Knowledge-Based Intrusion Detection

Teresa F. Lunt, R. Jagannathan, Rosanna Lee, Alan Whitehurst
SRI International, Computer Science Laboratory
Menlo Park, California 94025

Sherry Listgarten
Stanford University, Computer Science Department
Stanford, California 94305

Abstract

We describe the expert-system aspects of IDES, a system for computer intrusion detection. IDES employs two distinct approaches to detect anomalies (which could signify intrusions) in a computer system, namely, statistical and rule-based anomaly detection. In the statistical approach, recent behavior of a subject of a computer system is compared with observed behavior and any significant deviation is considered anomalous. In the rule-based approach, acceptable behavior of a subject is captured by a set of rules which is used to identify anomalous observed behavior. We claim that integrating the two approaches in IDES provides for a comprehensive system for detecting intrusions as they occur.

1 Introduction

Although a computer system's primary defense is its access controls, it is clear from numerous newspaper accounts of virus attacks, break-ins and computerized thefts that access control mechanisms cannot be relied upon in most cases to safeguard against a penetration or insider attack. Even the most secure systems are vulnerable to abuse by insiders who misuse their privileges, and audit trails may be the only means of detecting authorized but abusive user activity.

While many computer systems collect audit data, most do not have any capability for automated analysis of that data. Moreover, those systems that do collect audit data generally collect large volumes of data that are not necessarily security relevant. Thus, for security analysis, a security officer must wade through stacks of printed output of audit data. Besides the pure tedium of this task, the sheer volume of the data makes it impossible for the security officer to detect suspicious activity that does not conform to a handful of obvious intrusion scenarios. What is needed is the capability for automated security analysis of audit trails.

To address this problem, a team at SRI is developing a real-time intrusion-detection expert system (IDES). IDES is an independent system that runs on its own hardware and processes audit data characterizing subject activity received from a target system [1,2]. The goal of IDES is to provide a system-independent mechanism for real-time detection of security violations, whether they are initiated by outsiders who attempt to break into a system or by insiders who attempt to misuse their privileges. IDES is based on the intrusion-detection model developed by Denning [3,4]. This model is independent of any particular target system, application environment, system vulnerability, or type of intrusion, thereby providing a framework for a general-purpose intrusion-detection system using real-time analysis of audit data. See [5] for a survey of other intrusion-detection projects.

The rest of the paper is organized as follows. Section 2 discusses the threat we are addressing and introduces two distinct approaches to detecting intrusions. The overall architecture of IDES is presented in Section 3. Section 4 describes how anomalies are detected statistically. Section 5 describes how a rule-base is used to detect anomalies. In Section 6, we summarize the main ideas of the paper and address privacy issues.

2 Intrusion Detection

Anderson, who was apparently the first to seriously study the feasibility of automated audit trail analysis, categorized the threats that could be addressed by audit trail analysis as the following [6]:

- External penetrators (who are not authorized to use the computer).
- Internal penetrators (who are authorized use of the computer, but are not authorized for the data, program, or resource accessed), including the following:
  - Masqueraders (who operate under another user's id and password).
  - Clandestine users (who evade auditing and access controls).
- Misfeasors (authorized users of the computer and resources accessed who misuse their privileges).

Anderson suggested that masqueraders can be detected by observing departures from established patterns of use for individual users. This is one approach taken by IDES; thus, we can expect that IDES will be potentially capable of detecting masqueraders.
Anderson suggested that external penetrators can be detected by auditing failed login attempts, and that some would-be internal penetrators can be detected by observing failed access attempts to files, programs, and other resources. This suggests an approach of characterizing intrusions, as opposed to characterizing normal user behavior. IDES includes expert-system rules that characterize certain types of intrusions. This gives IDES the potential to detect external and internal penetrators.

IDES takes two approaches to detecting legitimate users who abuse their privileges. IDES’ expert-system rules include a priori rules for “socially unacceptable” behavior that are analogous to those that characterize intrusions. IDES also compares a user’s behavior with the norm established for the class of user, i.e., the group, to which the user belongs, to detect abuse of privilege. We expect this approach to be especially useful when there is a very large number of users (i.e., in the thousands or tens of thousands) who may operate in distinct roles or job capacities.

The clandestine user can evade auditing by using system privilege or by operating at a level below that at which auditing occurs. The former could be detected by auditing all use of functions that turn off or suspend auditing, change the specific users being audited, or change other auditing parameters. The latter could be addressed by performing auditing at a low level, such as auditing system service or kernel calls.

To detect the clandestine user, Anderson suggests monitoring certain system-wide parameters, such as CPU, memory, and disk activity, and comparing these with what has been historically established as “usual” or normal for that facility. IDES performs system-wide monitoring for several variables and keeps a profile for the system as a whole. We do not claim that IDES will ever be effective for detecting clandestine users. However, although a skilled penetrator will be able to disable the audit mechanisms in order to work undetected, auditing and intrusion-detection mechanisms are still of value in detecting the less skilled penetrator, because they increase the difficulty of penetration. As Linde has suggested [71], auditing and intrusion-detection mechanisms can make it so difficult for a penetrator to avoid detection that other methods, such as bribing a user, will be more attractive.

3 The IDES Architecture

IDES operates independently of the computer system that it monitor (which we call the target system.) The target system continually generates audit data in response to user activity, in a specific format as defined by IDES, and transmits the audit records to IDES. Upon receiving audit data, IDES validates it and makes it available to its statistical and rule-based components. Anomalies, when detected, are immediately reported to the security officer through a sophisticated user-interface. Figure 1 shows the overall architecture of IDES. Two aspects of the IDES architecture are worth highlighting. The first aspect is that IDES attempts to detect intrusions as they occur (as opposed to after they occur.) This is made possible by implementing IDES on a high-performance computer and by using a well-tuned database to access specific information with minimal latency. The second aspect is that separating IDES from the system that it monitors enhances the security of IDES itself and allows for multiple systems to be monitored simultaneously.

4 Statistical Intrusion Detection

IDES monitors subjects for abnormal behavior using information maintained in a dynamically updated knowledgebase. The subject types monitored by IDES are users, groups, remote hosts, and overall target systems. Groups can be groups of users, of hosts, or of target systems. Monitoring groups enables IDES to detect when the behavior of an individual member of a group deviates from the overall average behavior of the group. Monitoring a target system enables IDES to detect system-wide deviations in behavior, when such deviations cannot be attributed to a single user. For example, the number of system-wide login attempts may be indicative of an intrusion, although they might not all be attributable to a single user.

IDES determines whether observed behavior as reported in the audit data is normal with respect to past or acceptable behavior characterized by specific intrusion-detection measures. A measure is an aspect of a subject’s behavior on the target system. The measures are applied to individual sessions, whose definition depends on the type of subject;
for example, a user session is the time between login and logout. IDES implements two types of intrusion-detection measures. A continuous measure is a function of some numerically quantifiable aspect of observed behavior such that the function value changes (i.e., accumulates) during the course of a session; connect time is an example of a continuous measure. A discrete measure is a function of observed behavior whose range of values is a finite, unordered set that is used to characterize some aspect of behavior (e.g., the set of terminal locations).

IDES maintains a statistical knowledge-base for subjects called profiles. A profile is a description of a subject's normal (i.e., expected) behavior during a session with respect to a set of intrusion-detection measures. Profiles are designed to require a minimum amount of storage for historical data and yet record sufficient information which can be readily decoded and interpreted during anomaly detection. Rather than storing all historical audit data, the profiles keep only statistics such as frequency tables, means, and covariances. More specifically, each profile for a discrete measure consists of a list of values and the associated probability of occurring based on previous behavior for the subject for that measure. Each profile for a continuous measure contains the first and second moments of the joint probability distribution of the values of the measures over prior sessions for the subject.

The deductive process used by IDES in determining whether behavior is anomalous is based on statistics, controlled by dynamically adjustable parameters specific for each subject. Each session is described by a vector of intrusion-detection variables, corresponding to the measures recorded in the profiles. Measures can be turned "on" or "off" (i.e., included in the statistical tests) for each subject, depending on whether the measures are deemed to be useful for that subject. As each audit record arrives, the relevant profiles are retrieved from the knowledge-base and compared with the vector of intrusion-detection variables. If the point in N-space defined by the vector of intrusion-detection variables is sufficiently far from the point defined by the expected values stored in the profiles, with respect to the historical covariances for the variables stored in the profiles, then the record is considered anomalous. Thus, the statistical procedures pay attention not only to whether an audit variable is too high or too low, but also to whether any audit variable is too high or too low relative to the values of the other audit variables (in other words, the correlation between variables). Thus, IDES evaluates the total usage pattern, not just how the subject behaves with respect to each measure considered singly.

To be useful, IDES must maximize the true positive rate (i.e., the percentage of intrusive use correctly identified as abnormal) and minimize the false positive rate (i.e., the percentage of normal use incorrectly identified as abnormal). Although the false positive rate can be reduced by raising the threshold of the statistical test (so that fewer events are considered abnormal), this also lowers the true positive rate. We plan to make the false positive rates adjustable individually for each measure, for each subject. This will allow IDES to raise its sensitivity to certain measures depending on its knowledge of its target population.

The statistical knowledge-base is updated daily using the current day's observed behavior of the subjects. Before incorporating the new audit data into the profiles, the frequency tables, means, and covariances stored in each profile are first aged by multiplying them by an exponential decay factor. This has the effect of creating a moving time window for the profile data, so that the expected behavior is influenced most strongly by the most recently observed behavior. Thus, IDES adaptively learns subjects' behavior patterns as subjects alter their behavior, their corresponding profiles will change.

5 Rule-Based Intrusion Detection

There are obvious difficulties with attempting to detect intrusions solely on the basis of departures from observed norms for individual users. Although some users may have well-established patterns of behavior, logging on and off at close to the same times every day and having a characteristic level and type of activity, others may have erratic work hours, may differ radically from day to day in the amount and type of their activity, and may use the computer from several different locations and different time zones (e.g., in the office, at home, and on travel). Thus, for the latter type of user, almost anything is "normal," and a masquerader might easily go undetected.

The ability to discriminate between a user's normal behavior and suspicious behavior depends on how widely that user's behavior fluctuates and on the range of "normal" behavior encompassed by that user. And although this approach might be successful for detecting penetrators and masqueraders, it may not have the same success with legitimate users who abuse their privileges, especially if such abuse is "normal" for those users. Moreover, the approach is vulnerable to defeat by an insider who knows that his or her behavior is being compared with his or her previously established behavior pattern and who slowly varies their behavior over time, until they have established a new behavior pattern within which they can safely mount an attack. Trend analysis on user behavior patterns, that is, observing how fast and in which direction user's normal behavior changes over time, may be useful in detecting such attacks. In addition, trends in variances in user behavior can be used to measure the rate at which a user's normal range "spreads" over time. We are currently developing "second order" measures to detect such trends in user behavior.

An obvious second line of defense against these weaknesses is to enforce rules that describe suspicious behavior that is independent of whether a user is deviating from past behavior patterns. In IDES, we include the capability to characterize intrusions using rules. These rules describe suspicious behavior that is based on knowledge of past intrusions, known system vulnerabilities, and the installation-specific security policy. Thus, IDES is also sen-
sitive to known or posited intrusion scenarios that may not be easily detectable as deviations from past behavior. In effect, the rules in the expert system component of IDES constitutes a minimum standard of conduct for users of the host system. While the statistical anomaly detector attempts to define normal behavior for a user, the expert system component attempts to define proper behavior, and to detect any “breaches of etiquette.”

The IDES rule-based component is based on the P-BEST expert system shell. The Production-Based Expert System Toolset (P-BEST) is a forward-chaining, LISP-based expert system development environment. It provides an integrated set of facilities for the creation and debugging of complex knowledge bases. The basic P-BEST strategy revolves around maintenance of an expert system state. The system state is represented by the union of the states of the Fact Base, the Rule Base, and any external data structures unique to a given application. This union is referred to as the Knowledge Base (KB).

The P-BEST inference engine is a distributed mechanism for matching facts to rule antecedent patterns, performing binding analysis, and invoking rule consequents. It is distributed in the sense that for each rule in the rule-base, the P-BEST compiler will produce a number of primitive LISP functions which implement the semantics of the rule. As facts are asserted, each rule in the Rule Base which references the type of knowledge being asserted examines the fact for a potential match. When a pattern in the antecedent of a rule matches the fact being asserted, a binding is created and associated with both the fact and the rule. If such a binding is created, and if, as a result of this binding, all the patterns of a rule have been matched, then binding analysis is performed to determine if all of the variable values referenced by the rule/fact bindings are consistent. In other words, variable values are unified and any additional constraints are evaluated to determine if there is a consistent interpretation of the binding set which would allow a rule to fire. The rules with rule/fact bindings that meet binding analysis are gathered into a conflict set, and the “best” rule in this set is selected and fired. The process of selecting a rule to fire from a number of possibilities is called conflict resolution. The consequent of the selected rule will be executed and generally alter the state of the Knowledge Base, which will normally cause the creation of additional rule/fact bindings. This process then repeats indefinitely or until no further candidates for firing remain.

The IDES rule-based component operates in parallel with the component of IDES that performs statistical anomaly detection, but is loosely coupled in the sense that the inferences made by the two subsystems are independent. The rule-based detector and statistical anomaly detector share the same database of audit records, and may produce similar anomaly reports, but the internal processing of the two systems is done in isolation. We are investigating the feasibility and desirability of tightening this coupling so that the two independent systems could review each other’s conclusions, perhaps in some sense “voting” on the outcome. A more tightly-coupled system might be able to make more complex deductions, thereby reducing the false-positive rate of anomaly reports and eliminating the possibility of the same suspicious behavior generating multiple alarms (i.e., the same anomaly being reported by both the rule-based anomaly detector and the statistical anomaly detector).

The knowledge-base of the expert system contains information about known system vulnerabilities and reported attack scenarios, as well as our intuition about suspicious behavior. The rules are instantaneous in that they do not depend on past user or system behavior. As the activity of each active user on the system generates audit records, these records are evaluated by the rule-based component and each user is given a suspicion rating. This rating starts at zero and is incremented with each observation of suspicious or improper behavior. The amount incremented is dependent upon the seriousness of the observed misbehavior. When the suspicion rating of a particular user exceeds a pre-set threshold value, an anomaly report is generated.

The rule-based component also generates a textual justification of its conclusions.

Figure 2 is an example of an IDES rule expressed in the language of the P-BEST expert system shell. Programmed Login Attack looks for two login attempts by the same user in such a short period of time that it is doubtful that they could have been generated by a human at a conventional computer keyboard. This rule is designed to detect the type of attack popularized by the movie “War Games”, in which a penetrator uses a computer to repeatedly generate account-name/password pairs. Although this rule is fixed, in that it does not reason about the user’s past behavior, the actual LOGIN_DELAY is a parameter which is set by each site according to the type of system involved. For instance, some systems force a set time-delay between login prompts; in such systems the approximate proper value for LOGIN_DELAY would be the forced delay between prompts plus the expected minimum time for the user to enter the user identifier plus the expected minimum amount of time necessary to enter the user password. If two failed login attempts are generated by the same user within a time-span less than the expected LOGIN_DELAY, then the rule-based component increments the user’s suspicion rating.

Audit data from the monitored system is matched against these rules to determine whether the behavior is suspicious. A limitation of this approach, if used in isolation, is that one is looking for known vulnerabilities and attacks, and the greatest threat may be the vulnerabilities we do not yet know about and the attacks that have not yet been tried; one is in a position of playing “catch-up” with the intruders. Such an approach, used in conjunction with detecting departures from established user norms, addresses the vulnerabilities and has the strengths of both approaches.
We have described the expert-system aspects of a system for computer intrusion detection called IDES. IDES combines two distinct approaches to intrusion detection, namely, statistical and rule-based approaches. The statistical approach characterizes recent behavior of subjects of a computer system using profiles and detects anomalies (which can be intrusions) when the observed specific behavior of a subject deviates significantly from its profiled expected behavior. The rule-based approach uses a rule-base consisting of "rules-of-thumb" to detect events that may signify intrusions. Through the combination of statistical analysis and rule-based approaches, intrusion detection can be achieved more comprehensively than would be possible utilizing either of the approaches separately.

IDES raises the issue of privacy in the use of a computer system. Denning [8] has voiced concerns about the use of computer monitoring for security purposes, and has suggested that security measures such as intrusion-detection mechanisms may actually increase the threat of computer abuse by engendering employee dissatisfaction; whence the emergence of the disgruntled employee and the so-called insider threat. As Denning points out, there are privacy issues in even such apparently benign security mechanisms as file backups, archives, and keeping an audit trail of user activity (as is required by the DoD Trusted Computer System Evaluation Criteria [9] for systems rated C2 and above), in that there is great potential for abuse of such data. Real-time intrusion-detection systems make possible an even greater degree of invasion of privacy and other potential objectionable activity, such as employee performance monitoring. A recent study indicates that the use of computerized employee performance monitoring systems can lead to increased stress, lower levels of satisfaction, and a decrease in the quality of relationships with peers and management among the monitored workers [10]. How ever, this same study found that workers were not opposed to computerized performance monitoring in principle, but that it is how it is used by management that determines the effects. These findings underline the need for concern for appropriate use of intrusion-detection technology. Denning recommends that "a policy be developed regarding threat monitoring that addresses such areas as limits on threat monitoring, use of the results obtained from monitoring, obtaining informed consent of users, and providing due notice of intent to monitor" [8]. However, there are many, particularly in the intelligence community, who consider that even knowledge of the existence of an intrusion-detection system might be an aid to a would-be penetrator. It is our view that IDES needs to be used in a responsible manner in order to be effective in enhancing computer security. This includes using information on users of a computer system only for ensuring computer security.

6 Conclusions

Unix is a trademark of AT&T Bell Laboratories.
Acknowledgements

This work was supported by the U.S. Navy, SPAWAR, which funded SRI through subcontract 9-X5H-4074J-1 with the Los Alamos National Laboratory. We wish to acknowledge the other members of the project team, without whom this project would never have been accomplished: Dave Edwards, Hal Javitz, Peter Neumann, and Al Valdes.

References


