Computer Intrusion Detection: A Comparative Survey

M. Esmaili R. Safavi-Naini J. Pieprzyk

Center for Computer Security Research
University of Wollongong
Wollongong, NSW 2522,
Australia.

{mansour,rei,josef}@cs.uow.edu.au

May 17, 1995

Tech-report #95-07

Abstract

Advances in computer and communication technologies have resulted in highly integrated distributed systems that allow users to access information and resources from all over the globe. By introducing many more possible attacking points, this interconnectivity adds new dimensions to the long standing problem of providing security in a computer system. Rapid increase in the number of reported intrusions, break-ins and computer thefts results in an ever-increasing need for applying effective computer security measures.

The number of recently developed, or under development systems and tools that can be used for detection of abuse of computer systems is growing. In this report we present a comparative review of the state-of-the-art intrusion detection systems (IDS) and methods, underlining the strength and limitations of each. We also suggest directions for future development and research.

1 Introduction

There is a growing concern about the problem of ever increasing intrusions into computers and computer networks on the Internet and other networks. Intruders can be characterized as "joy riders" with no malicious intent, as thieves aiming to appropriate resources of the computer system or those controlled by the system, or as terrorists aiming to destroy or incapacitate the system. Intruders are sometimes known as *hackers* as well. The standard reference work on psychiatric disorders (DSM III; APA, 1980) defines the Narcissistic Disorder in these terms:

> 'The essential feature is a Personality Disorder ... in which there are a grandiose sense of self-importance or uniqueness; preoccupation with fantasies of unlimited success; exhibitionistic need for constant attention and admiration; characteristic responses to threats to self-esteem; and characteristic disturbances in interpersonal relationships, such as feelings of entitlement, interpersonal exploitativeness, relationships that alternate between the extremes of overidealization and devaluation, and lack of empathy ....
>
> ...In response to criticism, defeat or disappointment, there is a cool indifference or marked feelings of rage, inferiority, shame, humilia-

> tion, or emptiness ...Entitlement, the expectation of special favors without assuming responsibilities, is usually present. For example, surprise and anger are felt because others will not do what is wanted; more is expected from people than is reasonable.'

This sounds like criminal hackers. During the 1990 December holiday season, some 25 hackers gathered for their 'Christmas Con' in a hotel near Houston airport. After consuming too many beers and damaging fire alarms, the group was evicted from the hotel. This sort of behavior is associated with the Antisocial Personality Disorder, whose '...essential feature is ...a history of continuous and chronic antisocial behavior in which the rights of others are violated ...' (DSM III; APA, 1980).

The study of providing security in computer networks is a rapidly growing area of interest because the network is the medium over which most attacks or intrusions on computer systems are launched. One approach to solving this problem is the *intrusion detection* concept. The basic premise of this approach is that not only abandoning the existing and huge infrastructure of possibly-insecure computer and network systems is impossible, but also replacing them by totally secure systems may not be feasible or cost effective.

The importance of securing the data and information maintained by a corporation, has now become a driving force in the development of numerous systems that perform computer security audit trail analysis. These systems are generally classified as "*intrusion detection*" systems. The primary purpose of these systems is to expose computer security threats in a timely manner.

1.1 Background

An *intrusion* is an unauthorized access or usage of the resources of a computer system. The goal of intrusion detection systems is to provide mechanisms for detection of security violations (either in real-time or batch-mode). The violations are initiated by outsiders attempting to break into a system or by insiders attempting to misuse their privileges. While some threats can be effectively dealt with, using real-time methods [1, 6, 16], detection of others remain too resource-intensive and are usually handled by post facto security auditing.

A landmark study by Anderson [1] suggested that intrusions can be detected by analyzing audit records. External intrusion attempts can be detected by auditing login records, while other internal intrusions can be detected by analyzing the attempts to access resources. This observation has initiated wide spread research interest into ways of analyzing audit data and developing procedures and algorithms for automating the security analysis of audit trails [2, 3, 5, 11]. The importance of this automation is even more emphasized if the enormity of the produced audit data is considered. Some of these developments provided the required experimental evidence that users can in fact be distinguished on the basis of the pattern of the computer system usage.

The next step in the development of intrusion detection systems has been to apply expert systems to detect intrusions that are *similar* to already known incidents of intrusions. Intrusion Detection Expert System (IDES) [17], developed by SRI International is an early attempt in this direction.

Today's intrusion detection systems use one or a combination of the following methods:

1. encoding *known* intrusion patterns into knowledge bases and monitoring the system audit trail for sequences that resemble intrusive patterns (intrusion detection expert systems).

2. developing profiles of system subjects or *entities* that describe *normal* behavior patterns and attempting to identify deviations from the established patterns (profile-based intrusion detection systems).

This report attempts to survey several current approaches to intrusion detection. The survey summarizes both the strengths and weaknesses of each approach.

2 Audit Record

Despite significant progress in improving security of computer systems over the past decade most systems are still vulnerable to some kind of attack. These attacks include the threat posed by authorized users who abuse their privileges. Hence it is important to provide user accountability in the system. It is also important to have provisions that facilitate damage assessment once abusive usage is discovered.

To accommodate these needs, systems usually implement a form of transaction record keeping called *audit collection*. Audit collection is an important management tool for computer systems in that it establishes both accountability for users and aids in damage assessment once an abuse has been discovered. An *audit trail* is a chronological record of system activities that enables the system manager to reconstruct and examine the sequences of actions performed by an individual user.

However, until recently, the ability to utilize audit trails as a means of detecting system compromises has been limited by the enormity of the volume of audit data. To provide accountability and possibility of damage assessment, the audit service must record the occurrences of all *security-relevant events*, which on most systems results in a prohibitively large quantity of audit data.

Automated analysis of audit data provides a practical means for efficiently analyzing large volumes of audit data. *intrusion detection systems* are automated audit data analysis tools that not only aid in identifying security violations, but can also detect threats to security by real-time tracking of user behavior.

3 Early Work

One of the earliest studies on intrusion detection was done by Jim Anderson [1]. The study focused on the improvement of computer security auditing facilities and surveillance capabilities. This report, as well as another recent survey [16] by Lunt, distinguishes different kinds of security breaches which can be addressed by audit trail analysis.

1. *Legitimate users* with authorization to access the system. A legitimate user participates in illicit activity on a computer system is referred to as a *misfeasor*.

2. *Masqueraders* who subvert a computer system's access control mechanisms to exploit legitimate users' accounts.

3. *Clandestine users* who seize supervisory control of the system, and utilize this control to either operate below the level of audit collection or suppress audit collection altogether.

Anderson's study indicated that one could identify patterns of usage for a particular user by studying the user's audit trail over a period of time. In effect, the pattern of usage discerned from the user's audit trail (i.e., the user's "normal" behavior pattern) could be used to detect behavior that is not normal. The study concluded that these patterns of abnormal behavior or anomalies, would be useful in distinguishing a masquerader from a legitimate user.

Misfeasors are much more difficult to detect than masqueraders, as the distinction between abnormal and normal behavior does not apply. Given the small amount of misuse by a legitimate user, Anderson concluded that such violations would be undetectable by solely searching for anomalous behavior. However, if a legitimate user's authority is misused to access unauthorized data, such situations could be detected and the abuse recognized. However, Anderson was not as optimistic towards the problem of detecting clandestine users, who have the ability to gain privileged access and either disable or subvert the auditing mechanisms. The search for clandestine users through the analysis of audit data is in fact a search for what is not present in the trail.

The findings of Anderson's seminal study appear in different forms in each and every intrusion detection approach discussed here. The following terms were defined by Anderson in [1], and are used extensively in intrusion detection literature:

Threat: The potential for deliberate and unauthorized

- access of information,
- manipulation of information,
- rendering of a system inoperable or unreliable.

Attack: A scenario or set of actions formulated to carry out a threat.

Penetration or Intrusion: A successful attack.

An *attacker* is an individual who attempts an attack on a system. A *penetrator*, or *intruder*, is a successful attacker. However, for the purposes of this discussion, there will be no distinction between an attacker, a penetrator and an intruder.

The next section gives a brief review of different approaches to intrusion detection and discusses issues to be considered with regard to particular computer environments.

4 Issues

There are many issues to be considered while implementing an intrusion detection system. Among the most important ones are the topology of the target system (network vs stand-alone) and the analysis mode (batch vs real-time analysis).

4.1 Network vs Stand-alone

A *stand-alone system* is a system configuration consisting of a host computer with several terminals attached to it. The auditing process is done on the host computer. A *network system* is a collection of, possibly heterogeneous, hosts serving many terminals. To audit the network each host computer needs to be audited.

In [19], Schaen lists issues related to auditing in a networked environment. These include the data, the storage, the volume reduction, and the period of time for keeping the data. These issues are also of interest for stand-alone systems but become more complex and multi-faceted for network systems. Schaen argues that protection requirements for a network environment is more demanding than a stand-alone system because of the multiplicity of the components, the multiplicity of administrative domains, the need for a component to authenticate itself to another and the need to protect communications. Scahen also lists the issues regarding the integrity of the audit data, such as integration of the audit trails collected by different components which can be done at collection or analysis time.

Schaen also discusses the issues that arise in using audit data, and addresses the questions such as how to integrate multiple audit trails, how to trace transactions across multiple components, and how to analyze the vast amounts of data collected.

4.2 Batch vs Real-time

Another important issue in audit data analysis is whether it has to be done in real-time or in batch-mode. Batch-mode analysis has a key advantage, analysis can be done during low periods of cpu usage and/or at another computing facility. However, the disadvantage is that an impending compromise cannot be preempted, hence the damage will be detected after it is reflected.

Real-time analysis can also be done on a separate machine, but then a high-speed reliable data channel is needed between the monitored (target) system and the analysis system. Manual analysis of audit data is impractical and real-time manual analysis is absolutely impossible.

For a computer system with high security requirements, real-time analysis is the only solution. In this case, software and hardware performance is a major issue.

4.3 Different Approaches

There are three main approaches taken by various intrusion detection tools:

- *Statistical anomaly detection* approach uses statistical analysis to measure variations in the amount and type of audit data. There are two different techniques to be discussed under this category:
  - *Threshold Detection*: In this approach, each occurrence of a spe-

cific event is recorded. The idea is that an unusually high number of occurrences within a specified period may indicate the presence of an intruder. The difficulty with this technique is to determine the threshold number and the interval of analysis. NADIR [13] and MIDAS [20] employ threshold detection components in their systems.

  - *Profile-based*: This approach uses statistical measures to identify expected behavior of users or user groups. Masqueraders and misfeasors can be detected by monitoring the system's audit log for user activity that deviates from established patterns of behavior. The main advantage of profile-based anomaly detection is that it does not need any prior knowledge about the security flaws of the target system. The issue that may hinder the effectiveness of these systems are false positive/negative rates and gradual misbehaviors.

- *Rule-based Anomaly Detection*: The major difference between rule-based and statistical anomaly detection approach is that in the former sets of rules are used to represent and store the usage patterns, whereas in the latter statistical formulas are used to identify usage patterns in audit data. This approach shares the advantage and disadvantages of the statistical anomaly detection. Examples of this approach are W&S (Wisdom and Sense) [22] and TIM (Time-based Inductive Machine) [4] and ASAX (Advanced Security Audit trail Analysis on uniX) [12].

- *Rule-based Penetration Identification*: Tools which adopt this approach are characterized by their expert system properties that fire rules when audit information indicates illegal activities. Most current intrusion detection tools, such as IDES [17], NADIR, and W&S, supplement their anomaly detection components with rule-based expert system components.

5 Approaches to Intrusion Detection

In this section examples of intrusion detection tools that employ the above mentioned approaches will be given.

Essentially no approach stands alone as a catch-all for computer penetrations. Instead, each approach is suited to identify a subset of the security violations to which a computer system is subject to. For example, statistical anomaly detection is effective in identifying masqueraders, whereas in many cases it is unable to identify even well-known penetrations by misfeasors. Fortunately, other approaches such as Rule-based Penetration Identification, are able to represent and detect known penetrations from the audit trail. Hence, intrusion detection system designers often implement multiple approaches within a single intrusion detection system [2, 10, 11].

5.1 Statistical Anomaly Detection

One of the most prevalent approaches used in the development of intrusion detection systems is the use of statistical analysis to measure variations in the volume and type of audit data. The statistical analysis of audit data can be applied to the individual's audit trails or all of the target system audit records as a whole [16].

5.1.1 Threshold Detection

Threshold detection, or summary statistics, is one of the most basic forms of intrusion detection. Threshold analysis alone, however, is a poor detector of even semi-sophisticated intrusions. It is usually implemented as a small sub-component within those intrusion detection systems that do employ it. When implementing a threshold detector, the most obvious difficulty is in identifying the threshold number for a specific event. Both the threshold number and the time interval of the analysis may be left to the discretion of the site security officer (SSO) and depend on the security-relevance of the event being monitored, and also the number of occurrences in the past.

The Multics Intrusion Detection and Altering

System (MIDAS) [20] uses a threshold detector as one of its subcomponents. MIDAS is a real-time intrusion detection tool operating on the National Computer Security Center's (NCSC) network mainframe, Dockmaster. MIDAS relies on three heuristic approaches to detect abuse: the immediate attack heuristic, the user anomaly heuristic, and the system state heuristic.

The immediate attack heuristic offers MIDAS' first line of defense by employing threshold detection. Immediate attack rules are in place because certain audit entries, even in isolation from other data, may be anomalous enough to raise suspicion immediately. For example, break-ins to user accounts are often preceded by a large number of failed logins and hence MIDAS searches for, and acts upon, a high number of failed login audit records.

Other general examples of threshold detection include expert rules that fire when a predefined number of I/O errors occur, or when an extraordinary amount of deletions or references are observed. Remote connection audit trails may also be a suitable area to employ threshold analysis. Systems may install threshold detectors to track unusually large data transfers or other accesses from remote sites.

The Network Anomaly Detection and Intrusion Reporter (NADIR) [13] is another example of a network intrusion detection system that utilizes threshold detection. NADIR, developed at Los Alamos National Laboratory, is a real-time audit data analysis tool targeted for the Integrated Computing Network (ICN).

5.1.2 Statistical Profile-Based Anomaly Detection

Profile-based anomaly detection is one of the earliest, and perhaps most generally accepted, intrusion detection techniques. Both rule-based, and purely statistical forms of profile-based anomaly detection have been developed. As mentioned earlier, [1] was the first published study to propose that intrusions could be detected by monitoring a system's audit logs for usages that deviate from the established patterns of usage.

Several years after Anderson's paper, Denning and Neumann presented a more complete and focused study of statistical profile-based anomaly detection [7]. This paper introduced the SRI Intrusion Detection Expert System (IDES) model which has since moved from a model to a real-time audit data analysis tool. IDES, and its new version called NIDES (Next generation IDES), use both a profile-based anomaly detection component and a rule-based component for identifying known penetrations [17, 15]. IDES has made a significant impact on the intrusion detection community and has influenced the development of many intrusion detection systems. The profile-based anomaly detection component of IDES uses statistical measures to identify expected behavior at the user, group, remote host and target system levels.

Other examples of IDS that implement statistical forms of anomaly detection include MIDAS [20], NADIR [13] and Haystack [21]. All these systems are similar to IDES, in both the type of penetrations they focus on and the statistical formulas they use to detect penetrations. However, each has its unique way of implementing statistical profile-based anomaly detection.

MIDAS, which is heavily influenced by the earlier work on the IDES model, searches for anomalous behavior of users and systems. However, whereas IDES is designed to be highly portable, MIDAS was developed as a very system specific implementation for the NCSC's Dockmaster. NADIR is also a system specific implementation, targeted for the Integrated Computing Network (ICN). NADIR assigns weighted anomaly factors, based on deviations found within user session, to profiles. A graphics interface is provided to aid the SSO in determining the usage deviations among system users. Haystack, developed by Haystack Laboratories Inc., runs in batch-mode with both data reduction and statistical analysis components. Haystack employs two methods of statistical analysis. One method uses a weighted multinomial "suspicion quotient", which represents the degree of resemblance of a user's session to a target intrusion for that user's security group. The second method identifies significant variations be-

tween a user's past usage patterns and the user's most recent sessions.

Finally, the Information Security Officer's Assistant (ISOA)[23], developed by planning Research Corporation, combines both an expert system, along with a statistical analysis component to monitor user behavior as well as detect known penetrations. Like MIDAS, the statistical component of ISOA develops both user and system profiles.

The main advantage of anomaly detection is that it provides a means of detecting intrusions, without *a priori* knowledge of the security flaws in the target system. The approach requires very few, if any, system dependent rules, making tools that employ the approach portable. However, despite wide acceptance of the approach, its operational effectiveness is less studied. The following is a brief list of some of the issues that may hinder the effectiveness of profile-based anomaly detection tools, along with techniques used to address these issues.

- *False positive rate*: For every anomaly that corresponds to a security violation, there may be a number of legitimate uses that could also cause the same anomaly. The higher the number of false positives, the less likely the SSO will be able to distinguish true penetrations from false alarms. As a result, profile-based anomaly detectors usually work on the basis of suspicion ratings rather than absolute identification of security violations. Anomaly detectors are also equipped with parameters to decrease their sensitivity to certain anomalies, while maintaining enough sensitivity to identify penetrations. This fine tuning of the system is a difficult process and perhaps one of its most ad hoc aspects.

- *False negative rate*: Not all violations will produce anomalous behavior. There might be penetration scenarios that could compromise a system, and yet produce no identifiable anomaly. Profile-based anomaly detection is not claimed to be a "catch-all" identification method for security violations, and is usually employed in conjunction with other intrusion detection approaches.

- *Gradual misbehavior*: Misfeasors may attempt to avoid detection by gradually training the detector to accept their misuse as normal behavior. To counter this, implementations such as IDES also develop group profiles to compare the actions of one user to what is expected from the user's group. The above argument holds only when the gradual misbehavior is distinguishable within a particular group. Otherwise gradual misbehavior will be an effective way to circumvent detection.

- *Sporadic user environments*: Not all environments are equally capable of supporting profile-based anomaly detection. Some environments, such as banking system, have users who perform the same tasks day after day. The amount of variation in a clerk's behavior, for instance, as well as the group behavior, will be very minor. Thus, profiles of normal behavior may be established very easily. In other environments however, such as a university computer network, anomalous behavior will be common among users. In such cases, the premise that the past audit logs would produce patterns of usage does not hold and the number of false positives may render an anomaly detector unusable.

5.2 Rule-Based Anomaly Detection

There are only minor differences between rule-based and statistical approaches to anomaly detection. The two approaches share the advantages and disadvantages mentioned before. The main difference is that instead of using statistical formulas, rule-based anomaly detectors use sets of rules to represent and store the usage patterns. Two example systems that employ rule-based anomaly detection are Wisdom and Sense (W&S) [22] and the Time-based Inductive Machine (TIM) [4] approach.

W&S, developed at Los Alamos National Laboratories, operates on a number of systems at both the operating system and application level, and can be used in real-time or batch-

mode. Rules can be generated automatically from the past audit data, as well as entered manually, to express policies and administrative data, together with expert penetration rules. Automated rule generation is done by a categorical examination of past audit data, from which a rule set describing past patterns is generated. The rules created by W&S are not only rules that are obvious and predictable by experts, but can at times be non-intuitive, representing subtleties of behavior patterns of users, programs, privileges, time slots, terminals, etc.

Advanced Security Audit trail Analysis on uniX (ASAX) [12] is an experimental system developed by the University of Namur and Siemens Nixdorf Software S.A. (Belgium). It aims at defining and implementing a commercial system for universal, efficient and powerful audit trail analysis corresponding to security level B3. It also provides socalled *format adaptors* to translate native audit trails from different systems into normalized format. ASAX processes security audit records represented internally in a single format called *Normalized Security Audit Data Format* (NADF). An implementation of ASAX for on-line analysis of multiple distributed data streams is under development.

Another implementation of a rule-based anomaly detection can be found in TIM. The main difference between TIM and the other anomaly detection systems is that TIM focuses on patterns of event sequences, whereas the others focus on individual events. Like W&S, rules are automatically generated through the analysis of the past audit data. By grouping event sequences into single rules, TIM is also able to reduce the amount of space required in the rule base by producing a rule entry for often-used multiple events.

5.3 Rule-Based Penetration Identification

A penetration identification system is essentially an expert system whose rules fire when audit records indicate suspicious, if not illegal, user activity. The rules may recognize single events that represent significant danger to the system, or recognize sequences of events that represent an entire penetration scenario.

Rule-based penetration identifiers have become a common supplemental component of IDS that also employ anomaly detection components. The expert rules offer the additional capability of identifying dubious behavior, even when the behavior appears to conform with the established patterns of use. Examples of IDS implementing expert systems include IDES, NADIR, W&S and USTAT [14]. Unlike anomaly detection systems, the rule-base of a penetration identifier is usually machine specific.

IDES includes a rule-base component which allows the representation of suspicious behavior based on the site-specific security policies, known security flaws and past intrusions. In IDES the anomaly detection component provides a way of learning and representing normal behavior, while the penetration identifier rule-base provides a way of representing improper behavior. The two components enable the tool to identify both masqueraders and misfeasors.

W&S and NADIR are two other examples of IDS combining anomaly detection with penetration identification. In the case of W&S, the anomaly detection component is also rule-based and the penetration detection component is integrated into the same rule-base that represents site-specific policies, expert penetration rules and other administrative data. NADIR's implementation is close to IDES in which it employs a statistical anomaly component in conjunction with a small rule-base of expert penetration rules. NADIR's rule-base consists of penetration rules that are developed by interviewing and working with security personnel.

One weakness in most of the current rule-based penetration detection implementations is their dependence on the representation of knowledge in the knowledge-base. Due to little flexibility in representing penetrations this way, a given penetration scenario may correspond to several audit record sequences that will produce slight variations of the same penetration. Thus, even if a scenario is represented in the rule-base, a minor variation to the sce-

nario might slip by unnoticed.

Moreover, in practice, penetration scenarios are not formulated and executed in terms of audit records and so, describing them in terms of rules that depend on audit records is not very efficient. Garvey and Lunt have proposed a new intrusion detection approach, called *Model-Based Intrusion Detection* [11], that overcomes some of the above-mentioned problems. Model-based intrusion detection provides an audit record independent technique for representing intrusive behavior in a knowledge-base which improves the system's flexibility, in identifying penetrators, and readability and generation of rules.

Another example of expert system IDS is USTAT, a State Transition Tool for UNIX. USTAT is a UNIX-specific implementation of a generic design called STAT [18] developed by Philip A. Porras. *State Transition Analysis* is another new approach to representing computer penetrations. Like Model-based intrusion detection, STAT also implements a high-level, audit record independent representation of penetrations in the rule-base. In STAT, a penetration is identified as a sequence of state changes that take the computer system from an initial state to a target compromised state. USTAT uses audit trails that are collected by the C2 Basic Security Module of SunOS, and keeps track of only those critical actions that must occur for the successful completion of the penetration. This approach differs from other rule-based penetration identification tools that use pattern-matching on the sequences of audit record.

5.4 Intrusion Prevention

Tools that aid in increasing the protection of computer systems are not limited to IDS. Attempts have also been made to develop tools for preventing intrusions.

Security Administrator Tool for Analyzing Networks, or SATAN [9], is a tool for investigating the vulnerabilities of remote systems. Systematically moving through a given Internet subdomain, it probes for weaknesses in each responding system. The vulnerabilities uncovered are then reported to the user. SATAN is the join work of Dan Farmer, author of COPS, and Wietse Venema, from the Eindhoven University of Technology in the Netherlands. SATAN was promoted as a security tool for system administrators, not an attack tool for crackers. Unfortunately, it can be utilized in either manner. It is up to system administrators to decide what its impact will be. The safety of any particular system depends on who utilizes SATAN first.

Searching for system vulnerabilities is not a new idea. COPS reports many common vulnerabilities on a single system, by analyzing the system it resides on. Computer Oracle Password and Security System (COPS) developed by Farmer and Spafford [8] is a notable prevention tool. COPS is designed to aid UNIX system administrators in testing the system configurations for common weaknesses. It consists of a collection of shell scripts and programs that search for a myriad of known security holes on UNIX systems. COPS reports to the administrators indicating often neglected or misunderstood critical areas. It may be used to help to educate novice administrators or detect mistakes made by experienced administrators.

Tools which probe for vulnerabilities on remote systems are not a new idea either. The Internet Security Scanner (ISS), written by Christopher Klaus, has been available in both public and commercial versions for several years. While it certainly simplified probing of remote systems, the public version was not particularly powerful. It lacked a user interface, and provided a limited set of attacks. In contrast, SATAN is considerably more powerful, and utilizes a World Wide Web (WWW) client to provide a friendly, point and click interface. Extensive information is provided that explains what vulnerabilities are being identified, and how those vulnerabilities can be removed.

CIAC has also recently developed a program to defend against SATAN and other similar tools [9]. The program called *Courtney* monitors the connections to the ports probed by SATAN. It basically monitors the network and identifies the source machines of SATAN probes/attacks. Courtney receives input from

tcpdump, counting the number of new services a machine originates within a certain time window. If one machine connects to numerous services within that time window, Courtney identifies that machine as a potential SATAN host.

Also another SATAN detector, *Gabriel*, has been released by Los Altos Technologies, Inc. Gabriel gives the system administrator an early warning of a possible network intrusion by detecting and identifying unauthorized network probing, and it confirms that authorized probing reaches the expected network segments. It identifies the source of probing and can immediately notify the administration via a pager, a phone call, an email message, or a screen display.

6 Summary

In this report we have surveyed various approaches to intrusion detection. Without preemption capabilities intrusion detection systems are neither cure nor defense against computer penetrations. At best they discourage intrusions, provide quick recognitions of security violations and aid in damage assessment. At worst they may provide a false sense of security.

Along with the approaches surveyed, several key definitions and assumptions that appear commonly throughout intrusion detection literature, were presented. Three classes of malicious users were identified: the masquerader, the misfeasor and the clandestine user. Profile-based anomaly detection appears to be the most effective approach for identifying masqueraders, whereas Rule-based penetration identification appears best suited to detect misfeasors. Detection of clandestine users, however, is beyond the scope of audit data analysis.

The areas that are less investigated and need future development include:

- testing an IDS and measuring its effectiveness;
- detecting intruders on distributed systems;
- determining the most appropriate technologies to use in specific situations; and
- examining the computer user acceptance of intrusion detection (privacy issues).

References

[1] D. Anderson and et. al. Next generation intrusion detection expert system (NIDES): User manual for security officer user interface (SOUI). Technical report, SRI International, 26 March 1993.

[2] D. S. Bauer and M. E. Koblentz. NIDX - an expert system for real-time network intrusion detection. In *Proceedings of the IEEE Computer Networking Symposium*, pages 98–106, 1988.

[3] A. Brignone. Fuzzy Sets: An answer to the evaluation of security systems? In *Proceedings of Fourth IFIP TCII International Conference on Comp. Sec. (IFIP/Sec'86)*, pages 143–151, Monte Carlo, Monaco, 2-4 Dec. 1986.

[4] K. Chen, S. C. Lu, and H. S. Teng. Adaptive real-time anomaly detection using inductively generated sequential patterns. In *Proceedings of the Fifth Intrusion Detection Workshop*, SRI International, Menlo Park, CA, May 10 - 11 1990.

[5] H. Debar, M. Becker, and D. Siboni. A neural network component for an intrusion detection system. In *proceddings of the 1992 IEEE Computer Society Symposium on Research in Security and Privacy*, pages 240–250, 4 - 6 May 1992.

[6] D. E. Denning. An intrusion-detection model. In *Proceedings of the 1986 IEEE Symposium on Security and Privacy*, pages 118–131, 7 - 9 April 1986.

[7] D. E. Denning and P. G. Neumann. Requirements and models for IDES - A real-time intrusion detection system. Technical report, CSL, SRI International, 1985.

[8] D. Farmer and E. H. Spafford. The COPS security checker system. In *Proceedings of the Summer 1990 Usenix Conference*, pages 305 –312, Anaheim, CA, June 1990.

[9] J. Fisher. A look at SATAN. U. S. DOE's Computer Incident Advisory Capability, CIAC NOTES, Number 95-07, March 29 1995.

[10] J. Frank. Artificial intelligence and intrusion detection: Current and future directions. In *Proceedings of 17th National Computer Security Conference*, volume 1, pages 22–33, Baltimore, Meryland, 11 - 14 oct 1994.

[11] T. D. Garvey and T. F. Lunt. Model-based intrusion detection. In *Proceedings of the 14th National Computer Security Conference*, Washington DC, October 1991.

[12] N. Habra, B. L. Charlierand A. Mounji, and I. Mathieu. ASAX: Software architecture and rule-based language for universal audit trail analysis. In *Proceedings of ESORICS'92, European Symposium on Research in Computer Security*. Springer-Verlag 1992, 23 - 25 November Toulouse 1992.

[13] B. Hubbards, T. Haley, N. McAuliffe, L. Schaefer, N. Kelem, D. Walcott, R. Feiertag, and M. Schaefer. Computer system intrusion detection. Technical Report RADC-TR-90-413, Final Technical Report, Trusted Information Systems, Inc., Dec. 1990.

[14] K. Ilgun. Ustat: A real-time intrusion detection system for unix. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 16–28, May 24-26 1993.

[15] H. Javitz and A. Valdes. The NIDES statistical component description and justification. Technical Report Annual Report, A010, SRI Project 3131, SRI International, 7 March 1994.

[16] F. T. Lunt. Automated audit trail analysis and intrusion detection: A survey. In *Proceedings of the 11th National Computer Security conference*, pages 59–66, Baltimore, MD, Oct. 1988.

[17] T. F. Lunt. IDES: An intelligent system for detecting intruders. In *Proceedings of the Symposium: Computer Security, Threat and Countermeasures*, Rome, Italy, November 1990.

[18] P. A. Porras and R. A. Kemmerer. Penetration state transition analysis a rule-based intrusion detection approach. In *Proceedings of the Eight Annual Computer Security Applications*, pages 220–229. IEEE Comp. Soc. Press, 30 Nov. - 4 Dec. 1992.

[19] S. I. Schaen and B. W. McKenney. Research, standards, and policy directions for network auditing. In *Proceedings of the Fifth Intrusion Detection Workshop*, SRI International, Menlo Park, CA, May 10 - 11 1990.

[20] M. M. Sebring, E. Shellhouse, M. E. Hanna, and R. A. Whitehurst. Expert systems in intrusion detection: A case study. In *Proceedings of the 11th National Computer Security Conference*, pages 74–81, October 1988.

[21] S. E. Smaha. Haystack: An intrusion detection system. In *Proceedings of the IEEE Fourth Aerospace Computer Security Applications Conference*, pages 37–44, Orlando, FL, December 1988.

[22] H. S. Vaccaro and G. E. Liepins. Detection of anomalous computer session activity. In *Proceedings of 1989 IEEE Computer Society Symposium on Security and Privacy*, pages 280–289, Oakland, California, 1-3 May 1989.

[23] J. R. Winkler. A UNIX prototype for intrusion and anomaly detection in secure networks. In *Proceedings of the 13th National Computer Security Conference*, pages 115 – 124, Baltimore, MD, Oct. 1990.