Data Gathering for Insider Misuse Monitoring

Aung Htike Phyo and Steven Furnell

Network Research Group, University of Plymouth, Plymouth, United Kingdom

Email: aung@jack.see.plymouth.ac.uk, sfurnell@network-research-group.org

Abstract

The impact of insider IT abuse can be devastating compared to most outsider attacks. In principle some of the techniques used in Intrusion Detection Systems (IDS) are transferable to Insider Misuse detection. The difference between a traditional IDS and an Insider Misuse Monitoring system is the type of data collected and analysed. This paper discusses the types of data needed to monitor Insider Misuse and the different methods by which it may be collected, and then explains why application level detection has more potential over the others.

Keywords

Insider IT abuse, Misuse Monitoring, Data Gathering, Intrusion Detection, Misfeasor Detection

1. Introduction

In recent years society has become increasingly dependant on IT infrastructures, as many organisations (including telecommunication, healthcare, banking, transport, emergency services and the military) use IT for the smooth functioning of their operations. Therefore IT systems are critical to our everyday lives. In response, the IT industry has launched a variety of security tools to help the users and system administrators prevent, detect and sometimes respond to the abuse of the systems. Security tools frequently employed in today's computer systems include anti-virus toolkits, firewalls and Intrusion Detection Systems (IDS). In recent years, attacks from outside the organisations have increased, due to an increasing number of organisations getting connected to the Internet and being exposed to attacks. However the results of the surveys by CSI/FBI in recent years have constantly suggested that the dollar amount lost due to insider abuse is greater than the loss due to abuse from outsiders (Power 2002). Insider abuse can have a major impact upon an organisation since the perpetrators have a good idea of what is sensitive and valuable within the company. Knowing where these resources are stored, and what security mechanisms are used to protect them, also helps insiders in circumventing controls and evading detection. As such, it is essential for organisations to be cognisant of the threat, and for mechanisms to be available to facilitate detection of these incidents, as well as those that come from the outside. This paper considers the feasibility of such mechanisms, based upon principles of data collection and analysis that are already applied in the context of intrusion detection systems.

2. Background

Before discussing further on the issue of Insider Misuse, there is a need to define the terms 'Insider' and 'Misuse'. From the organisation's point of view, insiders can be employees, part-time employees, consultants, contractors and employees of partner firms. From the system's perspective, insiders are users with a valid login account to access the resources it manages. Users may be physically located inside or outside the organisation, but have the same logical presence. By contrast, some individuals may be physically inside the organisation, but lack a valid account to access the systems. In this context, they are regarded

as logical outsiders, and for the purpose of this paper the term 'insider' refers to users with valid login accounts (i.e. the logical insiders). In general misusers are the users who have legitimate access to the IT systems and the data stored on it, but abuse their privileges by using the resources in an inappropriate manner or for an unapproved purpose. According to Anderson (1980), such users can be termed 'misfeasors'. The word 'misuse' implies the presence of rules that specify the conditions of allowable usage for the resources concerned. These rules are often embodied within an IT usage policy. The nature of misuse is widespread, with a wide-range of possible misuse scenarios. Some of these misuse activities require a closer scrutiny due to the financial impact they can have on the organisation, such as:

- Net abuse
- Data theft
- Sabotage
- Fraud
- Use of unauthorised software

Aside from Net abuse and the use of unauthorised software, the activities listed are essentially old problems in a new environment. In the IT network, large amounts of data can be stolen unrecognisably in a short period of time. Electronic data can be destroyed at the click of a button if the perpetrator has the appropriate privileges, and such a process will not be immediately noticed unless monitoring facilities are carefully implemented. Fraud committed in the IT medium is difficult to prevent due to difficulties in implementing controls that resemble organisational hierarchy and the enormous amount of data involved. This in turn makes it even more difficult for automatic detection of the fraud due to the system's lack of knowledge in business processes and management hierarchy.

Common security mechanisms found in Microsoft and Unix-based operating systems (OS) are Identification and Authentication, Access Control, and Auditing. The purpose of Identification and Authentication is to make sure the user is who he claims to be, and it therefore represents a frontline defence against unauthorised users. Such controls are clearly ineffective against insider misusers, who have legitimate access into systems. Once a user is logged in, the role of Access Control is to prevent them from accessing systems and data to which they are not entitled. However, traditional access controls can only allow or deny access to a resource, and the problem is that insiders have legitimate access to the resources that they may subsequently misuse. As such, the main countermeasure at the moment is to retrospectively monitor what they are doing, and determine whether misuse has occurred. In this context, audit mechanisms produce audit trails of events and logs of data concerning the system usage. Most operating systems provide an audit mechanism that is at least capable of logging every file accessed by a user. From a security perspective, the main purpose of logging is to be able to hold users accountable for their actions. However, although the majority of the computers in sensitive environments log audit data, most of the audit data is generally utilised for performance measurement or accounting purposes, and not very useful for intrusion detection (Lunt 1993). Most systems allow the administrator to identify what data is sensitive and who needs access to it. However the ability to detect the manner in which the data is accessed and the actions after gaining access is somewhat limited. Therefore comprehensive auditing is required in order to monitor such operations. In an organisation with hundreds of users, large amounts of audit data is logged and it becomes very difficult for the system administrators to manually detect attacks by examining the log files. In dealing with externally sourced incidents, Intrusion Detection Systems can ease this

task by automating the process of looking for attack patterns in log files. With this in mind, consideration can be given to applying a similar technique as a means of identifying insider misuse activity.

Intrusion detection is based on auditing by helping the administrator look for known attack scenarios, anomalous user/system behaviour, combination of suspicious activities, and patterns of events that associate with malicious behaviour. Depending on the source of data used for analysis, IDS can be classified in to:

- *Network-based*: The IDS performs detection at the network level, and the network traffic is monitored to look for attacks patterns.

- *Host-based*: The IDS performs detection at the OS level. The main sources of data are the audit trails and event logs.

Host-based IDS can then be further sub-classed depending on the level of monitoring that they employ:

- *System-level monitoring*: Monitors system events such as system calls, CPU usage, file access and I/O.

- *Application-level monitoring*: Monitors user interactions with the application such as request-response, access patterns, user input, application output, and user utilisation of application functions.

Having collected such data, IDS can employ two main strategies to identify attacks, namely misuse-based detection and anomaly-based detection (Amoroso 1999).

- *Misuse-Based detection*: This approach relies upon knowing or predicting the intrusion scenario that the system is to detect. Intrusions are specified as attack signatures, which can then be matched to current activity using a rule-based approach. A similar approach could potentially be incorporated for misfeasor incidents, based upon those methods that employees have been known to exploit in the past, or those that can be anticipated they would attempt based upon the privileges and resources available to them.

- *Anomaly-based detection*: Rather than being based upon known or predicted patterns of misuse, this approach relies upon watching out for things that do not look normal when compared to typical user activity within the system. In standard IDS, the principle is that any event that appears abnormal might be indicative of a security breach having occurred or being in progress. The assessment of abnormality is based upon a comparison of current activity against a historical profile of user (or system) behaviour that has been established over time.

As with many of the IT security technologies, IDSs are geared towards detecting intrusion from outside the network or system security violations by legitimate users. However, some of the data collection and analysis techniques employed by traditional IDSs can theoretically be used to develop a misfeasor-monitoring system. As a first step towards achieving this, we need to review current data gathering techniques, the data that can be collected by such techniques, and their collective suitability for use in misfeasor monitoring.

3. Review of Data Collection Techniques

As already established in the context of traditional IDS, different types of data can be gathered at varying levels in a computer system. As different types of misuse can manifest themselves on different levels of a system, it is important that the relevant data is collected at the appropriate level. The different options, and their applicability to insider misuse detection, will now be considered in more detail.

3.1 Network-level Monitoring

This technique is used by network-level IDSs where network packets are the main source of data for monitoring. Network packets are captured by placing the network interface cards in promiscuous mode. Network data collection modules need to be strategically placed in the network in order to capture all the network traffic, usual places include the first node after the router in a subnet, on a gateway between two subnets, or just after a firewall in an organisation. Network environments are often divided into multiple subnets for security and performance reasons. In order to monitor network traffic for all subnets, each subnet would need a separate data collection station, and to monitor the traffic entering and leaving the sub-net, the monitors would need to pickup all the packets.

Packets are considered suspicious if they match some predefined signatures. Three main types of signatures are string signatures, port signatures and header condition signatures. By checking header fields in the packets, the IDS would be able to monitor attacks on the network protocols. By monitoring packet content, remote exploitation of application and/or system vulnerabilities can be monitored. Packet content can also be used to monitor web and email usage. This type of collector would pickup packets going in and out of a subnet, but do not monitor traffic in the subnet, since they are primarily designed for perimeter security. If encryption were implemented by network services, the monitor would not be able to analyse the data collected in this manner. For example, if IP tunnelling is established between two computers, the sniffer needs to be in the OS network stack of the concerned machines in order to see the packet in clear text. Again, this approach would not work if the encryption took place at application level, such as an SSL encryption. This approach will not allow detection of system level attacks, attacks from directly attached terminals or attacks via dial-in modems directly connected to the target computer.

From a misfeasor monitoring perspective, network-level data collection can help in detecting insiders who employ the same methods used by outsiders to attack the internal systems. In addition it can also help in monitoring:

- Web access
- Email content
- Excessive usage of network resources
- Anomalous access of isolated sub-nets
- Utilizing services from unauthorised terminals

Although many users may accept monitoring web access and excessive usage of network resources, monitoring or filtering of email is subject to debate of privacy in the workplace and legal issues. It is also important that anomalous access of isolated sub-nets is monitored. For example, questions need to be asked when a software developer establishes direct

network connection to the systems in the payroll department, as the user in question may be in process of modifying the payroll database in order to raise his earnings. Utilizing network services from unauthorised terminals should also be monitored, since access-terminal security is very important in trust-based distributed computing environments. The perpetrator here might be using a rogue client program to access the services. Again controls are sometimes placed within the application environment and the use of arbitrary programs to access the services may allow the user to by pass the controls either accidentally or intentionally by the user. Having stated the possible monitoring opportunities for insider misuse at the network level, we should consider the statement by Schultz (2002), "Insiders do not generally demonstrate the same attack signatures as external attackers". Insiders may already have user accounts to access the systems concerned and in most cases that also means physical access. Therefore, there might not be a need to remotely exploit the services or protocols in order to gain access. Insiders are also wary of setting off alarms in the process of misuse, and they are more likely to abuse their existing privileges than to exploit remote vulnerabilities. This leads us to the need for monitoring at the system level.

3.2 System-level Monitoring

Continuing from the previous discussion on collecting network data, it is possible to monitor network packets entering and leaving the system by running the data collection module as part of the OS, in the network stack at the system level (Kerschbaum et al. 2000). The disadvantages of this approach are the need to correlate the attack logs from each machine to get a network-wide view of the attacks, and performance degradation of the concerned system. At the system level, the main source of data collected is from audit trails, application logs, and system events. In addition system calls, kernel messages, system statistics and access violations can be monitored to characterize system/application behaviour. Audit logs usually provide information on access violations, change of system and configuration files. As stated previously, IDS automates the process of looking for known attack scenarios, anomalous user/system behaviour, such as a combination of suspicious activities, and patterns of events that associate with malicious behaviour. The following are types of suspicious activities that may be monitored at the system-level:

- *Covering tracks*: Example, a user attempts to modify audit configurations, deleting entries in the log files, and making changes to accounting configuration.

- *Unauthorised programs*: Monitor execution of unauthorised programs for they may be Trojan horses or rouge programs. There is also a chance of the user utilising such programs for a malicious purpose. For example access of database files with the use of an arbitrary program.

- *Monitor system consequences*: Example, presence of an unauthorised device driver, or the machine listening on an unauthorised port. The presence of a modem might indicate, the user directly connecting to the Internet, bypassing the network monitoring system. This also gives the opportunity to send information out of the organisation without being monitored.

- *Monitoring Access*: Monitor successful access in order to monitor frequency of access to certain files; this will later enable the system to characterize file access by users/processes. Monitor access to files tagged as confidential (this requires a database of confidential file names).

- *Monitor file deletes*: Monitor deletion of files, especially batch deletion of files. Deletion of files on the backup servers need even more care. Both of the mentioned activities may be intended to sabotage the system and resources it manages.

In addition to the above, there are a number of activities that can be monitored at system level for insider misuse monitoring. Some of those activities are:

- Check for events where the User ID of the owner of the process is not equal to the User ID of the owner of the object accessed (objects here can include File, Directory, or an executable program). Even though the user might have gained privilege to access the objects, such events might indicate breach of privacy by the privileged user.

- Atypical usage of I/O resources by systems may also indicate information leakage. For example, unusual access of the Internet by the backup server.

It is also possible to monitor user behaviour at the system level, such as the applications/commands the user often utilizes, system access times, and the type of network services used. Utilization of some of the applications/commands may indicate preparatory behaviour, for example the use of a port/vulnerability scanner by a user, who does not have system administration duties. It may also be appropriate to monitor the input source and output destination of data to and from an application. For example, when the tagged secret-file is used as an input to the encryption/steganographic program, the user might be in the process of disguising the information before sending it out of the organisation. The suspicion level should naturally increase when the output of the previously mentioned activity is attached in an email to be sent out of the organisation. However some types of abuse will be distinguishable from normal activity only with the knowledge of application-level semantics and subsequently may not exhibit malicious behaviour at the system level. Therefore some detection strategies will be necessary at the application and database level.

3.3 Application-level Monitoring

Although a few researchers have worked on misuse detection (Chung et al. 1999) and data collection (Almgren and Lindqvist 2001) at the application level, this is a relatively less explored area compared to the first two techniques. At this level, the main source of data can be input from user/processes, output produced by the application, user actions within the application environment and the application data itself. Monitoring criteria here include:

- *Range of input/output data*. By constantly monitoring maximum and minimum values for certain items in a record, some types of fraud may be detected. One real-life example would be the case of Joseph Jett (Dhillon et al. 2001), where Jett indefinitely postponed the time the actual losses could be recognised in a Profit and Losses statement.

- *Destination of output*. By monitoring the destination of output, information leakage could be monitored. For example, if the data is written to a world readable file, it could compromise the confidentiality of the data.

- *Type of input/output data*. By monitoring the type of input, such as numbers, strings and control characters, attempts to compromise the integrity of the running process and its data can be detected.

- *Format of input/output data*. By monitoring the format of the data entered such as time/date formats, some of the accidents that could otherwise compromise the integrity of the data can be detected.

- *Access patterns*. By monitoring user access patterns such as read/write, to certain items in a record, user access behaviour can be characterised over time to determine their normal activity.

Using the above data, it is possible to create profiles of the normal behaviour associated with a user or a user-class (with the latter being based upon the user's role within the organisation). The question of which is more effective requires more research and investigation. However, at the moment the authors conjecture that the class-based profiling has potential in misfeasor detection, as it is assumed that the users with the same responsibilities would exhibit similar if not identical activities within the system. Their similarities should be clear in terms of the applications frequently used and the actions performed within the application environment. Therefore, the individual profile of a misfeasor should be obvious when compared to the class-based profile the perpetrator belongs to. Another advantage of class-based profile comparison is that when the users of a particular role are assigned special assignments, the sudden change of user profile may not be considered anomalous, if the changes are similar for all users within the same role. Again this approach may also help monitor users who gradually train the system to accept anomalous behaviour as normal.

For the purpose of misfeasor monitoring, the authors feel that application level monitoring can provide most relevant data; because this is where the users directly interact with the application environment and the concerned data. Therefore the data collected here should reveal more about the user behaviour within the environment, and it gives a better understanding of the user's intentions. Again, the user actions and input to the application is more meaningful when monitored at this level. However, these hypotheses need to be proven, and our future research will focus on this. The advantages of collecting data at this level are that the data is unencrypted and it gives an insight into how the application interprets the transaction. It also gives the opportunity to reconstruct the session by logging request-response transactions. The ability to reconstruct the session is very important as it allows the security personnel to investigate what actually happened to find out if the actions were accidental or intentional. Session reconstruction also allows the characterisation of the particular misuse scenario, to automate future detection. The disadvantage of this approach is the potential effect on the performance of the application. If implemented without care the collected data may also reveal confidential information and system vulnerabilities that can be used by misfeasors. It is also vital how the collection module is implemented. With some of the applications it may be sufficient just to monitor the data logged, however, with some applications it might be necessary to modify the code in order to get the desired data. For the latter approach, it needs to be identified where in the application the data collection function should be placed. Again this might vary from one application to another. Therefore more research needs to be carried out to identify the best manner in which the data can be collected at this level and how it can be transferred or stored safely for analysis.

To understand how application level monitoring works, we can consider previous work in the domain. A good example is provided by DEMIDS (Detection of Misuse in Database Systems), which attempts to profile working scopes based on user access patterns in database systems (Chung et al. 1999). DEMIDS assumes that a user typically will not access all attributes and data in a database schema; therefore access patterns of users will form some working scopes, which are sets of attributes usually referenced together with some values. Based on that assumption, Chung at al defined the notion of a distance measure between sets of attributes that consider both the structure of the data and user behaviour. This notion is then used to guide the search for regular patterns that describe user behaviour in a relational database.

4. Predicting the insider threat

It is important to note that insider misuse is both a managerial and a technical problem. One of the complicating aspects with insiders, and the aspect that differentiates this from the outsiders, is that incidents will not always relate to something that is unauthorised. Indeed, the basic problem with insider misuse is that the person concerned has legitimate access to IT resources of the target organisation. Again it may not be system vulnerabilities that are exploited, but exploitation of the business processes and management loopholes in the IT environment. Therefore, knowledge of the business hierarchy, the segregation of duties and responsibilities of the users are important in monitoring insiders, as this type information can give an idea of who needs to be monitored closely. However, one advantage insider misuse monitoring has over outsider attack detection is that the insiders can be profiled not only based on their IT usage behaviour, but also their personality traits, job positions, responsibilities, knowledge of the system and understanding of the business processes. Based on this information, analysis may be made to calculate the possibility of misuse by certain users. Knowledge of job positions and segregation of duties are important as the opportunity for misuse arises when the individual is in a position of trust and the controls are weak. There are also prediction theories on this issue, such as privileged users who know more about the system are more likely to misuse (Magklaras and Furnell 2002). Privileged users are in better position to misuse and evade detection for a longer period, though it cannot be concluded that the majority of the privileged users would misuse the systems, actions by privileged users should be closely monitored as even the innocent errors may have serious consequences. Indeed, the opportunity for fraud often begins when a user realises that an innocent error has passed unnoticed, thus exposing a weakness in the internal controls (Coderre 1999). The same principle applies to insider misuse in general, and it occurs “when a ready mind meets an opportunity” (Tuglular 2000). However, having privileges and being in a position of trust is not enough to speculate misuse, a generic insider threat model referred to as “CMO” postulates that in order to misuse a computer system, the perpetrator must have: the Capability to misuse, Motive to do so and the Opportunity to launch the attack. Therefore, the user must have the technical ability, understanding of business processes, be in the position of trust to launch the attack and finally the motivation to do so. This requirements specification can be helpful in predicting the potential for insider misuse. Users can then be classified on their technical ability, length of time in the position, and their duties. Finally, if reasonable explanation can be provided on why the user would be motivated to misuse the system, then it would give a reason for closer monitoring of the concerned user.

5. Conclusions

Existing data collection and analysis technologies used by traditional IDSs can be used to monitor certain types of insider misuse. However, many insider misuses do not exhibit the same attack patterns as external attacks. Various types of insider misuse can manifest themselves on different levels of a system and it is important that the data is collected at the relevant level. While network-level data collection can help monitor insider abuse of net-usage, system-level data collection can help monitor data-theft, sabotage and use of unauthorised software. However, fraud may only be detected at the application level with the help of domain knowledge. Data collection at the three levels of the system is only the first part of the data gathering process. Additional knowledge of the users, organisation's management hierarchy, business processes and job responsibilities are equally important in monitoring insider misuse. The authors' future research will focus on the development of a misfeasor monitoring system that utilizes the data collection techniques and user profiling strategies discussed in this paper.

References

Almgren, M. Lindqvist, U. (2001) "Application-Integrated Data Collection for Security Monitoring", in the *Proceedings of RAID 2001*, pp. 22-36.

Amoroso, E. (1999) *Intrusion Detection: An Introduction to Internet Surveillance,Correlation, Traceback, Traps and Response*, First Edition, Intrusion.Net books, NJ, ISBN: 0966670078.

Anderson, J.P. (1980) "Computer Security Threat Monitoring and Surveillance".

Chung, C.Y. Gertz, M. Levitt, K. (1999) "DEMIDS: A Misuse Detection System for Database Systems", in the *Proceedings of the 3rd International Working Conference on Integrity and Internal control in Information Systems*, pp. 159-178.

Coderre, D.G. (1999) *Fraud Detection: Using Data Analysis Techniques to Detect Fraud*, Global Audit Publications, ISBN 0-9684400-8-8

Dhillon, G. Moores, S. (2001) "Computer crimes: therorizing about the enemy within", *Computers & Security*, Vol.20, No. 8, pp. 715-723

Kerschbaum, F. Spafford, E.H. Zamboni, D. (2000) "Using embedded sensors for detecting network attacks", in the *Proceedings of the first ACM Workshop on Intrusion Detection Systems*, Athens, Greece.

Lunt, T.F (1993) "Detecting Intruders in Computer Systems", *1993 Conference on Auditing and Computer Technology*.

Magklaras, G.B, Furnell, S.M (2002) "Insider Threat Prediction Tool: Evaluating the probability of IT misuse", *Computers & Security*, Vol. 21, No. 1, pp. 62-73.

Power, R. (2002) "2002 CSI/FBI Computer Crime and Security Survey", *Computer Security Issues & Trends*, Vol. VIII, No. 1. Computer Security Institute. Spring 2002.

Schultz, E.E. (2002) "A framework for understanding and predicting insider attacks", *Computers & Security*, Vol. 21, No.6, pp. 526-531

Tuglular, T. (2000) “A preliminary Structural Approach to Insider Computer Misuse Incidents”, *EICAR 2000 Best Paper Proceedings*: pp. 105-125.