normal) Data mining can help improve intrusion detection by addressing each and every one of the above mentioned problems [9]

Data Mining and Event Correlation: Across all industry sectors and scientific research areas, the amount of data collected and warehoused is growing at an explosive rate.

Data Mining Technology

Data mining is, at its core, pattern finding. Data Mining is to extract knowledge interested by people from large database or data warehouse; the knowledge is implied, unknown and potentially useful information. Extracted knowledge is represented as concept, rule, law and model. The purpose of data mining is to help the decision-maker to find potential association between data, found neglected elements which are perhaps very useful for trends and decision-making behaviour.

Common data mining methods and technologies are:

Correlation Analysis: correlation analysis was also called association rules, it is to find item set model knowledge frequently appeared from given data set, the purpose is to excavation the relationship that was hidden in data, for example, the customers that buy computer will buy some software, this is an association rules.

Sequential patterns: it is similar with correlation analysis, the purpose is also to excavate connection that between data, however, time series analysis focused more on the relationship of data in times, for example, and 80% people among printer buyer will buy printing paper after three months.

Classification: classification is to find model or function that can describe the typical characteristics of data set, so that it can identify ownership or categories of unknown data. Typical classify models have the linear regression model, the decision tree model, the model based on rule and the neural network model.

Clustering: Data was divided into a series of meaningful subset according to certain rules. In the same cluster, the gap between the individual is smaller, and in the different cluster, the gap is greater.

Deviation analysis: to find abnormal data from the database

Forecast: to find law according historical data, establish model, and to predict types, characteristics of the future data, etc based on the model.

Event correlation

Number of events happens in network in one second. An event in network management is typically defined as a piece of information dealing with a happening in the network, and may also be referred to as an alarm, due to its nature usually being something causing problems. Event correlation is defined in many different ways, but in its barest essence, an event correlator attempts to do exactly as the name suggests: associate events with one another in useful ways. Sometimes the sheer number of events which come in can be enough to overwhelm a engineer who cannot possibly treat each symptom separately. The object of event correlation is to attempt to pinpoint larger problems which could be causing many different symptoms to emerge. There are several subcategories of event correlation, including compression (deduplication), count, suppression, and generalization. Compression reduces multiple occurrences of the same event into a single event, likely with some kind of counter. Count is defined to be somewhat similar to compression: it is the substitution of a specified number of similar alarms with a single alarm. It is important to note that these need not necessarily be the same event, and also that there is a threshold associated with such a relation. Suppression associates a priority with alarms, and may choose to hide a lower priority alarm if a higher priority alarm exists. Finally, in the practice of generalization, alarms are associated with some sort of a superclass which is reported rather than the specific alarm. This could be seen to be useful to correlate events referring to multiple ports on the same switch or router if it has completely failed; it is unnecessary to see each particular failure if it can be determined that the entire unit is having problems.

Types of Event Correlation Systems

Rule-based Systems

A somewhat traditional approach to event correlation is that of rule -based analysis. In this approach, sets of rules are matched to events when they come in. Based on the results of each test, and the combination of events in the system, the rule-processing engine analyses data until it reaches a final state. It will then report a diagnosis, which could include, unfortunately, none at all, depending on the depth and capability of the ruleset. Unfortunately, this approach does not necessarily perform well in respect to either of our criteria. For the results to be very accurate, an excessive amount of expert knowledge is typically needed to input the correct rules and keep them updated in case of any changes or new data. The rigidity of the path through the rule sets makes it so that events are may always be compared with an inordinate amount of test cases, slowing the system down and making correlation even more difficult.

Codebook Systems

The codebook approach is somewhat similar to the rule-based approach, but rather than treating events separately, they are grouped into an alarm vector which represents all of the events. This alarm vector is then matched to problem signatures in a so-called codebook. A codebook system consists of an optimal set off alarms which must have distinguishable characteristics. In other words, no two alarms can have the same signature, as there would be no way to correctly choose one over the other. In order to optimize performance, a design should use small sets of symptoms while still providing the best guess of the cause. This can be done in the creation of the codebook by forcing a minimum Hamming distance, a measure of differing symptoms, between different causes and by reducing otherwise indistinguishable cases into one which encompasses all of them. The codebook approach always produces a diagnosis, as opposed to the rule-based scheme, though some diagnoses will be more likely to be correct than others. Unfortunately, the codebook technology needs the same expert knowledge as a rule-based system in order to accurately populate the codebook, problem signatures can be created automatically. Unfortunately, many of the details of this system appear hidden behind corporate patents; so many such claims have little researchable evidence to back them up.

Artificial Intelligence Systems

An approach that is radically different from the rule-based and codebook approaches uses various forms of artificial intelligence (AI). There are many different types of artificial intelligence, and event correlation techniques have been proposed which utilize various combinations of them, including Bayesian belief networks and expert systems. AI systems have an advantage in that, if well-programmed, they have the capability to be somewhat self-learning, helping to eliminate the continuous need for the expert knowledge of the previous systems. They also have the capability to sift through data at least as fast as the other systems to produce their results. They claim that when they incorporate a technique called inverse learning into their scheme, their system will never return a wrong answer as noise in the system increases [10].

Proposed System

Most of the NIDSs focus on low-level attacks or anomalies, and raise alerts independently, though there may be logical connections between them. In situations where there are intensive intrusions, not only will actual alerts be mixed with false alerts, but the amount of alerts will also become unmanageable. As a result, it is difficult for human users or intrusion response systems to understand the alerts and take appropriate actions. In our proposed system we are trying to develop a practical technique to address this issue with. Basically there are many approaches to solve this problem but we are going to use detecting anomaly access pattern. The proposed approach constructs attack scenarios by correlating alerts on the basis of prerequisites and consequences of intrusions. Intuitively, the prerequisite of an intrusion is the necessary condition for the intrusion to be successful, while the consequence of an intrusion is the possible outcome of the intrusion. Based on the prerequisites and consequences of different types of attacks, the proposed approach correlates alerts by (partially) matching the consequence of some previous alerts and the prerequisite of some later ones.

In our proposed system we are using signature based and anomaly base system intrusion detection methods. When any new attack pattern is discovered by the anomaly engine that signature will be include in the signature database of signature model and that signature can later be used for detection of same attack. This proposed system also includes the host based and Network based IDS. Because of this we can highly protect our server. For host based IDS in web server system logs will be analysed by the administrator and if it is that any unwanted process is running on the server then that process will be killed by the administrator, this can be done manually or by the system. True alerts belong to intrusion generated by the IDS systems are not isolated; they also reflect the sequential pattern of the attacker. However, IDS systems consider these alerts as individual events and report that to the administrator with huge amount of alerts most of them are false positives or not critical for the protected system. False alarm rate can be reduced by finding logical connection between the alerts and ignoring the isolated alerts.

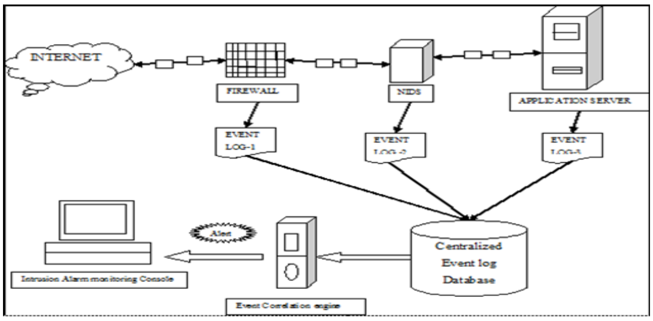

Fig. 1- System Architecture of Event Correlation System

Conclusion

In our proposed system we are going to develop NIDS which will minimize false positive alarm rate which is one of the key factors that affects the efficiency of the NIDS. Reducing false positive alarm reduces the complex process of analysing steps, reduces the frustration of Analyzer to do the tedious process of analysing when intrusion is not present. It will also reduce the cost of the security management. In this proposed system we are also considering the new attack pattern generation for that anomaly model is used this will increase the efficiency of system by detecting unseen attack.

References

[1] Chunyu Miao, Wei Chen (2010) *A Study of Intrusion Detection System Based on Data Mining.*

[2] Garcı´a-Teodoroa P., Dı´az-Verdejoa J., Macia´-Ferna´ndeza G., Va´zquezb E. (2009) *computers & security* 18 - 2 8.

[3] *Intrusion Detection Systems [Online] Available on http://ids.nic.in/jces%20tnl%20oct%202008/ids/ids.htm*.

[4] Teresa F. Lunt, Ann Tamaru, Fred Gilham, Jagannathan R., Caveh Jalali, Peter G. Neumann (1992) *A Real time intrusion detection Expert System*.

[5] *Intrusion Detection Systems [Online] Available on http://en.wikipedia.org/wiki/Intrusion_detection_system*.

[6] Liu Wenjun An (2010) *IEEE* 978-1-4244-8217-7110.

[7] Darrin Wassom (2001) *Intrusion Detection Systems An Overview of RealSecure SANS Institute*.

[8] James Cannady, Jay Harrell. *A Comparative Analysis of Current Intrusion Detection Technologies*.

[9] Theodoros Lappas, Konstantinos Pelechrinis *Data Mining Techniques for (Network) Intrusion Detection Systems.*

[10] Andreas M□ller, Christoph Goldi, Bernhard Plattner (2009) *Event Correlation Engine.*

[11] Michael Tiffany (2002) *A Survey of Event Correlation Techniques and Related Topics*.