Computers & Security, 20 (2001) 676-683

A comparison of Intrusion Detection systems

E. Biermann

Dept of Computer Technology, Technikon PTA, Private Bag X680, Pretoria, South Africa

E.Cloete

Dept of Computer Science & Information Systems, P.O. Box 392, UNISA, 0003, South Africa

L.M. Venter

School of Modelling Sciences, PU vir CHO (Vaal Triangle Campus), South Africa.

Abstract

A computer system intrusion is seen as any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource.[1] The introduction of networks and the Internet caused great concern about the protection of sensitive information and have resulted in many computer security research efforts during the past few years. Although preventative techniques such as access control and authentication attempt to prevent intruders, these can fail, and as a second line of defence, intrusion detection has been introduced. Intrusion detection systems (IDS) are implemented to detect an intrusion as it occurs, and to execute countermeasures when detected.

Usually, a security administrator has difficulty in selecting an IDS approach for his unique set-up. In this Report, different approaches to intrusion detection systems are compared, to supply a norm for the best-fit system. The results would assist in the selection of a single appropriate intrusion detection system or combine approaches that best fit any unique computer system.

Keywords

Intrusion detection systems, security, anomaly detection, misuse detection, security models.

Introduction

Security techniques such as authentication[2] and access control[3] have been developed to achieve the objective of computer security — namely to prevent unauthorised intruders from accessing and manipulating information. These prevention systems are implemented as a first line of defence. The advantages of the Internet, namely the availability and amount of information, also pose the largest threat to information security. Anderson[4] introduced the concept of an intrusion detection system (IDS) as a second line of defence.

An IDS is a security technology attempting to identify and isolate computer systems intrusions. Mainly two techniques, namely anomaly detection and misuse detection, have been identified since the introduction of this field. Different approaches within each technique exist. The security administrator is now faced with the problem of selecting a suitable IDS for his/her particular computer system. The choice is complicated by the availability of this myriad of available techniques.

We approach the problem by first defining the characteristics that have to be displayed by an IDS to function optimally as a security system. A part of the problem is to select criteria to compare the different

approaches. The defined characteristics are put through a selection process deciding on the most important criteria that will best perform this task. These are then used to compare and evaluate the different IDS approaches.

Theoretical Background

One of the main approaches of IDS, namely anomaly detection is based on the assumption that an attack on a computer system will be noticeably different from normal system activity, and an intruder will exhibit a pattern of behaviour different from that of the normal user.[5] In the second leading approach, misuse detection, a collection of known intrusion techniques is kept in a knowledge base, and intrusions are detected by searching through the knowledge base for the same techniques.[5]

Two critical errors can be generated by an IDS namely false positive errors and false negative errors. A false negative error is intrusive behaviour defined by the IDS as normal user behaviour while a false positive error is legitimate user behaviour that is regarded by the IDS as intrusive behaviour.

Security models

We focus on the two primary security models defined in the past few years namely the IDES and the CIDF models. The IDES model[6] forms the basis of many IDS utilities and products. The model includes five components:

1. Subjects and objects: Subjects are the active initiators of operations that are audited while objects are the information repositories on which the subjects perform their actions or operations.
2. Audit records: In order for an IDS to work in a practical setting, the various types of information and their positions in the audit record must be known in advance so that information is processed properly by the intrusion detection mechanism.
3. Profiles are used to characterise expected normal behaviour on a computer system. Typical types of information in these profiles are login activity and file access.
4. Anomaly records: Alarms that are created whenever certain behaviour that is observed does not match the profiles.
5. Activity rules: These are programs that describe what action should be taken when an alarm is triggered.

This model make provision for provides for the development of statistical information on normal behaviour, so that abnormal behaviour or illegal behaviour can be identified and reported [3].

The CIDF model defines a set of components that describes an IDS:

- E-boxes/Event generators: The purpose of an E-box is to provide information about events to the rest of the system;
- A-boxes/Analysis engines: The purpose of the A-box is to analyse input from event generators;
- D-boxes/Storage mechanisms: The D-box component of an IDS defines the means used to store the security information produced by the A-boxes and E-boxes, and makes it available at a later time;
- C-boxes/Countermeasures: Most commercially available intrusion detection systems are equipped with some form of countermeasure capability, ranging from shutting down TCP connections to modifying router filter lists.

Approaches to Intrusion Detection Systems

Table 1 and Table 2 list the different approaches to anomaly and misuse detection respectively. These approaches were used in the comparison.

Comparison Criteria

Parker[7] outlines six elements of security to be addressed by an administrator when either securing a computer system or when designing an IDS.

1. Availability: The computer system and especially critical data must at all times be available for use

Approach	Description
Statistical	These approaches define normal or expected behaviour by collecting data relating to the behaviour of legitimate users over a period of time. Statistical tests are then applied to the observed behaviour to determine the legitimacy of the behaviour.
Predictive pattern generation	In this method future events are predicted based on events that have already occurred. Rules, generated by the IDS, define the probability that a certain event will occur. A rule consists of a left-hand side, defining two concurrent events, and a right-hand side, providing the probability of a specific event following the events defined on the left- hand side of the rule. An event is flagged as intrusive if the left-hand side of a rule is matched, but the right hand side is statistically deviant from the prediction.
Neural networks	These systems learn to predict the next command based on a sequence of previous commands by a specific user. Building a neural network an IDS consists of three phases: 1. The collection of training data by obtaining the audit logs for each user for a certain period. A vector is formed for each day and each user, which shows how often the user, executed each command. 2. Train the neural network to identify the user based on the command distribution vectors. 3. The neural network to identify the user based on the command distribution vector. If the network's suggestion is different from the actual user, an anomaly is signalled.
Sequence matching and learning	Lane & Brodley[8] introduced an application of machine learning to anomaly detection. This approach uses the hypothesis that a user responds in a predictable manner to similar situations, which leads to repeated sequences of actions. To form a user profile their approach learns characteristic sequences of actions generated by users. The differences in characteristic sequences are used to distinguish a valid user from an intruder masquerading as that specific user.

Table 1: Approaches to anomaly detection

when needed by authorised users. Service supplied by the computer system must not be denied to authorised users;
2. Utility: The computer system and data on the computer system must be useful for a specific purpose;
3. Integrity: The computer system and the data on the computer system must be complete and in a readable condition. Information and programs on the computer system must be changed only in a specified and authorized manner;
4. Authenticity: The computer system must be able to verify the identity of the users, and the users should be able to verify the identity of the computer system;
5. Confidentiality: The information in a computer system and transmitted information must be accessible only to authorized users;

Approach	Description
Expert systems	An expert system encodes knowledge about past intrusions, known system vulnerabilities and the security policy. As information is gathered, the expert system determines whether any rules have been satisfied.[9]
Keystroke monitoring	Keystroke monitoring is the process used to view or record both the keystrokes entered by a computer user and the response of the computer during an interactive session.
Model-based	In this approach, known intrusion attempts are modelled as sequences of user behaviour, these behaviours are then modelled as events in an audit trail. The IDS is responsible for determining how identified user behaviour is manifested in an audit trail.
State transition analysis	The monitored computer system can be represented as a state transition diagram which is a graphical representation of the actions performed by an intruder to archive a system compromise. In state transition analysis, an intrusion is viewed as a sequence of actions performed by an intruder that leads from some initial state on a computer system to a target compromised state. State transition analysis diagrams (the graphical representations of state transition analysis) identify the requirements and the compromise of the penetration. They also list the key actions that have to occur for the successful completion of an intrusion.
Pattern matching	The basis of this model is the encoding of known intrusion signatures as patterns that are matched against the audit data. It attempts to match incoming events to the patterns representing intrusion scenarios. This model is based on the notion of an event, which consists of monitored changes in the state of the system, or part of the system, or part of the system. It can represent a single action by a specific user on a system, or an action by the system, or it can represent a series of actions resulting in a single, observable record.

Table 2: Approaches to misuse detection

6. Possession: The owners of the computer system must be able to control the system. If control is lost, it affects all the users authorized to work on the system.

We used above elements of security to select the different criteria by which the approaches were compared. Table 3 lists the different criteria that we defined and used to compare the different approaches to intrusion detection systems.

Results and Conclusions

By studying tables 4 and 5, that compare the different approaches to intrusion detection systems, it becomes apparent that no single approach can detect all types of intrusions. In an environment where security is of the utmost importance, the ideal is to combine a specific anomaly approach with a specific misuse approach into one IDS. If the security administrator needs known as well as unknown attacks to be detected, an anomaly approach has to be used. The downside is that anomaly approaches have accuracy and low completeness rates. Misuse detection approaches, on the other hand, detect only known attack patterns with high accuracy.

A major problem with current approaches to anomaly detection is that it is difficult to define normal user

Criterion	Evaluation Elements
Data	Type of data Amount of data Origin of data
Detection Range	Accuracy Completeness Known attacks Masquerade attacks Denial of service Malicious use Leakage Attempted break-ins Penetration of security control systems
Resources	Overhead
Network	Network-based or not Portability
System Architecture	Methods of detection Real-time operation Human supervision Manipulation level Behaviour modelling Attack resistance
Alarm	Countermeasure activities Detection time
System Change	User behaviour Sensitivity levels Expanding system Knowledge base

Table 3: Comparison Criteria

behaviour. In a dynamic environment it will be almost impossible to create user profiles that determine the normal behaviour. In these cases it would be better to look at intrusion detection systems that observe the behaviour of processes rather than users. For example, more recently Forrest et al.[10] and Lee et al.[11] have tried instead to determine the normal behaviour for privileged processes (those that run at root), rather than to determine the normal behaviour of users or groups of users. Another solution could be to create profiles for groups of users adding to the single user profiles.

Current intrusion detection systems operate at high level of data manipulation and are ineffective for detecting intrusions that can occur at a low network level. If the IDS is installed on a single host computer, problems can arise when trying to detect intrusions on a large network. In this case, the IDS rather has to be implemented on a number of host computer systems. Problems can also arise if the IDS is implemented on a number of hosts computers. For example: if different hosts were used, the audit data would be in different formats and would be transmitted between the different hosts, which would cause the integrity of the data to be questioned. The only two approaches that are able to detect intrusions on a network level are state transition analysis and pattern matching. If an IDS needs to be expanded or transferred to another computer system, then it would be easier to make use of misuse detection approaches. It is a difficult process to expand or change the sensitivity levels of anomaly approaches.

All the intrusion detection approaches use input in the form of audit data created by the operating system. These audit data proved in most cases to be problematic; special programs such as data mining are needed to subtract meaningful data from the records.

All the current approaches except model-based detection and state transition analysis examine large quantities of data in order to detect intrusive behaviour. These examinations can cause extra overhead on the computer system that the IDS is being installed on. The amount of data taken as input by the IDS needs to be analysed as the amount of data has direct impact on the time taken to detect an intrusion.

State transition analysis is designed to detect intrusions in a high-security risk environment and can foresee impending intrusions. State transition analysis as well as expert systems are not feasible on small computers due to being resource intensive.

Approach	A/M	Type of data	Amount of data	Origin of data
Statistical	A	Audit data	All incoming audit data	Operating system
		User profiles	Large amount of audit data to form detailed profiles	Created by designer
Predictive pattern	A	Audit data	All incoming audit data	Operating system
		Rule base	All possible event sequences	Created by designer
Neural networks	A	Sequence of commands	All commands	Operating system
		Audit records	All the audit logs for each user	Operating system
Sequence matching and learning	A	Set of behavioural sequences to form user profiles	All sequences of actions generated by users	Operating system
		Audit records	All incoming audit data	Operating system
Expert systems	M	Audit data	All incoming audit data	Operating system
		Knowledge base of known intrusions	Large amount	Created by expert
Keystroke monitoring	M	Knowledge base of known intrusions	All intrusions	Created by designer
		Keystrokes	All keystrokes	Special programs to capture keystrokes
Model-based	M	Audit data	Only portion of audit data related only to intrusion are examined	Operating system
		Knowledge base of known attack scenarios	All attack scenarios	Created by designer
State Transitional analysis	M	State transitional diagram of known attack patterns	Key actions for each intrusion	Created by designer
		Audit data	All attack scenarios	Operating system
Pattern matching	M	Audit data	All incoming audit data	Operating system
		Patterns of attack	All attack scenarios	Created by designer

Table 4: Comparison of data criteria - A/M - Anomaly / Misuse

Approach	*A/M*	*Accuracy*	*Completeness*	*Known attacks*	*Unknown attacks*	*Masquerade*	*Denial of service*	*Malicious use*	*Leakage*	*Attempted breakins*	*Penetration of security*
Statistical	A	Low if threshold too low	Low if threshold is high	Yes	Yes	Yes	No	Yes	No	--	No
Predictive pattern generation	A	Low if event sequence is not listed	Low if event sequence is not listed	Yes	Yes	Yes	No	Yes	No	--	Yes
Neural networks	A	Low if window is small	Low if window is large	Yes	Yes	Yes	No	Yes	No	--	No
Sequence matching and learning	A	Low	Hi	Yes	Yes	Yes	No	Yes	No	--	No
Expert systems	M	High	Low	Yes	No	No	No	Yes	No	--	Yes
Model-based	M	High	Low	Yes	No	No	No	Yes	No	--	
State transition analysis	M	High	Low	Yes	No	No	No	Yes	No	No	Yes
Pattern matching	M	High	Low	Yes	No	No	No	Yes	No	Yes	Yes

Table 5: Comparison of detection range criteria A/M - Anomaly / Misuse

In the case of anomaly approaches, user behaviour can fluctuate or gradually change, but still be normal behaviour. An advantage of current intrusion detection systems is that they are normally highly adaptive to such changes. In the case of misuse detection, the knowledge base needs to be updated regularly in order to add new intrusion scenarios. This updating has to be done by experts or the designers of the system. It is done manually and puts an extra workload on the security administrator. Research still has to be done in order for the IDS to update the knowledge base itself.

One of the functions of an IDS is to sound an alarm if an intrusion is recognised, and to provide some countermeasure activity. All the current approaches can detect some types of intrusions in real time, but the ability to stop the intrusion is still an open field. The ability to detect intrusions in real time also affects the performance of monitored computer system. On-line intrusion detection systems are computationally very expansible because they require continuos monitoring. Decisions need to be made quickly with less data and therefore they are not as reliable. Batch mode audit analysis has a key advantage: analysis can be done during low periods of central processing unit usage and/or at another computing facility.

The field of IDS needs to be researched and developed more intensively in order to provide a hundred percent secure system that can detect all types of intrusions in real time, without creating false alarms and without any human supervision.

Bibliography

[1] Heady, R., Luger, G., Maccabe, A. & Servilla, M. 1990. The architecture of a network level network intrusion detection system. Technical report CS90-20, Department of Computer Science, University of New Mexico.

[2] Russel, D. & Gangemi, G.T. 1992. Computer security basics. CA: O=Reilly & Associates Inc. 448p.
[3] Caelli, W., Dennis, L. & Shain, M. 1994. Information Security Handbook. First edition. Wilthire: Macmillan Press Ltd. 833p.
[4] Anderson, J.P. 1980. Computer Threat Monitoring and Surveillance. (In Anderson, J.P. Technical report, Fort Washington, Pennsylvania.)
[5] Sundaram, A. 1996. An introduction to intrusion detection. Crossroads: The ACM student magazine, 2(4), April.
[6] Denning, D.E. 1987. An Intrusion-Detection Model. IEEE Transactions on software Engineering, 13(2):222-232, Feb.
[7] Parker, D.B. 1994. Demonstrating the elements of information security with threats. (In Proceedings of the 17th National Computer Security Conference, pages 421-430.)
[8] Lane, T. & Brodley, C.E. 1997. An application of machine learning to anomaly detection. (In 20th National Information System Security Conference.)
[9] Frank, J. 1994. Artificial intelligence and intrusion detection: current and future directions. (In Proceedings of the 17th National Computer Security Conference, October 1994.)
[10] Forrest, S., Hofmeyer, S.A, Somayaji, A. & Longstaff, T.A. 1996. A sense of self for Unix process. (In Proceedings of the 1996 IEEE Symposium on Security& Privacy at Los Alamitos. CA. p. 120-128.)
[11] Lee, W., Stolfe, S.J. & Chan, P.K. Learning patterns from UNIX process execution traces for intrusion detection. (In AAAI Workshop: AI approaches to fraud detection and risk management, AAAI press, July 1997, p.50-56.)