Neural Network Intrusion Detection Architecture for Distributed System

ShahidNaseem
Reg # 2113246
School of Computer Sciences, NCBA&E, Lahore Pakistan
shahid.naseem@gmail.com

Abstract----Attacks on network infrastructure presently are main threats against network and information security. With the rapid expansion of computer networks, security has become a crucial issue for distributed systems. Unauthorized activities in networks, intrusion detection (ID) as a component of defense-in-depth is very necessary because traditional firewall techniques cannot provide complete protection against intrusion. Introduction detection is the problem of identifying unauthorized use, misuse, and abuse of computer systems by both system insiders and external. Intrusion Detection Systems combine distributed monitoring and data reduction through individual host and LAN monitors with centralized data analysis to monitor a network of computers. IDS's in distributed systems are based on the belief that an intruder's behavior will be noticeably different from that of a legitimate user. A main problem considered in this paper is the network-user identification problem, which is concerned with tracking a user moving across the network, possibly with a new user-id on each computer. The increased connectivity of computer systems gives greater access to outsiders and makes it easier for intruders to avoid detection. This paper proposes a new way of applying neural networks to detect intrusions i.e. "Neural Network Intrusion Detection Architecture that will use for intrusion detection. Neural network intrusion detection architecture can be used to identify each user much like detectives to place people at crime scenes.

Index Items----- Neural, Intrusion Detection, Distributed System, defense –in-depth, cracker, legitimate.

1. Introduction

Intrusion detection schemes can be classified into two categories: misuse and anomaly intrusion detection. Misuse refers to known attacks that exploit the known vulnerabilities of the system. Anomaly means unusual activity in general that could indicate an intrusion. If the observed activity of a user deviates from the expected behavior, an anomaly is said to occur.Misuse detection can be very powerful on those attacks that have been programmed in to the detection system [1]. However, it is not possible to anticipate all the different attacks that could occur, and even the attempt is laborious. Some kind of anomaly detection is ultimately necessary. One problem with anomaly detection is that it is likely to raise many false alarms. Unusual but legitimate use may sometimes be considered anomalous. The challenge is to develop a model of legitimate behavior that would accept novel legitimate use. It is difficult to build such a model for the same reason that it is hard to build a comprehensive misuse detection system: it is not possible to anticipate all possible variations of such behavior.The task can be made tractable in three ways:

i. Instead of general legitimate use, the behavior of individual users in a particular system can be modeled. The task of characterizing regular patterns in the behavior of an individual user is an easier task than trying to do it for all users simultaneously.
ii. The patterns of behavior can be learned for examples of legitimate use, instead of having to describe them by hand-coding possible behaviors.
iii. Detecting an intrusion real-time, as the user is typing commands, is very difficult because the order of commands can vary a lot. In many cases it is enough to recognize that the distribution of commands over the entire login session, or even the entire day, differs from the usual.

Level	Name	Explanation
1	Data	Audit or OS provided data
2.	Event	OS independent

		representation of user action (finite number of these)
3.	Subject	Definition and Disambiguation of network user
4.	Context	Event placed in context
5.	Threat	Definition of categories of abuse
6.	Security State	Overall network security level

Table 1: Intrusion Detection Model

A computer system is accessed by a user through an interface that translates its typing into commands. The user is free to submit what he wants (command line), or isguided by a constraining environment (menus, transaction monitor). These actions generate audit trails that we obtain at the previous levels.

2. Problem Statement

The hacker, attacking from inside as an authorized user or from outside as an intruder, uses vulnerabilities or flaws on the system. The neural network cannot take advantages of all the information in the audit data.One of the more interesting challenges for intrusion detection in a networked environment is to track users and objects (e.g., files) as they move across the network. For example, an intruder may use several different accounts on different machines during the course of an attack.Since all attacks that utilize the network for system access will pass through the LAN segment, the LAN monitor will be able to monitor all of this traffic. This architecture will motivate our work by describing the types of behavior to be detected by formulating the network-user identification, an identifier for a network-wide user and bydescribing its use in the distributed systems.It is therefore, interesting to build a tool that monitors the activities of users without specifically looking for known vulnerabilities. The data come from the audit mechanisms activated on the systems, either for security purposes or for others such as accounting.

3. Proposed Architecture

NNID (Neural Network Intrusion Detection) architecture is based on these three ideas. NNIDA is a neural network trained to identify users based on what commands they use during a day. The system administrator runs NNID at the end of each day to see if the users' sessions match their normal pattern. If not, an investigation can be launched. The NNID model is implemented in a distributed systems environment and consists of keeping logs of the commands executed, forming command histograms for each user, and learning the users' profiles from these histograms. NNID provides an elegant solution to off-line monitoring utilizing these user profiles. It is predicted that if there are 10 users, NNIDA will 96% accurate in detecting anomalous behavior (i.e. random usage patterns), with a false alarm rate of 7%. These results show that a learning offline monitoring system such as NNIDA can achieve better performance than systems that attempt to detect anomalies on-line in the command sequences, and with computationally much less effort. The rest of the paper presents the implementation and an evaluation on a distributed computer systems.

4. Intrusion Detection Systems

Many misuse and anomaly intrusion detection systems (IDSs) are independent of the platform, system vulnerability, and type of intrusion. It maintains a set of historical profiles for users, matches an audit record (user actions) with the appropriate profile, updates the profile whenever necessary, and reports any anomalies detected. Another component, a rule set, is used for detecting misuse.Often statistical methods are used to measure how anomalous the behavior is, how different e.g. the commands used are from normal behavior [2]. Such approaches require that the distribution of subjects' behavior is known. The behavior can be represented as a rule-based model in terms of predictive pattern generation or using state transition analysis.IDSs also differ in whether they are on-line or off-line. Off-line IDSs are run periodically and they detect intrusions after-the-fact based on system logs. On-line systems are designed to detect intrusions while they are happening, thereby allowing for quicker intervention. On-line IDSs are computationally very expensive because they require continuous monitoring. Decisions need to be made quickly with less data and therefore they are not as reliable.

Several IDSs that employ neural networks for on-line intrusion detection have been proposed. These systems learn to predict the next command based on a sequence of previous commands by a specific user. The network is recurrent, that is, part of the output is fed back as the input for the next step; thus, the network is constantly observing the new trend and "forgets" old behavior over time. The size of the window is an important parameter: Ifit is too small, there will be many false positives; if it is too big, the network may not generalize well to novel sequences. The most recent of such systems can predict the next command correctly around 80% of the time, and accept a command as predictable (among the three most likely next commands) 90% of the time.

One problem with the online approach is that most of the effort goes into predicting the order of commands. In many cases, the order does not matter much, but the distribution of commands that are used is revealing. A possibly effective approach could therefore be to collect statistics about the users' command usage over a period of time, such as a day, and try to recognize the distribution of commands as legitimate or anomalous off-line. This is the idea behind the NNID Architecture [3] [4].

5. The NNID Architecture

The NNID anomaly intrusion detection architecture is based on identifying a legitimate user based on the distribution of commands she or he executes [5]. This is justifiable because different users tend to exhibit different behavior, depending on their needs of the system. Some use the system to send and receive email only, and do not require services such as programming and compilation. Some engage in all kinds of activities including editing, programming, email, web browsing, and so on. However, even two users that do the same thing may not use the same application program. This approach works even if some users have aliases set up as short-hands for long commands they use frequently, because the audit log records the actual commands executed by the system. An event reported by a host monitor is called a host audit record. The record syntax isMonitor-ID, Host-ID, Audit-UID, Real-UID, Effective-UID, Time, Domain, Action, Transaction, Object, Return Value, error Code.Of all the possible events, only a subset is forwarded to the expert system. For the creation and application of the NID, it is the events which relate to the creation of user sessions or to a change in an account that are important. The Neural Network Intrusion Detection Architecture consists of the following:-

5.1 Data Acquisition

This module gathers the various audit trails (user's actions) on the system and transfers them to the station supporting the intrusion detection system [6].

5.2 Data Classification

This module arranges the data into a common format that can be interpreted toa data analysis unit. It also checks the correctness of the records.

5.3 Data Analysis Unit

This is the analysis unit of the intrusion detection system. Its main function is to receive the formatted data from data classification unit and alarming the network security administrator if there is any intrusion detect in the data on the basis of some security policiesdefined by the neural network.

5.4 Data Formation Unit

This module analyses the intrusion based data coming from data analysis unit. The purpose of this unit is to encoding the intrusion based data and to format the data bit by bit and then decoding the data again and forward to analysis and control unit. [7].

5.5 Analysis and Control

It receives the output of the data from unit and translates it into a format suitable for intrusion detection. It also monitors the internal parameters of the network for additional information. It analysis the data by comparing the output data from data formation unit with the original data in the external memory. Then for the next bit, it will check the data from data formation unit and compare it with both internal and external memories and will generate the results.

5.6 Learning Unit

It is responsible for tuning and verifying the learning process to avoid biasing the model with intrusion behavior[8].

5.7 Internal & External Memory

Internal memory module is responsible for holding the results from analysis and control unit as well as forward this result to data analysis unit for alarming the security administrator about the task completion. The external memory is responsible for holding the

initial data after applying the security policies. After gaining knowledge, the data analysis unit will gain the expertise to detect and remove the intrusions at external memory. In this way, the system performance will increase.

6. Conclusion

Most of the organizations use traditional firewall techniques in their networks for intrusion detection but it cannot provide complete protection against intrusion. Introduction detection is the process of identifying unauthorized use, misuse, and abuse of computer systems by both system insiders and external. Intrusion Detection Systems combine distributed monitoring and data reduction through individual host and LAN monitors with centralized data analysis to monitor a network of computers. IDS's in distributed systems are based on the belief that an intruder's behavior will be noticeably different from that of a legitimate user. The current IDS's do not consider the impact of the LAN structure when attempting to monitor user behavior for attacks against the system. Intrusion detection systems designed for a network environment will become important as the number and size of LAN's increase [9]. Experimental evaluation on real-world data shows that NNIDA can learn to identify users simply by what commands they use and how often, and such an identification can be used to detect intrusions in a network computer system. The order of commands does not need to be taken into account. NNIDA is easy to train and inexpensive to run because it operates off-line on daily logs. As long as real-time detection is not required, NNIDA constitutes a promising, practical approach to anomaly intrusion detection. Neural network intrusion detection architecture will be helpful in solving the network-user identification problem.

References

[1] D. D.E, "An intrusion detection model," Vols. SE-13, no. IEEE Transactions on Software Engineering, pp. 222-232, 1987.

[2] D. H, B. M and S. D, "A neural network component for an intrusion detection," no. IEEE Computer Society Symposium on Research in computer security , pp. 240-250, 1992.

[3] L. F. K, R. H. R and H. R. J, "A neural network approach towards intrusion detection in proceedings of the 13th National Security Conference," pp. 125-134, 1990.

[4] F. J, "Artificial Intelligence & Intrusion Detection: current and future directions in proceedings of the 17th National computer security conference," 1994.

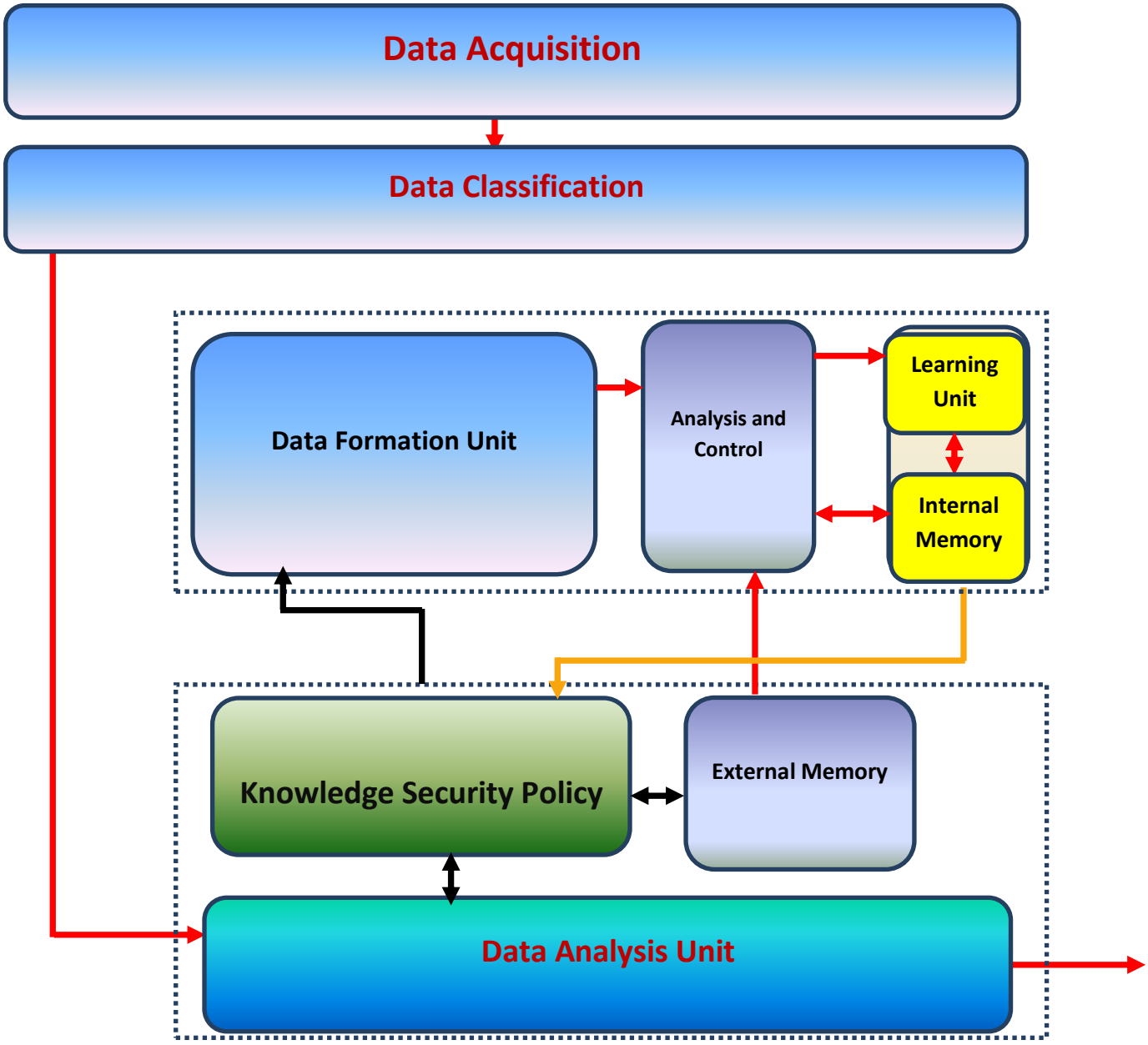
[5] G. T.D and L. T.F, "Model-based intrusion detection in proceeding of the 14th National Computer Security Conference," 1991.

[6] L. T, "Automated Audit Trail analysis &Intrusion Detection A survey proc: 11th National Computer Security Conference," no. Baltimore MD, pp. 65-73, Oct 1998.

[7] L. T.F, A. Tarraru, G. F.G and J. J, "A real-time introduction Expert System interim report 6784, SRI International," May 1990.

[8] S. W.O, "Auditory in a Distrbuted System: Sun OS MLS Audit trails Proc: 11th National Computer Security conference," no. MD Baltimove, Oct 1988.

[9] S. SE and Haystack, "An intrusion detection system: An IEEE 4th Aerospace computer security conference," vol. FL, Dec 1988.



Intrusion Detection Architecture