Networked System Intrusion Detection Models

B. C. Soh P. White
School of Computer Science & Computer Engineering, and
Applied Computing Research Institute
La Trobe University, Melbourne, Bundoora 3083, Australia

1 Introduction

The basic hypothesis of intrusion-detection systems (IDS) is that unauthorised use of a computer system is always accompanied by anomalous behaviour. Anomalous behaviour can be explained in terms of departures from established patterns of use. Consequently, the analysis of computer usage makes it possible to detect intrusive activity. The audit records collected by a computer system can provide the information needed to detect anomalous behaviour, which indicates unauthorised use. For example, an abnormally high rate of password failures may imply an attempted break-in or a legitimate user routing information to unusual printers may attempt to leak sensitive documents.

This paper investigates several models in relation to a intrusion-safe approach to system security. In Section 2 the intrusion-safe and intrusion-avoidance approaches are discussed. Section 3 deals with the relationship between intrusion detection and security requirements, while Section 4 defines host-based and network-based intrusion-detection systems. Intrusion-safe models and their implementations are respectively given in Section 5 and Section 6. Section 7 concludes the paper.

2 Intrusion-Safe vs Intrusion-Avoidance

Intrusion-safe (or intrusion-tolerance) approach [1] and hence intrusion detection contribute a different notion of security to previous work done in the field. Conventional security approaches, such as access controls and authentication, involve building a shell around the computer or network [3] and are actually of a intrusion-avoidance type, which aims to make a system 100% secure. In contrast, the principle of an intrusion-safe approach is not to protect a system's vulnerabilities but to monitor attempts to exploit them. In this way, an IDS can overcome some of the disadvantages of purely utilizing conventional approaches to securing a system. The limitations of the intrusion-avoidance approach includes:

- It is very difficult, if possible at all, to build a 100% secure system [2, 3]. The problem is the same in nature as producing an error-free program. The cost to produce a totally secure system would be prohibitive (given that information security is a decision based upon the information assets of an organisation). As Denning [2] points out that existing systems contain security vulnerabilities and even patching these might prove too costly or technically difficult. Furthermore secure systems will still be susceptible to the way they are used. Incorrect usage or configuration may result in security flaws.
- A large amount of effort and cost has been invested in current computer systems and networks. While most are not highly secure, it seems impractical to assume that the vast existing infrastructure of possibly insecure computer and network systems will be scrapped in favour of new, secure systems [3].
- Cryptographically protected systems are vulnerable to poor or cracked passwords, and stolen keys.
- Currently utilised systems may include attractive features missing in more secure alternatives. An example of this would be UNIX, one of the original purposes of the operating system was to facilitate program development. A more secure system utilising extensive "preventive-based' controls may not allow the flexibility of UNIX.
- Secure computer systems are still vulnerable to abuse from privileged or authorised users.

The use of intrusion-safe approach can alleviate the above outlined problems. Utilisation of the approach means that a computing environment need not be realised by an expensive secure system. Of course an integrated approach of intrusion-safe and intrusion-avoidance would be the best option for making a system more secure. In this way, even unsuccessful attempts to penetrate a system can be detected and dealt with. A special advantage of automated intrusion-detection systems is that they have the ability to learn.

3 Relationship between Intrusion Detection and Security Requirements

The intrusion-safe approach can be applied to the following 3 security requirements:

- Availability: An intrusion-detection system should be able to monitor aspects of a computer system that is critical to the availability of the system.
- Integrity: By monitoring specified objects, security against modification can be increased.
- Confidentiality: Monitoring of all disclosures of information (eg printing) can help to detect any breaches in the confidentiality of the system.

4 Host-Based or Network-Based

Many of the papers in the literature, in particular [3], use 'where an intrusion-detection system should monitor' as the foremost classification: network-based or host-based. If the intrusion-detection monitors the audit records generated by an operating system, it is a host-based system. Even if it were to run over a number of systems it remains in the same category. This is for 2 reasons: (i) network traffic is not analysed; (ii) these systems were originally designed for a single host and they transfer the audit data from the monitored hosts to a central host for processing. On the other hand, a network-based intrusion-detection system carries out a network traffic analysis. The analysis of network resources usage means to detect suspicious network usage patterns (primarily in traffic).

Extensions of single host-based systems have shown the incorporation of many of the previously network-based features. Intrusion-detection systems that have progressed in this manner include the IDES/NIDES [5], which monitor the transactions on the network with an ability to detect intrusive behaviour across the network. Another development is the extension of the network-based system Network Security Monitor [6] to have host monitoring capabilities, as in the DIDS [4].

5 Intrusion-Safe Models

The "model of intrusion" [2] that an intrusion-detection system is based on is more important than the host/network aspect. Here, a couple of questions are posed:

- Should an intrusion-detection system attempt to detect intrusion by comparing current behaviour with a profile of system users' normal behaviour?
- Should an intrusion detection monitor users' commands in an attempt to detect the sequence of actions that leads to a penetration of security?

Basically, the intrusion-safe approach possesses the following features:

- Audit reduction and threshold detection
- Anomaly detection
- Intrusion-identification (or misuse detection).

5.1 Audit Reduction and Threshold Detection Model

Threshold detection and audit reduction provide a basic level of reporting in a computer system environment. The features of audit reduction and threshold detection are in contrast to the higher level of analysis carried out by anomaly detection and intrusion identification, both of which do not carry out any analysis on the audit information but merely reorganise the data into a more manageable format [7] and produce summary statistics concerning computer operations.

The audit reduction systems process the audit information and prune them of extraneous data. The remaining information is the audit trails pertaining to certain events which warrant further investigation. Often the reduction merely means that unneeded fields of auditing information are discarded and that the reorganisation of the data into a format results in making further analysis easier.

A threshold detection system maintains summary statistics of the audit trails. This is carried out by recording the number of occurrences of specified events. When the number of times that the event occurs exceeds the previously defined threshold, intrusive behaviour may be occurring. The threshold is designed to capture the level at which an operation becomes suspicious over a time period. The threshold detector should monitor events upon which a large number of occurrences in a relatively short period of time may indicate the presence of an intruder.

Events extracted by an audit reduction system and monitored by a threshold detection system commonly include: login attempts and failures, unusually large data transfers or unusual time of the day logins take place, which may suggest a penetration is attempted or underway.

The ability of threshold analysis to detect penetrations alone is deemed poor by Ilgun et al in [10, 11]. For this reason, it is unlikely that an intrusion-detection system will rely solely on threshold detection and audit reduction. The use of audit reduction still leaves much work in analysis to an already busy system security officer. This work could be better handled by an intrusion-detection system which is able to carry out a more intelligent analysis on the audit data.

5.2 Anomaly Detection Model

Anomaly detection attempts to identify variations of current usage from historical patterns of usage. Historical patterns of usage are developed from audit trails collected over time. The historical pattern of usage is termed a profile [2, 8]. It is the profile that is used as a user's behavioural norm to compare and contrast with the present behaviour. There are a number of advantages and limitations for the anomaly detection technique.

The major advantage of the anomaly detection technique is that knowledge about the monitored system is not needed. This is an important feature of the anomaly detection technique. The reason why many systems are insecure is that high level of understanding of system software and hardware architectures is needed to design a secure system (but security flaws are often obscure). The approach of anomaly detection demonstrates a desirable quality of an intrusion-detection system: it is system independent. System independence allows the intrusion-detection system to be portable, aside from implementation-language considerations.

The basic challenge to ensure that anomaly detection is feasible involves handling the data which represents the past behaviour of users. The amount of data concerning user actions is substantial, especially since the purpose to gather a representative sample of behaviour is aided by a longer period of collection. The information must be captured in a manner that is not voluminous and able to be used by the system easily. The data can be summarised into an abstract form. However, it is important that no information about the behaviour is lost during this abstraction, which would otherwise precipitate a non-representative abstract.

Anomaly detection systems also have other constraints on their effectiveness, including (see [8] for details):

- The development of normal usage patterns may not be possible. Some users may use the computer system in a wide variety of authorised tasks, or a user may be very intermittent in his/her use of the computer. This problem can be influenced by the variance of the sensitivity of the intrusion detection to certain events.
- The settings and sensitivity of thresholds can also be a problem. For example, new users may make more mistakes logging in than more experienced ones: decreasing the sensitivity of failed login attempts for new users may reduce false alarms and hopefully not true alarms. Obviously the intrusion-detection system needs to be customable. This customarisation, and the need for it, leads to the difficulty of threshold setting. If a threshold is set too low the number of false alarms would be large. On the other hand, if it is set too high a penetration may go undetected. Work has been done on threshold setting in [9, 10, 1].
- Some cases of intrusive behaviour may result in little or no observable anomalous behaviour, especially at a level below the sensitivity of the intrusion-detection system. This leads to another concept of intrusion identification or misuse detection, which is the subject of the following subsection.

5.3 Intrusion-Identification Model

In the intrusion-identification approach, the system monitors a user's actions to determine attempts of exploiting known security flaws. The user's actions are compared to known 'attack signatures" or 'known attack methods'. A match means that the user has used the computer in the exact manner required to exploit a security flaw.

An attempt to exploit a system security flaw often constitutes a series of actions so the intrusion-detection system must be able to deal with multiple actions. Another attempt may include multiple users, working in unison to compromise system security. The exact method to achieve this varies; but there is one common point: all of the intrusion-identification systems could not be termed system independent. They rely on the specification of system specific security flaws and the way these flaws may be exploited. Because such knowledge is specific to an operating environment, diligence needs to be applied to ensure the system is up to date with system security holes.

The techniques used to implement an identification system involves the applications of artificial intelligence. Rule-based expert systems have become a common supplement to anomaly detection components of an intrusion-detection system. Systems using such approach can be found in [11, 12, 2, 10, 5].

The advantages of intrusion identification are its ability to detect intrusive behaviour generating an insufficient level of anomalous behaviour, which would otherwise be undetectable in a anomaly detection system. Environments in which a profile of users behaviour is difficult to be developed also benefit from the use of the intrusion-identification technique.

However, there exist limitations of the current examples of intrusion-identification systems. These limitations are not inherent in the concept of intrusion identification itself but involve issues of the artificial intelligence techniques currently used. The limitations are detailed in the next section.

6 Implementations of the Intrusion-Safe Models

The actual implementations of the intrusion-safe models bear upon their effectiveness and limitations. While broad advantages and disadvantages of the implementations can be raised, it is difficult to quantify how this relates to the ability of a particular implementation to detect intrusive behaviour. It is in this area that a comparative study of the implementations is required.

6.1 Anomaly Detection Technique

There exist two major approaches to detect anomalous behaviour: (i) statistical methods, and (ii) artificial intelligence techniques. Statistics are applied to the problem of recognition of variations in users behaviour by Denning [2] and others [5]. Artificial intelligence related techniques, eg of rule-based expert systems, are used in the Wisdom & Sense (W&S) [13] system and the Time-based Inductive Machine (TIM) [14]. Another artificial intelligence related approach to anomaly detection, ie neural network-based techniques, is suggested in [15, 16].

6.1.1 Statistical Methods

Here, the historical patterns of usage are formed into an activity profile, which "characterises the behaviour of a given subject (or set of subjects) with respect to a given object (or set of objects), thereby serving as a signature or description of normal activity for the respective subject(s)". The behaviour that is to be monitored (eg login frequencies and read failures) is described in terms of a statistical model and metric. Observed behaviour, obtained from audit records, is compared to the relevant activity profiles to determine whether the current behaviour can be classified as anomalous.

The profile metric describes what the observed behaviour x, a random variable, represents. There are 3 types of representations. First, an event-counter x counts the number of times an event occurs in a time period (eg number of password failures per minute). Second, an interval-time x is the length of time between two related events (eg logins and logouts). Third, a resource-measure x is the amount of resource used by a subject/user during a time period (eg CPU time used).

There are several statistical methods that can be used to determine if a new observed x_n is abnormal with respect to previous observations of x $(x_1, ..., x_n - 1)$, including:

- Operational method: this method compares the new x to a fixed threshold limit
- Mean and standard deviation method: the new x is compared to the longer-term derived mean and standard deviation. If the new observation is outside a given confidence interval, the behaviour is classified as anomalous.
- Multi-variate method: this is similar to the mean and standard deviation method except the correlations between two or more metrics are taken into consideration. The usefulness of this method depends on whether any relationship exists between the metrics in question.
- Markovian method: this method only applies to event counter metrics. If the probability to change from one state to another is relative small, then that transition is deemed anomalous.

If a newly observed sample of a metric is considered anomalous, a certain system activity must take place: eg to record the behaviour or perhaps to take further action. Otherwise, the sample must be incorporated into the model's long-term history. It is in this way that the system is able to dynamically adjust its own parameters and thresholds.

6.1.2 Rule-Based Anomaly Detection Methods

Here, historical patterns of usage are represented as rules in a rule-base. Any deviation from the rules may indicate anomalous behaviour.

In Wisdom & Sense (W&S), a large forest of rules representing common sequences of events are created. The rules are those of an expert system, with the branches of the tree representing normal patterns of usage. W&S builds one tree for all users and hence lacks sensitivity to tailor the system for individual users (or sets of users).

In the Time-based Inductive Machine (TIM), sequential rules that characterise users behaviour over time are created. The rule-base stores the usage patterns. The rules could be based on sequential relationships between audit records or the temporal records. The rules can be generated by the TIM system itself or supplied manually. The rules generated are then inductively modified.

A newer artificial intelligence related technique for anomaly detection is neural network. A neural network is a system which transforms an input to an output by the action of a large set of simple, highly connected elements. The transformation function is determined by the strength of the connections between the elements. The strength is in turn determined by the net learning the correct weights from supplied sample pairs of audit records and correct classifications. The net would then be trained to recognise the patterns of users behaviour. So, when a user's actual behaviour differs from the pattern an anomaly is detected. The benefits of using neural networks for intrusion detection include: avoiding the high cost of new algorithm development; avoiding assumptions about statistical distributions; and being scalable to environments of thousands of users.

6.2 Intrusion-Identification Method

As mentioned previously, intrusion-identification systems utilise known scenarios of intrusion to identify attempts to compromise the security of a system. This approach relies on already known security flaws.

6.2.1 Intrusion Identification using a Rule-Based Expert System

The expert system approach comprises a rule-based and an inference engine. The rule-base contains rules which fire (are activated) when parsed audit records contain suspicious activity. The rule may contain a single user's action or a whole sequence representing a known intrusion attack.

An example system that has pioneered the work in using expert systems for intrusion identification is IDES [5]. IDES has a rule-based component that monitors for exploitation of known security flaws and site-specific security policies. IDES evaluates audit records when they are produced. The expert system component attempts to match fields of the audit records to a rule or a portion of a rule in the rule-base. If a match exists, the rule is fired and the suspicion rating of the user is increased. When the suspicion level increases beyond a threshold, action is taken, be it an alarm or report generation.

6.2.2 Intrusion Identification from Network Monitoring

Quite a different approach to intrusion identification is offered by the Network Security Monitor (NSM) [6]. Firstly, the source of the information is not from the host-based audit trails but from information collected in the network, which includes the type of service, the connection ID, source host and destination host. These four attributes of any network dialogue make up a four-dimensional matrix called Access Control Matrix (ACM). Each cell in the matrix represents a unique network connection and contains two values: the number of packets sent and the sum of the data sent. Data patterns in the current ACM are compared to a masking matrix holding a certain pattern. For example, the ACM is compared to another matrix containing the representation of a specific attack. Another method of detecting possible intrusions is to apply a set of rules against the ACM.

Advantages of this intrusion-identification approach are associated with network attack-handling capabilities and performance considerations. Firstly, the earlier host-based systems have not considered the methods of network intrusion. Secondly, the collection of audit records and the processing required can degrade system performance of the host. The NSM avoids this by operating on a node of the network, which monitors all traffic.

7 Conclusion

The features of an intrusion-safe system and the methods to implement these features were investigated in the paper. The discussions point to the direction for future work of using cognitive modelling approach to intrusion detection.

References

[1] B.C. Soh and T.S. Dillon, "Setting optimal intrusion-detection thresholds," J. of Computers & Security, vol. 14, pp. 621-631, 1995.

[2] D.E. Denning, "An intrusion-detection model," IEEE Transactions on Software Engineering, vol. SE-13, Feb 1987, pp. 222-231.

[3] D. Farmer and E.H. Spafford, "The COPS security checker system," in Proc. of Summer Usenix Conference, Anaheim, Jun 1990, pp. 305-312.

[4] B. Mukherjee, "Network Intrusion Detection," IEEE Network, vol. 8, May/Jun 1994, pp.26-41.

[5] T.F. Lunt et al, A Real-Time Intrusion-Detection Expert System (IDES), Final Report, Technical Report, Computer Science Laboratory, SRI International, Feb 1992.

[6] L.T. Heberlein, G.V. Dias, K.N. Levitt, B. Mukherjee, J. Wood, and D. Wolber, "A netowrk security monitor," in Proc. 1990 IEEE Symp. on Research in Security and Privacy, Oakland, CA, pp. 296-303, May 1990.

[7] G.F. Luger and W.A. Stubblefield, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Readwood City, Benjamin/Cummings, 1993.

[8] P. Anderson, Computer Security Threat Monitoring and Surveillance, Fort Washington, James P. Anderson Co., April 1980.

[9] H.S. Javitz and A. Valdes, "The SRI IDES statistical anomaly detector," in Proc. of the IEEE Symp. on Research in Security and Privacy, Oakland, May 1991, pp.316-376.

[10] K. Ilgun, R.A. Kemmerer and P.A. Poras, "State transition analysis: a rule-based intrusion detection approach," IEEE Transactions on Software Engineering, vol. SE-21, March 1995, pp. 181-199.

[11] K. Ilgun, "USTAT: a real-time intrusion-detection system for UNIX," in Proc. of the IEEE Symp. on Security and Privacy, Oakland, May 1993, pp. 16-28.

[12] H.S. Vaccaro and G.E. Liepins, "Detection of anomalous computer session activity," in Proc. of the IEEE Sypm. on Security and Privacy, Oakland, May 1989, pp. 280-289.

[13] W. Shieh and V.D. Gligor, "A pattern-oriented intrusion detection model and its application," in Proc. of the IEEE Symp. on Security and Privacy, Oakland, May 1991, pp. 327-342.

[14] D.E. Denning and P.G. Neumann, "Requirements and model for IDES-A real-time intrusion detection system," Technical Report, Computer Science Laboratory, SRI International, Aug 1985.

[15] H. Debar, M. Becker and D. Siboni, "A neural network component for an intrusion detection system," in Proc. of the IEEE Symp. on Security and Privacy, Oakland, May 1992, pp. 240-248.

[16] T.F. Lunt, "Real-time intrusion detection," in Proc. of COMPCON Spring '89, March 1989.