Intrusion Detection Systems and A View To Its Forensic Applications

The University of Melbourne
Department of Computer Science
Parkville 3052, Australia

ABSTRACT

Traditional computer security has often emphasised prevention, and to a lesser degree, the detection of system security violations. However, it is recognised that the forensic aspect to the overall model of computer security is equally as important. The area of computer forensics lends itself heavily to the response of a criminal violation that has already occurred on a system. This paper views a forensic application within the framework of Intrusion Detection and details work accomplished on a prototype anomaly Intrusion Detection system.

1 Introduction

The information revolution currently gripping the world has not only elevated the awareness of information as a valuable commodity, but also the awareness of the computer as the main repository for this information. Together with the evolution of networked computing, and the increasing degree of network interconnectivity, we have acquired the ability to easily access greater volumes of information at continuously increasing speeds.

As a result, there exists an ever increasing need to ensure that the the laws that use to govern the old paper world evolve toward the challenges of a new computer dominated environment. Information confidentiality and integrity are both subject to greater vulnerabilities within the electronic medium of today than they ever were in a paper based society.

It is therefore a foregone conclusion that computer security will need to keep in step with the unstoppable computerisation of the world's knowledge, information and operations. It is important that security procedures particularly on systems inhabiting security sensitive environments must not only encompass the protection of information against unauthorised disclosure, modification or even destruction, but also maintain the integrity of the computing system and address the ever increasing problem of computer misuse.

Today, computer security can be roughly divided into two areas. These are basically those procedures that attempt to prevent security breeches, and those that attempt to detect security violations. The role of both these areas in an ultimate model of system security is clear. However there is an additional aspect to system security, an area often termed computer forensics. It runs closely and often in parallel with the other two in terms of providing a greater depth to the overall scope of system security and response. Forensics has often been defined in terms of software, however computer forensics can often be extended to include the hardware and the executing/logging facilities of a system. Computer forensics can therefore be roughly defined as the process of examining the remains in order to obtain evidence about the factors involved in a crime or violation.

As can be expected, levels of computer crime, vandalism and fraud have become prevalent and are rising sharply given the heavy reliance on computers in our society today. Often, criminal activity is perhaps a little misguidedly attributed solely to the presence of external connections to a system. This is typically in response to the perceived threat that accompanies the burgeoning levels of Internet access and the increasingly extensive networking prevalent in many environments. Institutions that erect firewalls and other security procedures, do succeed quite effectively in negating external threat. However, often the serious threat and usually the more insidious, arises from internal violations *within* the protection of firewalls and other preventive security procedures, ie. from users within a system. An affliction that is more common in society than is often aware of, particularly in commerce.

Computer crime arising from computer misuse often manifests itself as anomalous behaviour, both of individual system users and of the system as a whole. Although improvements to operating system security continue, the available computer security features are still not good enough to detect many anomalous behaviour patterns by system users. Current computer security systems do not generally protect against [2]:

- intruders posing as legitimate system users
- a highly privileged user behaving destructively
- a legitimate user taking advantage of mistakes in the configuration of system security measures or other system vulnerabilities
- Trojan horses, or executable programs that have been altered to perform some new improper function

Any of these occurrences may prove to be a suspect vulnerability that may or may not jeopardise the security of a system. However at its worst any one or more of these may serve as a launching point for destructive criminal behaviour on the system. Therefore system anomalies can be viewed as symptoms that may herald criminal activity. It is important to note that crime committed within the electronic environment has the added attraction of a natural anonymity often afforded the perpetrator. Tracing the crime back to the criminal is generally an extremely difficult task. Should a system or user behavioural anomaly be a launching site for criminal activities, it is apparent that as much information as possible should be gathered about the act. This is not only for issues concerning the current security of the system but also in case the need arises for evidence after the fact and even for future reference.

Intrusion Detection Systems (IDS), have been explored for many years in an attempt to inject some artificial intelligence to the task of identifying any anomalies that may surface in a system. Many are based on extensive logging, and aim to collect as much knowledge/information as possible about system users.

Despite the many forms of intrusions, they can be divided into two main classes or models often employed in IDSs [7]:

- Misuse intrusions, where well defined attacks are aimed at known weak points of a system. They are often spotted by observing certain well documented actions being performed on certain objects. Due to the fact that these attacks have been experienced before and are therefore well defined/documented, very often a purely rule based detection system encapsulating the known information about the attack is applied.
- Anomaly intrusions. These are harder to quantify and are based on observations of normal system usage patterns, and detecting deviations from this norm. There are no fixed patterns that can be monitored and as a result a more "fuzzy" approach is often required.

Intrusions are inherently difficult to detect due to the great number of varying methods that are often use to affect the task. Apart from being able to exploit known architectural or operating system weaknesses, intruders may also exploit flaws within the fixes to these known weaknesses. Furthermore, a fix to a flaw in a system may also expose other existing vulnerabilities that may have been initially overlooked. The important underlying point is that the *vulnerability state* of a system is in a continual state of flux [7]. Therefore when the first, often fallible, preventive line of system defence is breached, the role of the intrusion detection mechanism comes clearly into play.

2 Intrusion Detection Systems

The proliferation of computers, resulting in an almost exponential growth of the amount of sensitive data being placed on them, has generated an increasing need to certify the level of trustworthiness of computer systems. However, it is important to note that although systems may have been certified to be secure, users may not. They can be cleared to use the system, but they can never be fully trusted. This situation is referred to as the *insider threat problem*. The user is often the weak link in an otherwise trusted system [3]. The electronic medium obscures physical recognition and, as such, a user on the system can hide the presence of hackers and impersonators or even give misbehaving legitimate users the ability to claim innocence.

The effort to address these problems has given rise to research aimed at producing viable automated intruder/anomaly detection systems, where the goal is to identify threats to the system by anomalous user behaviour or anomalous system behaviour.

The ideal intruder detection system not only has to run as close to real time as possible and encompass an expert system that will address definite system violations or *misuse* violations, but it also has to perform one of the most important tasks that a human systems programmer has to perform in order to address *anomaly* intrusions. It has to be able to acquire knowledge on the behavioural patterns characteristic of each individual user on the system in order to target possible intruder presence. The dynamic quality of user behavioural patterns is one of the greatest source of ambiguous 'rules' in intruder detection. The challenge is to implement a system that adapts to the changing nature of user behaviour, which is the motivation for initially applying neural networks to this problem. Furthermore, there is also the inherent ability of neural networks to generalise about a user's behaviour should it have limited information to work with.

The ultimate goal of Intrusion Detection is to identify, preferably in real-time, unauthorised use, misuse, and abuse of computer systems by both system insiders and external penetrators [6]. In the case of anomaly intrusions, intrusion detection is based on the idea that the anomalies that may surface in a system are symptoms indicating illegal, intrusive or criminal activity.

The ultimate goal, with a view to a forensic application however, would be to obtain sufficient evidence to in order to trace the crime back to the criminal. Within a computer system the natural blanket of anonymity afforded the criminal encourages destructive behaviour while making it extremely difficult for law enforcers to prove the identity of the criminal. Therefore the ability to obtain a fingerprint of system users and their typical behaviour is imperative in order to acquire some hold on identifying the perpetrator.

The study of available log files would always stand as a fundamental in evidence collection. However, often at a higher level it is necessary to possess a more in-depth ability to narrow the field or even establish a list of possible suspects. Computer crime is always the result of human activity on a system, be it system users or intruders. At this level, it is not only desirable to have some logging activity to provide evidential information, but also some artificially intelligent mechanism to collate and create profiles of system users. For example, it is useless to know that UserA has logged in at 8 p.m. by viewing the logs without knowing that UserA *never* logs in at 8 p.m. The knowledge that User A never logs in at 8 p.m. can only be obtained by knowing the typical behaviour of User A, or the behavioural profile of User A.

To this end the use of an anomaly ID system would be invaluable. The system would not only encapsulate user behaviour for an insight into the characteristics of a perpetrator, it also shows promise in being able to distill key signatures that would identify that perpetrator, and provides a mechanism that would correlate the log information to any alarm or anomalous activities that may arise in the course of system operations.

To date, research in this area has concentrated on the classification of statistical information of users with more complex models of neural networks [4][5]. We show that it is possible to achieve effective results using a simple variation of the feed forward network with only directly available instantaneous information of the user's behaviour. As a result, many traditional hurdles inherent to the Intrusion Detection and to neural networks [7, 8] have been overcome [1].

The prototype ID system is currently running on a student machine in the Computer Science Department, at Melbourne University known as the Obsidian Project this neural networks based ID system is aimed at machines running the UNIX operating system. The implementation of the project was targeted for UNIX systems for the following reasons:

- it is the most popular multi-user environment within educational facilities where security measures are less stringent
- it supports multiple users, therefore increasing the risk of user impersonation
- it has powerful networking capabilities that provide it with the ability to link to the Internet

3 Detection of Anomalous User Behaviour

The modelling of user behaviour on Unix machines were based on common system logs that were used as sources of information of the IDS These logs being /etc/utmp, /etc/wtmp, /usr/adm/pacct, and /usr/adm/sulog, provided the required user activity information from where we derived the following behavioural characteristics which typifies users on the system :

- User activity times - The time at which a user is normally active.
- User login hosts - The set of hosts from which a user normally logs in from.
- User foreign hosts - The set of hosts which a user normally accesses via commands on the system (eg. FTP hosts).
- Command set - The set of commands which a user normally uses.
- CPU usage - The typical CPU usage patterns of a user.

Figure 1 illustrates how a complete system for the detection of user behavioural anomalies is structured. It basically consists of five separate networks for each user behaviour and a unifying correlation network.

A coordination process is responsible for channelling system information into the neural networks. Each of the behavioural characteristics are modelled by a modified feed forward network, as well as checked by a limited static rule filter for easy breaches of security. The processing performed by these networks produce results for each user characteristic. The correlation network combines several different behavioural measures to provide a higher order model for a user's trends. It should be slow to raise alarm in order to allow a legitimate user to change. If repeated violations occur it needs to reach alarm levels fast enough to alert a systems administrator when necessary.

These separate networks aid in the learning of a unifying correlation network as well as provide detailed information should an alarm be triggered. The desired response of the correlation network is a value close to -1 for a user who is not deviating from their normal behaviour, and a value of 1 for a user who is deviating in the most extreme sense. For the results presented in this paper we used an arbitrary threshold value of 0.5 to determine when to raise an alarm for anomalous user activity. If the output of the correlation network is above this threshold then an alarm is raised. It is also important to note that our system will supply several levels of alarm, where the threshold values are adjustable by the system programmer in order to allow adaptation to the required sensitivity needed in the computer system. The lowest level of alarm could be used to trigger extensive auditing of the targeted account, while a higher level of alarm could be used to alert the system administrator immediately.

4 Results

Results were obtained on the departmental student machine (mundil.cs.mu.OZ.AU) which is a Silicon Graphics 4D/340S server. This machine is a medium to large multi-user system with 128Mb of memory, 4 CPUs. It has

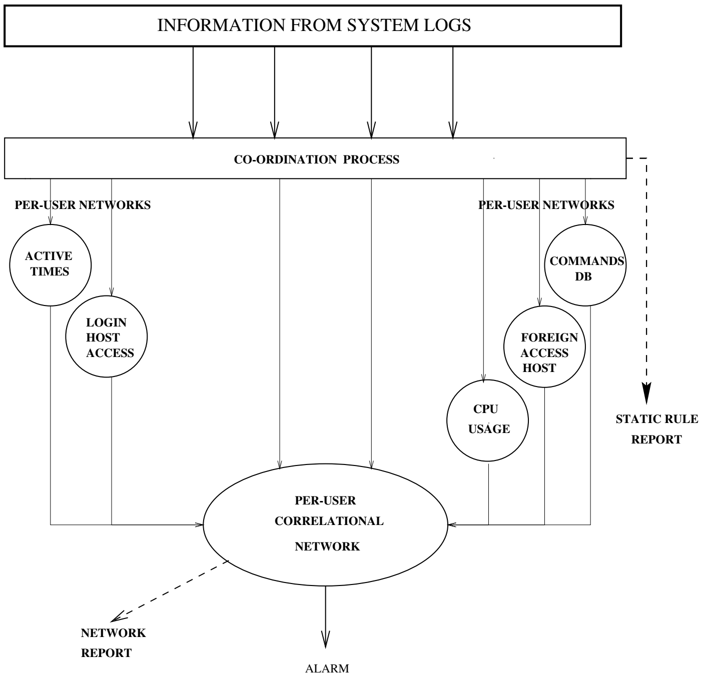


Figure 1: Structure Of An Automated User Behaviour Anomaly Detection System

approximately 35 users on the system on average and approximately 157 users maximum. There is also an average of about 200 logins per day, peaking at over 1000.

In order to test the effectiveness of this system, it was allowed to run for a period of two weeks. Then the entire neural network detection system was tested by introducing a sample of anomalous behaviour profiles obtained from various system administrators. The goal of this exercise is to determine if the correlation network's output indicated anomalous activity in these accounts

The response of the correlation network for each user was recorded and graphed. A summary of the resulting graphs are displayed in Figure 2. They illustrate the response of the network when processing a normal user with non-varying behaviour, a normal user with changing behaviour, and and account under attack. In addition, we tested the response of the network should a user use a another users account and vice versa.

The following is a table that illustrates the percentage of CPU, memory and I/O load our system placed on the tested student system to maintain the neural networks. These results are of particular importance considering that one of the major reasons why expert systems versions of these detection systems are not viable is the heavy demands that they place on system resources. For practical viability, the detection system must use between 1 to 2 percent of the available system resources.

System Load	% Of Total CPU Resources Required	% Of Total Memory Resources Required	% Of Total Disk IO Resources Required
Average	0.84%	0.4%	0.02%
Maximum	3.5%	1.7%	0.1%

Table 1 : Summary of the Percentage of System Resources Required by the Neural Network Detection System

Our tested system under maximum load had 157 active users at once, while on average it had 35 users. Under maximum load the memory requirements are approximately 2.3Mb. Under normal load this comes down to approximately 0.5Mb.

The total disk space requirements of our proposed system for a student machine with 1000 accounts would be 14.6Mb. On the machine we tested this would account for 1.4% of the machine's disk space.

I/O resources are consumed during the reading and writing back of the state of each user's neural network. The state of each active network in memory is written out to disk every 1/2 an hour to prevent loss of information.

5 Conclusion

The introduction of the computer has brought about, what is popularly termed, the "Information Revolution". Never before has such a wealth of data, both public and private, been so accessible and obtainable. With this revolution has come certain undesirable elements, being the underworld of computer fraud and crackers, who quite often achieve their aims by breaking into computer systems and impersonating legitimate system users.

Many methods preventing intrusion into computer systems have been implemented, but these will always be imperfect. When preventive measures fail, many sites rely on the local expertise of their systems administrators to apprehend the intruders. This task is both time consuming and error prone, requiring a human to sift through copious quantities of data looking for deviations in user behaviour.

An automated system for detecting anomalous user behaviour will help alleviate a, sometimes unrealistic, burden on systems administrators. However, these automated systems are not only useful during a violation but can be an invaluable forensic tool after the fact.

References

[1] K.Tan, "An Application Of Neural Networks To UNIX System Securi ty", *IEEE International Conference On Neural Networks*, November 1995.

[2] H.S. Vaccaro and G.E. Liepins, "Detection of Anomalous Computer Session Activity", *1989 IEEE Computer Society Symposium on Security and Privacy,* May 1989, pp. 280-289.

[3] J.R. Winkler and W.J. Page, "Intrusion and Anomaly Detection in Trusted Systems", *Fifth Annual Computer Security Applications Conference,* Dec 1989, pp. 39-45.

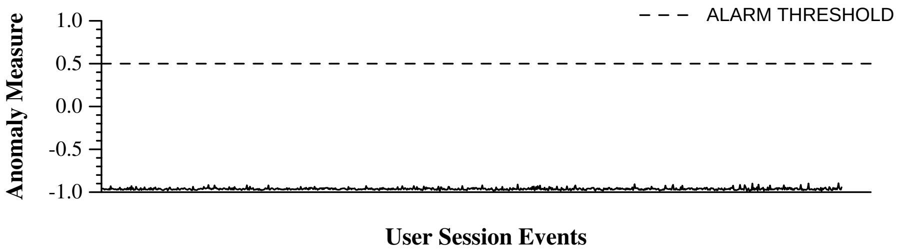
[4] H. Debar and B. Dorizzi, "An Application of a Recurrent Network to an Intrusion Detection System", *IJCNN Intl Joint Conference in Neural Networks,* June 1992.

[5] H. Debbar, M. Becker and D. Siboni "A Neural Network Component for an Intrusion Detection System", *Proceed ings of the Symposium on Research in Security and Privacy,* May 1992.

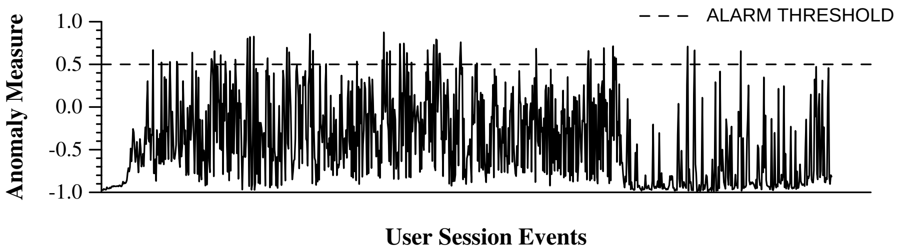
[6] B. Mukherjee, L. T. Herberlein and K. N. Levitt, "Network Intru sion Detection", *IEEE Network,* May/June 1994, volume 8.

[7] M. Crosbie and G. Spafford, COAST Group, "Active Defens e of a Computer System Using Autonomous Agents", *Technical Report No. 95-008,* February 1995,

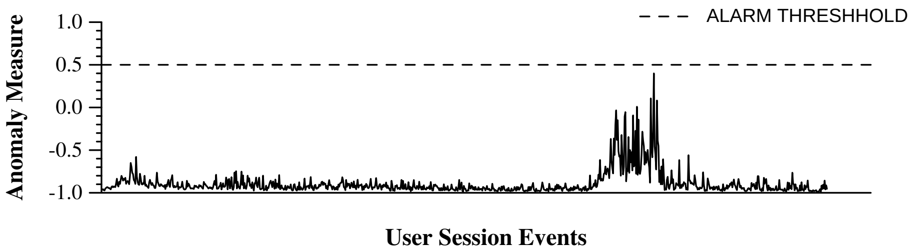
[8] S. Kumar and G. Spafford, "An Application of Pattern Matching i n Intrusion Detection", *Technical Report CSD-TR-94-013*, June 1994.



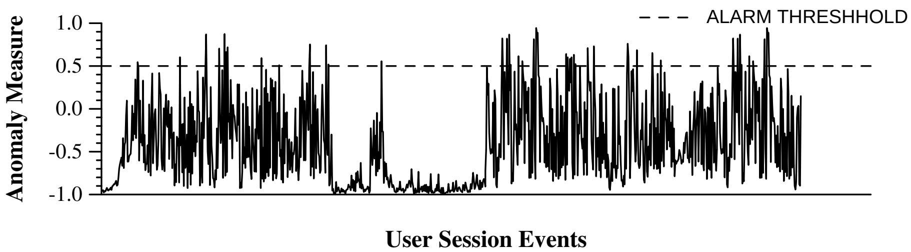
Test Login Session - Normal Input - User 1



Test Login Session - Attack - User 2



Test Login Session - Changing Behavior - User 3



Test Login Session - Switching Two Users

Figure 2: Summary Graphs of Correlation Network for Three Accounts Under Testing