A GENERIC ARCHITECTURE FOR INSIDER MISUSE MONITORING IN IT SYSTEMS

By

AUNG HTIKE PHYO

A thesis submitted to the University of Plymouth
in partial fulfilment for the degree of

DOCTOR OF PHILOSOPHY

School of Computing Communications & Electronics
Faculty of Technology

September 2007

University of Plymouth
Library
9007951 25X
THESIS 005.8 PHY

Abstract

A Generic Architecture for Insider Misuse Monitoring in IT Systems

Aung Htike Phyo
BSc (Hons)

Intrusion Detection Systems (IDS) have been widely deployed within many organisations' IT networks to detect network penetration attacks by outsiders and privilege escalation attacks by insiders. However, traditional IDS are ineffective for detecting of abuse of legitimate privileges by authorised users within the organisation i.e. the detection of misfeasance. In essence insider IT abuse does not violate system level controls, yet violates acceptable usage policy, business controls, or code of conduct defined by the organisation. However, the acceptable usage policy can vary from one organisation to another, and the acceptability of user activities can also change depending upon the user(s), application, machine, data, and other contextual conditions associated with the entities involved. The fact that the perpetrators are authorised users and that the insider misuse activities do not violate system level controls makes detection of insider abuse more complicated than detection of attacks by outsiders.

The overall aim of the research is to determine novel methods by which monitoring and detection may be improved to enable successful detection of insider IT abuse. The discussion begins with a comprehensive investigation of insider IT misuse, encompassing the breadth and scale of the problem. Consideration is then given to the sufficiency of existing safeguards, with the conclusion that they provide an inadequate basis for detecting many of the problems. This finding is used as the justification for considering research into alternative approaches.

The realisation of the research objective includes the development of a taxonomy for identification of various levels within the system from which the relevant data associated with each type of misuse can be collected, and formulation of a checklist for identification of applications that requires misfeasor monitoring. Based upon this foundation a novel architecture for monitoring of insider IT misuse, has been designed. The design offers new analysis procedures to be added, while providing methods to include relevant contextual parameters from dispersed systems for analysis and reference. The proposed system differs from existing IDS in the way that it focuses on detecting contextual misuse of authorised privileges and legitimate operations, rather than detecting exploitation of network protocols and system level vulnerabilities.

The main concepts of the new architecture were validated through a proof-of-concept prototype system. A number of case scenarios were used to demonstrate the validity of analysis procedures developed and how the contextual data from dispersed databases can be used for analysis of various types of insider activities. This helped prove that the existing detection technologies can be adopted for detection of insider IT misuse, and that the research has thus provided valuable contribution to the domain.

Table of Contents

LIST OF TABLES XI

LIST OF FIGURES XIII

GLOSSARY XV

ACKNOWLEDGEMENTS XVI

AUTHOR'S DECLARATION XVII

CHAPTER 1 INTRODUCTION 1

1.1 INTRODUCTION 2

1.2 MONITORING INSIDER MISUSE 2

1.3 AIMS AND OBJECTIVES OF RESEARCH 4

1.4 THESIS STRUCTURE 5

CHAPTER 2 NATURE OF INSIDER IT MISUSE 8

2.1 INTRODUCTION 9

2.2 DEFINITION OF INSIDER 9

2.2.1 The scope of insider within the thesis 10

2.3 DIFFERENT CATEGORIES OF IT MISUSERS 10

2.3.1 Scope of misuse within the thesis 13

2.4 TYPES OF MISUSES 14

2.4.1 Internet access abuse 15

2.4.2 Information theft 15

2.4.4 Breach of privacy/confidentiality 17

2.4.5 Sabotage 18

2.4.6 Fraud 19

2.5 FACTORS LEADING TO MISUSE 19

2.5.1 Capability *21*

2.5.2 Motivations behind misfeasance *21*

2.5.2.1 Financial Gain 22

2.5.2.2 Ego and Pressure 23

2.5.2.3 Vengeance 24

2.5.2.4 Recreation, Breach of Privacy, Curiosity, Social Engineering, Naivety, Accidental Misuse, Un-Wittedly Running a Trojan Application 25

2.5.3 Opportunity *25*

2.6 THE SCALE OF MISUSE 27

2.7 FAMOUS CASES 31

2.7.1 Joseph Jett vs. Kidder Peabody (Dhillon and Moores 2001) *32*

2.7.2 Nick Leeson vs. Barings Bank (Asiaweek 1995) *32*

2.7.3 Tim Lloyd vs. Omega (Gaudin 2000) *33*

2.7.4 Robert Hanssen vs. FBI *34*

2.8 CONCLUSIONS 35

CHAPTER 3 AN EXAMINATION OF EXISTING SAFEGUARDS AGAINST MISUSE 36

3.1 INTRODUCTION 37

3.2 NON-TECHNICAL MEASURES 38

3.2.1 Physical security *39*

3.3 TECHNICAL MEASURES 40

3.4 COMPUTER SECURITY MODEL 41

3.4.1 Identification and Authentication *43*

3.4.2 Security and Usage Policy *43*

3.4.2.1 White listing and black listing 44

3.4.3 Access control *44*

3.4.3.1 DAC (Discretionary Access Control) 45

3.4.3.2 MAC (Mandatory Access Policy) 46

3.4.3.3 Privilege and Access Rights Management 47

3.4.3.4 Role Based Access Control (RBAC) 48
3.4.4 Ownership of information *51*
3.4.5 Limitations and challenges of access control *51*
3.4.6 Misfeasance vs. Access Control *55*
3.5 AUDITING 55
3.6 CONCLUSIONS 56

CHAPTER 4 IT SECURITY MONITORING AND DETECTION TOOLS 59

4.1 INTRODUCTION 60
4.2 INTRUSION DETECTION 60
4.2.1 Components of Intrusion Detection System *62*
4.3 DATA COLLECTION FOR INTRUSION ANALYSIS 64
4.3.1 Typical Network Structure of an Organisation *66*
4.3.2 Network-Level Intrusion Detection Systems *67*
4.3.3 Host-Level Intrusion Detection Systems *73*
4.3.3.1 Application-level monitoring 76
4.4 DETECTION STRATEGIES 78
4.5 DETECTION TECHNOLOGIES 82
4.5.1 Expert Systems *83*
4.5.2 State Transition Analysis *83*
4.5.3 Model-based detection *83*
4.5.4 Statistical Profiling *84*
4.5.5 Predictive pattern generation *84*
4.5.6 Neural networks *85*
4.6 RELEVANT SYSTEMS 87
4.6.1 DEMIDS *87*
4.6.2 DIDAFIT *88*
4.6.3 eXpert-BSM *89*

4.6.4 Orchestria *92*

4.6.5 PortAuthority 5.0 ..... *92*

4.6.6 NetReplay *93*

4.7 CONCLUSIONS 93

CHAPTER 5 A DETECTION-ORIENTED CLASSIFICATION OF MISFEASANCE 96

5.1 INTRODUCTION 97

5.2 A REVIEW OF CURRENT INTRUSION TAXONOMIES 98

5.3 A DETECTION-ORIENTED APPROACH TO CLASSIFICATION 103

5.3.1 Network-level misuses *107*

5.3.2 System-Level misuses *111*

5.3.3 Application and data-level misuses *116*

5.4 CONCLUSIONS 118

CHAPTER 6 A CHECKLIST FOR IDENTIFYING MISFEASOR MONITORING OPPORTUNITIES 120

6.1 INTRODUCTION 121

6.2 RELEVANT ENTITIES 122

6.2.1 Content *123*

6.2.2 Policy *124*

6.2.3 User Details *124*

6.2.4 Application/Command Capability *125*

6.3 APPLICATIONS WITH NO DIRECT ACCESS TO FILES AND DATABASES 126

6.3.1 Communication function *127*

6.3.2 User/Registry management function *128*

6.3.3 Configuration function *128*

6.3.3.1 Security-related functions 129

6.4 APPLICATIONS WITH DIRECT ACCESS TO FILES AND DATABASES 129

6.4.1 File managers *130*

6.4.1.1 Propagation 130

6.4.1.2 Move 131

6.4.1.3 Rename 131

6.4.1.4 Delete 132

6.4.2 Databases *132*

6.4.2.1 Static databases 132

6.4.2.2 Dynamic databases 135

6.4.3 Access to entire files *142*

6.5 THE SPECIAL CASE OF BROWSERS 143

6.5.1 External Content *143*

6.5.2 Internal Content *144*

6.6 SOFTWARE DEVELOPMENT TOOLS 144

6.7 CONCLUSIONS 144

CHAPTER 7 CONCEPTUAL ARCHITECTURE FOR A MISFEASOR MONITORING SYSTEM **146**

7.1 INTRODUCTION 147

7.2 OVERVIEW OF THE CONCEPTUAL MONITORING TOOL 149

7.2.1 Statistical Analysis *150*

7.2.2 Inferential Analysis *150*

7.3 COMPONENTS OF THE MISFEASOR MONITORING SYSTEM 151

7.3.1 Parser *153*

7.3.2 Fact Processors *154*

7.3.3 Alert Generator *154*

7.4 APPLICATION UTILISATION MONITOR 156

7.5 INTERNET ACCESS 158

7.6 CONFIGURATION CHANGES 160

7.7 ISSUES REGARDING FILE USAGE 165

7.7.1 File Access *168*

7.7.2 File Deletion *172*

7.7.3 File Replication *175*

7.7.4 Partial replication of file content *178*

7.7.5 File Transfer *181*

7.8 DATABASE ACCESS 193

7.8.1 Registry Management *202*

7.8.2 Fraud Detection *207*

7.9 CONCLUSIONS 209

CHAPTER 8 PROTOTYPE MISFEASOR MONITORING SYSTEM 212

8.1 INTRODUCTION 213

8.2 OVERVIEW 215

8.3 EVENT GENERATOR 216

8.3.1. Events *218*

8.3.2 File Access *219*

8.3.3 File Deletions *221*

8.3.4 FileReplications *222*

8.3.5 DataReplications *223*

8.3.6 Clipboard *224*

8.3.7 DatabaseAccess *224*

8.3.8 Registry Management *226*

8.3.9 Settings *227*

8.3.10 Flags *228*

8.3.11 Data Transfers *228*

8.3.12 Data Retrievals *230*

8.3.13 Add Query *231*

8.4 KNOWLEDGE DATABASE 232

8.4.1 Employees table *232*

8.4.2 File Inventory *234*

8.4.3 Roles table *236*

8.4.4 UserRoles table *237*

8.4.5 FileAllowedRoles table *237*

8.4.6 FileAllowedEmployees table *238*

8.4.7 Machines table *238*

8.4.8 Settings table *240*

8.4.9 ServerAllowedMachines table *241*

8.4.10 Database Access *242*

8.4.11 Queries table *243*

8.4.12 QueryVerificationReference table *245*

8.4.13 Adding New Queries *246*

8.4.14 Registries table *247*

8.4.15 RegistryCustodians table *249*

8.4.16 Adding New Registries *250*

8.5 EVENT ANALYSIS ENGINE 251

8.6 ALERT GENERATOR 252

8.6.1 Arbitrary File Access Alert *254*

8.6.2 File Deletion Alert *255*

8.6.3 File Replication Alert *258*

8.6.4 Partial Content Replication Alert *259*

8.6.5 File Transfer Alert *261*

8.6.6 Database Access Alert *265*

8.6.7 Registry Access Alert *266*

8.6.7.1 Identifying appropriate authority for verification 266

8.6.7.2 Ensuring existence 268

8.6.7.3 Status check 269

8.6.7.4 Ensuring equality 269

8.6.8 Arbitrary Settings Alert *270*

8.7 CONCLUSIONS 272

CHAPTER 9 CONCLUSIONS **275**

9.1 ACHIEVEMENTS OF THE RESEARCH 276

9.2 LIMITATIONS OF THE RESEARCH 278

9.3 SUGGESTIONS AND SCOPE FOR FUTURE WORK 279

9.4 THE FUTURE OF MISFEASOR MONITORING 281

REFERENCES 282

APPENDIX A – EVALUATION OF PROTOTYPE 300

APPENDIX B – LIST OF PUBLICATIONS 348

List of Tables

Table 2.1 Annual losses for selected incidents from CSI/FBI surveys 30

Table 5.1 Cheswick & Bellovin's seven categories of attacks 99

Table 5.2 SRI Neumann-Parker Taxonomy 100

Table 5.3 Extension of SRI Neumann-Parker Taxonomy 101

Table 5.4 Detection Oriented Classification of Insider IT Misuse 107

Table 7.1 Active Window Log 157

Table 7.2 User Application Usage Log 157

Table 7.3 Application Utilisation Characteristics 158

Table 7.4 Application Utilisation Reference Thresholds 158

Table 7.5 Internet Sites Accessed Log 159

Table 7.6 Address Reference 159

Table 7.7 Bandwidth Usage Statistics 159

Table 7.8 Bandwidth Usage Norms 160

Table 7.9 Configuration Changes Log 161

Table 7.10 Flags Associated With Configuration Change Event 161

Table 7.11 Application Configuration Policy 162

Table 7.12 File Inventory Table 166

Table 7.13 File Access Log Table 169

Table 7.14 File Inventory Table 170

Table 7.15 File Deletion Log Example 173

Table 7.16 File Replication Log 176

Table 7.17 Content Replication Log 179

Table 7.18 File Transfer Log 187

Table 7.19 List of Insiders 187

Table 7.20 Users allowed to receive each inventoried file 188

Table 7.21 Roles allowed to receive each inventoried file 188

Table 7.22 List of internal machines 188

Table 7.23 List of machines allowed to receive files from each server 188

Table 7.24 Database Access Log 196

Table 7.25 Data required for identifying the affected record 197

Table 7.26 Data required for identifying the reference record 198

Table 7.27 Registry Update Log 203

Table 7.28 List of registries to be monitored 204

Table 7.29 Information to locate appropriate authority to be alerted 205

List of Figures

Figure 3.1Computer Security Model 42

Figure 4.1 Access Control vs. Intrusion Detection 61

Figure 4.2 Illustration of an Organisation's IT Network 66

Figure 4.3 Interaction of Entities at Operating System Level 74

Figure 4.4 Neural Nets in Intrusion Detection 86

Figure 6.1 Relationship between the Entities Involved 123

Figure 7.1 Basic Components of IDS 149

Figure 7.2 Overview of Misfeasor Monitoring System Components 153

Figure 7.3 Illustration of the entities involved in data transfer 183

Figure 7.4 Relationship between affected record and reference record 195

Figure 8.1 Components of the Prototype Misfeasor Monitoring System 215

Figure 8.2 Event Generator Main Interface 217

Figure 8.3 Events Interface 219

Figure 8.4 File Access Log Generator Interface 221

Figure 8.5 File Deletion Log Generator Interface 222

Figure 8.6 File Replication Log Generator Interface 223

Figure 8.7 Database Access Log Generator Interface 226

Figure 8.8 Configuration Change Log Generator Interface 227

Figure 8.9 File Transfer Log Generator Interface 230

Figure 8.10 Data Retrieval Log Generator Interface 231

Figure 8.11 Interface for Adding Queries to be Monitored 247

Figure 8.12 Interface for Adding New Registries to be Monitored 250
Figure 8.13 Misfeasor Activity Alerts Main Interface 253
Figure 8.14 Arbitrary File Access Alert Interface 254
Figure 8.15 File Deletion Alert Interface 256
Figure 8.16 File Details Interface 257
Figure 8.17 File Replication Alert Interface 259
Figure 8.18 Content Replication Alert Interface 261
Figure 8.19 File Transfer Alert Interface 263
Figure 8.20 Registry Modification Alert Interface 267
Figure 8.21 Arbitrary Settings Alert Interface 271

Glossary

CIA	Confidentiality, Integrity, and Availability
DAC	Discretionary Access Control
DoS	Denial of Service
IDS	Intrusion Detection System
IP	Internet Protocol
ICMP	Internet Control Messaging Protocol
MAC	Mandatory Access Control
OS	Operating System
PDA	Personal Digital Assistant
RBAC	Role-Based Access Control
TCP	Transfer Control Protocol

Acknowledgements

The first two years of the research programme was funded by the Engineering and Physical Sciences Research Council (EPSRC), and the rest of the research was funded by my father. I wish to thank both sources for their support.

I would like to express my gratitude to the following people:

- My director of studies Prof. Steven Furnell for giving me the opportunity to carryout this research. Without his support, vision, patience, guidance, encouragement, and vast knowledge, I would not have started nor completed this research.

- My supervisor Dr. Andy Phippen for his optimism, encouragement, and valuable advice.

- Dr. Paul Dowland and Dr. Nathan Clarke for discussions regarding various aspects during the research.

I would like to thank my colleagues within the Network Research Group for their support, collaboration, and friendship.

Special thanks are due to all my friends, and relatives. Without their support, humour, and encouragement, the journey would have been unbearable.

Last but not the least; I would like to thank my family: my father for his support, my brother for being a good role model, and especially my mother for her love and endless faith in me.

Author's Declaration

At no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other University award.

This study was partially financed by the Engineering and Physical Sciences Research Council (EPSRC) of United Kingdom.

Relevant scientific seminars and conferences were regularly attended at which the work was often presented; external institutions, international exhibitions were visited for consultation purposes and several papers prepared for publication, details of which are listed in the appendices.

Signed.....

Date.13/12/2007.....

Chapter 1 Introduction

1.1 Introduction

This chapter provides an introduction to the context of the research by presenting an overview of the main issues concerned with the monitoring of insider misuse. It then outlines the aims and objectives of the research, followed by a brief summary of each of the later chapters.

1.2 Monitoring insider misuse

Since the invention of the first computer, organisations of various disciplines have relied upon computing machines for solving complex problems such as calculations, code cracking, data storage, data manipulation, and statistical analysis. Today, components of an IT system include: network infrastructure, computer hardware, operating systems, and business applications. With the invention of the World Wide Web and the development of critical applications, such as database systems, web browsers, and email, organisations' dependence upon IT infrastructure has increased, and they are now used to support many aspects of business.

Along with the commercial success of the World Wide Web, the Internet has become one of the mediums where large numbers of business transactions are carried out daily. Many organisations conduct their business through the web, and provide web-based services to their customers. This attracted both the attention of the customers, and also malicious hackers, resulting in increased level of attacks upon the public facing servers (Power 2002). External hackers may perform protocol exploitation attacks against the public

facing server, or exploitation of server applications to gain higher privileges within the server system to gain a foothold. However, abuse of granted access by organisation's employees (i.e. misfeasance) has also become a significant issue (Gordon et. al. 2006).

The increased dependence upon IT systems, and the threat posed by external attackers and insider misuse highlighted the need to protect the systems and data. This has resulted in the development and employment of security tools and mechanisms, such as virus scanners, firewalls, and encryption tools, in order to protect the systems and data. Despite the employment of preventive security mechanisms and tools, the attackers still managed to penetrate the IT networks and continue to compromise the systems and avoid detection for long periods until the system administrator notices log entries that indicate the system security may have been compromised. However, it is impractical, if not impossible to manually check log entries in an organisation with a large network of computers. To solve this problem, Denning (Denning 1987), introduced the idea of automated log analysers, which became widely known as Intrusion Detection Systems (IDS). Denning's idea was that when the preventive mechanisms fail, the IDS would analyse the log data related to network and systems to detect indications of possible security breach. The concept of IDS has been widely accepted since then and many IDS tools have been commercially developed in order to improve the monitoring of network and host security. In 2006 the CSI/FBI survey indicated that 69% percent of 616 respondents employed IDS products in their organisations (Gordon et. al. 2006)

However, traditional IDS were designed in order to detect type of attacks usually carried out by outsiders. Thus, any inherent ability to detect misuse by authorised users would be a coincidence, rather by design. Nonetheless, it is conjectured that theoretically existing detection technologies and monitoring strategies can be adopted for monitoring of insider IT misuse.

The hypothesis behind this research is that although existing IDS technologies can be applied to insider misuse monitoring, IDS needs to be provided with the relevant knowledge and information for analysis before it can detect misfeasor activities.

1.3 Aims and objectives of research

The goal of the research is to determine whether existing detection technologies employed in traditional intrusion detection systems can be utilised for detecting insider misuse activities, and if so how they may be applied for successful detection of insider IT misuse. The main objectives of the research can be summarised as follows.

1. Identify the motives of misuse and the type of activities that may result in misuse.
2. Identify existing detection technologies and strategies that can be applied to insider misuse monitoring.
3. Investigate and analyse the ways in which insider misuse incidents can be most effectively identified and monitored at a technology level
4. Identify applications/operations and relevant contextual information required for identifying each type of misfeasor activity

5. Design and evaluate a generic conceptual misfeasor monitoring system

1.4 Thesis Structure

Chapter 2 presents a general overview of the insider misuse problem in order to create awareness, explaining the meaning of insider in the context of the research; defining the scope of research and describing the kind of activities that will be considered for the research. This is followed by identification of motivations behind misuse, and analysing the source of the problem i.e. what created the opportunity for misuse? The chapter concludes by listing the factors that created the opportunity for misuse.

Chapter 3 considers the factors that presents the opportunity for misuse, and outlines the list of technical and non-technical controls, which if implemented can dramatically reduce the likelihood of some of the misuses occurring. It then presents the preventive mechanisms currently available to protect the data and systems, highlighting the factors crucial to maintaining system and data security. The chapter concludes by reasoning why preventive mechanisms alone cannot assure system security, and emphasises on the need for monitoring mechanisms.

Chapter 4 compares and analyses current IDS in terms of the architecture, technologies utilised, and the strategies employed for detection of misuse. This is then followed by a discussion of their relevance and limitations with regard to monitoring insider misuse. The chapter also evaluates the information analysed by

current IDS and their relevance in detecting misfeasor activity. The chapter ends with highlighting the need to analyse relevant data for each type of misuse.

Chapter 5 starts with a review of current intrusion taxonomies, and presents a detection-oriented classification of insider misuse. This outlines where the relevant data for analysis may be collected within the various levels (i.e. network, system, application, and data) of the IT infrastructure. It concludes by emphasising the need to provide the monitoring system with the log data of user interaction within the application environment.

Chapter 6 starts by highlighting the need to identify information, applications and functions that require misfeasor monitoring. It then presents a checklist for identifying applications and functions that require misfeasor monitoring. It concludes by explaining why acceptable usage policy (regarding data, and operation) and contextual reference data is required in order to detect abuse of legitimate access.

Chapter 7 presents the conceptual architecture of a novel Misfeasor Monitoring System, describing how existing technologies and strategies fit within the architecture. It then explains how the logs, policies, and contextual information will be analysed for decision-making during the monitoring process.

Chapter 8 describes the components of Misfeasor Monitoring System demonstrator, detailing how the contextual information provided is utilised. It then demonstrates the applicability of existing detection technologies, and strategies to misfeasor monitoring by evaluating the demonstrator system against a number of misfeasor activities.

Chapter 9 presents the main conclusions of the research carried out, outlining the principle achievements and limitations of the work, as well as suggestions for further improvements.

Chapter 2 Nature of Insider IT Misuse

2.1 Introduction

Insider IT misuse is a broad subject, and it is not possible to cover all aspects of misuse activities that may be performed by employees of the organisation. This chapter will present the focused scope of the research, and define the meaning of insider, and misuse within the context of the thesis.

2.2 Definition of Insider

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 and services it provides. Users may be physically located insider or outside the organisation, but have the same logical presence within the IT system. For example, a telecommuting user who works from home by connecting to the organisation's network via VPN has logical presence on the system, but not physically inside the organisation where servers are located. By contrast, some individuals may be physically inside the organisation, but lack a valid user account to access the systems. For example, a cleaner who has physical access to the building and the offices within it, but do not have a valid user account in order to access the computers located in the building. In this context, they are regarded as logical outsiders, and for the purpose of this research the term 'insider' refers to users with valid login accounts (i.e. the logical insiders), and do not consider physical insiders. Although, one thing to note is that once and individual has gained access to a system,

whether the individual is legitimately authorised or not, it is very difficult to differentiate between legitimate user and illegitimate user from the system's point of view.

2.2.1 The scope of insider within the thesis

Logical insiders i.e. authorised users of the system with legal login accounts to access the organisation's IT systems. Physical insiders who have no logical presence are not considered for monitoring, i.e. theft of hardware or wiretapping is not within the scope of the research. It is also assumed that the BIOS is protected to prevent the system from loading an alternative operating system, and physical security is present to prevent addition of unauthorised hardware (such as a wireless network card).

2.3 Different categories of IT misusers

Although the great majority of the people are familiar with the generic meaning of the word 'misuse', when attempting to map it to an IT context, there is a need to clarify certain issues. Insider IT misuses can be a very subjective term, and one of the most challenging tasks is to draw a clear line that separates an IT misuser from a person who is using a system in an acceptable way and for an approved purpose. 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. Although this is not difficult to grasp, vagueness is introduced by the term misuse and what it means to different people or organisations. What is considered illegitimate use in one particular organisation can be perfectly acceptable for another. For example,

browsing the web for personal use is outlawed entirely in some companies, whereas others are somewhat more relaxed about it and impose varying limits upon what is acceptable (e.g. some may permit up to 20 minutes per day, whereas others may allow twice this). However, such a policy, and hence the definition of misuse, can differ from one organisation to the other. Thus no single definition of misuse is appropriate for all organisations.

The term "misuse" is interchangeably used for both outsider attacks and insider abuse. In addition, the nature/methods of insider misuses can also differ. Therefore, the usage of the term within the context of this research needs to be made clear.

Anderson (1980) provided classification of internal security breaches within an IT context. Anderson divided internal misuses into three distinct categories:

- *Masqueraders*: These are internal users, who have exploited the flaws within the operating system to impersonate the identity of another user, and thus evading accountability. When this occurs, the malicious activities the masquerader has carried out will be audited under the identity of the user the perpetrator has impersonated. Some of the privilege escalation attacks fall within this category. This type of misuse is not considered within the scope of this research.

- *Clandestine users*: These are internal users, who evade auditing by operating at a lower level than where the actual auditing occurs. For example, if the auditing occurs at the application level, and the user accesses the data through an arbitrary application that does not provide auditing facility, detection can be avoided. It is very difficult to detect what actually happed to the system and data stored on it, when the user has evaded auditing. Therefore, avoiding detection for a longer period of time, and by the time someone notices that something is wrong with the system/data; the perpetrator might have left the organisation. This emphasise the need to monitor the application utilised for the access of file (resources) or performing an activity. IMMS requires applications to provide audit data in order to analyse user activity within the application environment. Thus, if the user accessed the file (resource) through an arbitrary application, auditing may be avoided. This turns a misfeasance (legitimate operation at the operating system level) to a clandestine activity.

- *Misfeasors*: These are the internal users, who abuse their existing system and application level privileges by utilising the system and resources in an inappropriate manner or for unapproved purpose. This can happen when the user has been assigned more privileges than actually necessary to carryout his/her daily tasks. The users may be assigned higher privileges by mistake, or because the access controls are not granular enough and hence forcing the system administrator to assign the user slightly higher privileges than necessary.

However, even when the principle of lease privilege is applied, such privileges may still be abused, such as transferring a confidential document to someone not authorised to access, or configuring the system in such a way that the security is weakened.

Various other classifications of security breaches exist, and they will be detailed in Chapter 5.

2.3.1 Scope of misuse within the thesis

Within the scope of this research, only misfeasor activities are considered for monitoring and clandestine users, or masqueraders are not included. Prior to proceeding further, the definition of misfeasance need to be made clear. Anderson's (1980) definition of misfeasance is fairly technical and not easily understood. A less technical and more general definition of "misfeasance" is defined in Microsoft Encarta World English Dictionary as:

> *Misfeasance: "illegally performing something legal: acting improperly or illegally in performing an action that in itself is lawful"*
>
> Microsoft Encarta World English Dictionary (Encarta 2005)

Therefore, interpretation of misfeasance in the scope of this research is that the operation itself is lawful when viewed from the perspective of the operating system and access

controls, yet unlawful when view from the perspective of the application/business context and organisation's code of conduct. However, the acceptable usage policy may also differ from one organisation to another. For example, some organisation may permit/encourage information sharing among colleagues, while others may deem such a practice unacceptable. The acceptability of sharing information may also depend upon the content, the users involved, the responsibilities of the users involved, and the tools (machine, and application) utilised for access.

2.4 Types of misuses

Various legitimate activities carried out by authorised users may result in misuse within the context of the acceptable usage policy defined by the organisation. This section presents the nature of activities that may violate the acceptable usage policy of some organisations. The categorisation of misuse activities in this section have been derived from the triennial Audit Commission Computer Fraud and Abuse surveys. The categorisation here is based on the potential consequences of the activities, while the classification in Chapter 5 is based upon the level of the system at which the detectable evidence may manifest itself. The nature of activities presented in this section is considered on the basis of their potential impact upon the organisation's finance, reputation, productivity, and legal liability.

2.4.1 Internet access abuse

The organisation can be held liable for the external content accessed/downloaded by its employee. Not only can result in loss of employee productivity (Haines 2007), but the downloaded images, video, and audio can bring disrepute to the organisation. If the accessed multimedia content breached copy rights, the organisation may be held liable. The websites accessed and the content downloaded may also introduce malware to the organisation's IT systems. The file types requiring investigation are vocal, instrumental and visual content. In terms of downloading, no organisation to date has been held liable for accessing inappropriate textual content by its employees. However, uploading inappropriate content of all types (textual, vocal, instrumental, visual) by its employees can implicate the organisation.

In 2004, the Department of Work and Pensions was shamed by the discovery of pornographic pictures on organisation's computers. The investigation revealed that over two million pornographic pages were accessed within a period of eight months, and more than 18,000 of the images and sites accessed involve child abuse (Lea 2004). This undoubtedly damaged the reputation of the organisation, and led to the sacking of sixteen employees, and two hundred disciplinary cases.

2.4.2 Information theft

Espionage of political and corporate nature has existed long before the utilisation of IT systems by organisations for the management of its information. However, within non IT

environment, the amount of information stolen can be limited, physical activity can be noticed by colleagues, and physical security may deter documents being stolen. If an employee wanted steal information he/she must memorise the information little by little each time to take home, steal the original copy or photocopy the documents and attempt to pass through security checks to take it out of the premises. If the perpetrator chose to memorise the information, it may be less accurate, and it will take time. If the perpetrator chose to steal the original copy of the information, someone else needing access to the information may notice it missing. If the perpetrator chose to photocopy the documents using one of organisation's photocopiers, colleagues may notice and arouse suspicion and leading to a closer investigations. However, within IT environments large volumes of information may be stolen within a short period. User activity within the system cannot be easily noticed by colleagues. Information can be easily duplicated without anyone noticing, the original copy won't be missing from its original location for a second and still accommodating access by other users. Information can be transferred through the network, bypassing physical security checks. When connection to the external network is not available, the information may be hidden within removable storage media prior to sneaking it out of the premises.

Recently, a former employee (Nigel Stepney, head of performance engineering) of the Formula One team Ferrari has been accused of providing proprietary information to the chief designer of the rival McLaren team. The 500 page technical documents containing Ferrari proprietary information were found at the McLaren chief designer's home, and the

computers are being examined by experts for forensic evidence (BBC 2007a). Such a feat would have been very difficult without the utilisation of IT systems. The employee would have had to memorise the information in small pieces, and the accuracy of the information could be in doubt. Assuming physical security is present, it would be very difficult to steal a 500 page document in one go without the use of IT.

2.4.4 Breach of privacy/confidentiality

Inappropriate access of organisation's databases can result in breach of confidentiality and privacy. Targets of abuse include:

- Organisation's business database
- Organisation's employee database
- Organisation's customer database

In March 2000, a police officer in North Queensland, Australia, admitted utilising the police database for personal reasons (Taylor 2000). The 20 year old police officer performed 6,900 searches on the police database within a period of two months, and at least 300 of the searches were not related to official work. The perpetrator used the police database to check for potential girlfriends, domestic violence within his neighbourhood, old school friends, neighbours, and government cars he wanted to buy.

The above example is a classic case of misfeasance, where the perpetrator has authorised access, the operation itself is legitimate, and part of the perpetrator's work. Although, none of the system level access controls were violated, and the operation itself is legitimate, the context in which the operation was performed violated moral and ethical conduct. The fact that system level controls were not violated, the operation itself is legitimate and part of the perpetrator's responsibilities makes detection of misfeasance more difficult in comparison to detection of outsider attacks.

2.4.5 Sabotage

Sabotage can result from various activities, such as deletion of important files, writing bad data to corrupt information, or deliberate misconfiguration of the system to compromise security. The consequence of sabotage can result in:

- Destruction of critical data (business, and customer databases)
- Disturbance of service (web, ecommerce, production, or internal data services)
- Weakening of system security (disabling virus scanners, addition of vulnerable networked services)
- Deliberately causing the application to malfunction (inappropriate configuration of application settings).

Therefore, activity verification and establishing of accountability is required.

In May 2000, a former employee of Omega Engineer Corp. who held the position of system administrator was convicted for deletion of proprietary software belonging to the organisation (Gill 2002), which resulted in US$10 million damages to the organisation. The fact that the perpetrator was a system administrator makes automated detection useless due to lack of segregation of duties. Therefore, system administrative operations should be verified by a second party, and segregation of duties needs to be employed between the person performing the operation and the person verifying the operation.

2.4.6 Fraud

Within IT systems fraudulent activities usually involve database access and entry of inappropriate data. Fraud can affect data integrity, financial loss, and damage to reputation.

In 1996 two credit union employees colluded for several months to alter credit reports in return for payment (Randazzo et. al. 2004). The employees were responsible for updating credit reports based on the information provided. However, the perpetrators abused their authorised access by removing negative credit indicators in exchange for money.

2.5 Factors leading to misuse

According to Schultz (2002), a combination of three main factors can lead to insider misuse of IT systems. The outlined factors are:

- ***Capability***

  The perpetrator must have the technical know-how, and/or a good understanding of the targeted system. The system here does not limit to the network or the operating system, it can include user application, security policy, or business rules, i.e. in order to bend the rules one must first know what the rules are and the loopholes that may be exploited.

- ***Motivation***

  Merely having the capability does not necessarily mean that the individual will misuse. The perpetrator must also have the motivation to misuse, although sometimes accidental misuses can occur. However, one thing to note is that an innocent error gone unnoticed can also lead to misuse as the individual may believe that the activity will not be noticed.

- ***Opportunity***

  Having the motivation and capability are essential ingredients for a potential misfeasor. However, the perpetrator also needs the opportunity to misuse. The opportunity may present itself in the form of technical vulnerabilities (protocol, system, and policy), lack of segregation of duties by management, or loopholes within business rules. The opportunity may sometimes appear in the form of an error gone unnoticed (Coderre 1999). Possibly due to lack of

security and acceptable usage awareness training, many of the misfeasors did not consider the negative consequence that may result from their activities (Randazzo et. al. 2004).

When the three factors combine, the potential for occurrence of misuse becomes higher. If the exploitation of loopholes is involved detection may be avoided for long periods, and only detected by accident or when management changes.

2.5.1 Capability

Depending upon the role of the employee, and the associated responsibilities, the capability and the nature of misuse may differ (Magklaras and Furnell 2002). System administrators have the capability for sabotage and data theft, accountants have the capability for financial fraud, sales representatives have the capability to steal customer information, data analyst have the ability to access sensitive business data, and common users have the ability to abuse internet access. Therefore, almost every system users have the capability to misuse in one form or another, although the severity of the actions can differ.

2.5.2 Motivations behind misfeasance

First of all, it is important to understand the motivations for insider misuse in order to identify the type of applications that are most likely to be misused, and the nature of information that is subject to misuse. It has been identified that the main motivations

behind insider misuse can be categorised in to that of financial gain, ego, vengeance, and recreation. The motivations identified are based on the study carried out by Shaw et. al. (1998b). Some of the scenarios described here are manifestation of each motive within IT systems in order to gain insight for identifying the type of activities and the nature of data targeted for each motive.

Intentional misfeasor cases are performed for a variety of reasons including greed, revenge, ego gratification, express anger, impress others, to protect or advance career, or a combination of the motives mentioned (Shaw et. al. 1998a). A suitable way to sub-divide them is to consider the motives in a way that could detect the ultimate goal of the abuser. It might be inferred, for example, that a legitimate user is trying to access sensitive data (data theft), take revenge against a particular person or an entire organisation (personal differences), cover indications of unprofessional behaviour, or deliberately ignore a particular regulation of the information security policy.

2.5.2.1 Financial Gain

In this category, the activity carried out results in direct financial gain to the individual. The users may commit fraud within financial systems, or steal proprietary/confidential information contained within files or database systems. The nature of content may differ from one organisation to another. Examples of important files include, source code, research documents, product designs, contracts, and internal memos. Examples of financial systems include payroll, inventory control, stock management, order

management, invoice systems, claimant systems, accounting systems, and banking systems. Examples of information databases include customer databases, product databases, financial records, salary information, personal records, criminal databases, supplier records, and technical data.

Under certain circumstances, the misfeasor may only have to create opportunity so that someone else can steal valuable information. An example of this is, knowingly installing a Trojan program on a machine, which has access to important information, so that the creator of the Trojan program can access the information. Another example of this kind is, intentionally weakening the security of the machine or disabling a monitoring system, which has access to confidential information, so that someone else may compromise the machine in order to access the desired information.

2.5.2.2 Ego and Pressure

In the cases of Nick Leeson (1997), and Iguchi (Dowell 1997), their initial intentions were to hide small errors in order to save face and live up to expectations. However, in order to recover the losses they took higher risks, which led to more loss.

In the case of Jett (Dhillon and Moores 2001), it may have been the pressure to perform, although motivation may also involve financial gain through performance bonuses.

Therefore, databases referenced for performance monitoring and bonus calculation are subject to fraudulent modifications.

2.5.2.3 Vengeance

In this category, any activity carried out by the individual may not result in direct financial gain, but nonetheless give a sense of power and satisfaction to the culprit. However, the result of the activities will be damaging to the organisation. The affect may be on products, organisation's image/reputation, or productivity. Targets of abuse are all valuable files, documents, records, and services. These are subject to disturbance, destruction, and undesired exposure. The culprit may delete files containing valuable content, such as source codes, and product designs. Important records such as customer data and financial records may be doctored to compromise the integrity of valuable data. Systems, Applications, and other mechanisms within the organisation may be attacked in various ways to halt/delay productivity. Web services provided by the organisation may become subject to DoS attacks, rendering the organisation's services unavailable to customers. Therefore, any application or service that is essential to the day-to-day functioning of the organisation is subject to abuse, and any information that is valuable or potential embarrassment to the organisation is subject to exposure. In certain cases, the user may not actually carry out the attacks, but rather create an opportunity for someone else to create havoc within organisation's IT systems.

2.5.2.4 Recreation, Breach of Privacy, Curiosity, Social Engineering, Naivety, Accidental Misuse, Un-Wittedly Running a Trojan Application

In this category, the activities may not result in significant financial gain to the individual. However, it may still result in damage to organisation's assets, reputation, and productivity as side effect. While surfing the web, users may access websites that contain inappropriate content that may contain viruses, and Trojans that may affect organisation's IT systems. Some of the content accessed by users such as pornography may also damage organisation's reputation, and sometimes the organisation may be held legally liable. In addition, while the users are surfing the web, writing personal emails, perusing personal emails, downloading entertainment media, and playing online games, the user are unable to carry out productive work, resulting in reduced overall productivity (Carr 2005). A study based on 3,500 UK companies revealed that the users who visit social network sites during work hours may be costing firms over £130 millions a day (BBC 2007b). Web browsers, emails, and other communication programs are becoming essential to many organisations, and monitoring user activity within these environments are important. However, the users can be prevented from installing undesired programs, such as computer games, and entertainment applications.

2.5.3 Opportunity

Insider misuse occurs when a ready mind meets an opportunity (Tuglular 2000, Audit Commission 1994). Therefore, circumstances that present opportunity for misuse needs

to be identified. The opportunity may arise due to weakness or lack of managerial and technical controls (Cappelli et al. 2006) and the situations in which this can arise are:

- *Lack of awareness by users*

  Sometimes misuse may result from users being unaware of the security culture or the acceptable usage policy of the organisation. Employees may share passwords, or critical files, when they are not provided with security awareness training.

- *Lack of properly defined security (or acceptable usage) policy*

  For example, employees may abuse Internet access inadvertently when they are not provided with the acceptable IT usage policy guidelines.

- *Failure to define proper segregation of duties*

  The opportunity to misuse granted privileges arise when the person performing an operation is also responsible for verification of the activities. This principle is applicable across all systems, environments, and organisations. This part of the problem is at the managerial level of the organisation (assuming the system environment facilitates segregation of duties), and formal segregation of duties needs to be defined.

- *Failure to enforce segregation of duties*

  When an IT system administrator is responsible for all aspects of system maintenance (including development of security policy, and detection of security breaches), and a second party cannot be involved for verifying the activities of the system administrator, opportunity to abuse trust arises. Technical limitations of the system environment may prevent enforcing segregation of duties, which will be discussed in detail in Chapter 3.

- *Failure to enforce principle of least privilege*

  Sometimes, the opportunity for misuse arises when the users are assigned more privileges than required to perform their daily duties (Brackney and Anderson 2004). In such cases, the users may have authorised access to confidential files that is not required to perform their assigned tasks, or have the privilege to perform operations that is not part of their responsibilities. This kind of situations can arise when security policy is poorly defined, or implemented, which may be due to lack of understanding by the person responsible for assigning privileges. Therefore, it is essential that the assigned privileges and access rights are verified to ensure conformance to need to know/perform basis.

2.6 The scale of misuse

If one takes a look back to computer crime literature and surveys dating up to the mid-90s, the evidence presented would certainly suggest that the main threat was to be found

from one's own staff, with as much of 80% of computer crime believed to be the result of insider activity (Power 1995). In more recent years, however, many sources have indicated a significant rise in externally sourced incidents (principally in terms of Internet-based attacks such as website defacement and denial of service), with the consequence that although insider misuse is still significant, it now accounts for a far lesser proportion of raw incidents. For example, in the UK, results from the Department of Trade & Industry's Information Security Breaches Survey 2006 revealed that overall only 32% of businesses considered their worst security incident to have been caused by an insider (DTI 2006). However, when considering the large businesses (with over 250 employees) only, 52% considered their worst security incident to have been caused by an insider.

Another source that has monitored the changing trend regarding internal and external attack is the annual CSI/FBI Computer Crime and Security Survey. Looking back to 1995, a key observation from the CSI was that "the greatest threat comes from inside your own organisation" (Power 1995). In more recent years, however, the survey results have painted a rather different picture, and by 2005 it was reported that, incidents originating from both inside and outsider are almost equal (Gordon et. al. 2005). This indicates that insider incidents require as much attention as those coming from outside. In addition CEOs of most organisations will be more interested in the effect that the incidents had on their bottom line. In 2006 CSI/FBI survey 7% of the respondents

considered that more than 80% of the loss is accountable to insider incidents (Gordon et. al 2006).

Many of the categories used in the CSI/FBI results encompass incidents that could potentially have been both internally and externally sourced (e.g. theft of proprietary information, sabotage of data networks, and virus). However, three of the categories very clearly indicate the source, and it is interesting to see the level of the annual losses that were associated in each case. The relevant information is presented in Table 2.1.

Supporting results from the ICT Fraud and Abuse 2004 survey (Audit Commission, 2005) also reveal that the majority of the perpetrators (over 80%) originated from inside the organisation, with operational staff 37%, administrative/clerical staff 31%, and managers 15%.

It is quite evident from the results that, although the proportion of externally sourced incidents had exceeded internal ones, the quantifiable losses in the latter case are significantly higher than those attributable to outside hackers. It is therefore clear that, in real terms, the level of the insider threat is still much greater than that exhibited by external hackers.

Year	System penetration by outsider	Insider abuse of Internet access	Unauthorised insider access
1998	$1,637,000	$3,720,000	$50,565,000
1999	$2,885,000	$7,576,000	$3,567,000
2000	$7,104,000	$27,984,740	$22,554,500
2001	$19,066,600	$35,001,650	$6,064,000
2002	$13,055,000	$50,099,000	$4,503,000
2003	$2,754,400	$11,767,200	$406,300
2004	$901,500	$10,601,055	$4,278,205
2005	$841,400	$6,856,450	$31,322,100
2006	$758,000	$1,849,810	$10,617,000
Total	$49,002,900	$155,455,905	$133,877,105

Table 2.1 Annual losses for selected incidents from CSI/FBI surveys

The CSI figures relating to insider abuse of network access clearly suggest that, as well as bringing considerable advantages in terms of web and email communication, Internet access has also ushered in a whole range of new problems. This can be further evidenced by a survey of 544 human resources managers, conducted in 2002 and targeting large UK companies (with 'large' in this case being defined as those employing an average of 2,500 people). The results revealed that almost a quarter of them (23%) had felt obliged to dismiss employees in relation to Internet misconduct (with the vast majority of these cases – 69% - being linked to the downloading of pornographic materials) (Leyden, 2002). Many other cases resulted in less severe courses of action, such as verbal

warnings or a discreet word in the ear of the person concerned, and in total the results indicated that 72% of respondents had encountered Internet misuse in some form.

In 2002 Information Security magazine survey, 23% of respondents rated authorized users as their most important problem, while 11% reported unauthorized users as their most important problem (Briney and Prince 2002). Similarly, results from the Department of Trade & Industry's Information Security Breaches Survey 2002 revealed that 34% of businesses considered their worst security incident to have been caused by an insider (DTI 2002). Indeed, the fact that insiders are already within the organisation often puts them in an ideal position to misuse a system if they are inclined to do so. The insider abuse can be more damaging than many outsider attacks, 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 (Einwechter 2002).

2.7 Famous Cases

Although mainstream media have focused mainly on viruses, worms, and hacker attacks, there have been a few insider cases that gained fame or notoriety in recent years. The cases that follow demonstrate that although insider incidents may be rare, the cost of a single incident can have a significant impact.

2.7.1 Joseph Jett vs. Kidder Peabody (Dhillon and Moores 2001)

Joseph Jett was an employee of securities bank Kidder Peabody. His responsibilities included arbitrage of bonds. Kidder Peabody relied heavily on expert systems for the valuation of transactions in the bond market, and for automated calculation of loss and profits. Jett had good understanding of how the bond market, and how the expert system at Kidder evaluates loss and profits. Jett realised that by performing forward transactions, the time for registering losses in lost and profit statements will be postponed indefinitely, generating profits only. As a result of phoney profits from his trades, Jett earned more than $14million in salary and bonuses, and his activities remain undetected for two years. He was only charged with record keeping violations, and avoided charges of fraud.

Jett's case is a great example of misfeasor behaviour within IT environment. He was authorised for all the activities he carried out. He was performing legitimate operations, yet in an unexpected and inappropriate manner. He did not violate network, operating system, or application level controls. He merely exploited the loopholes regarding the evaluation of transactions with the understanding of the bond trading market.

2.7.2 Nick Leeson vs. Barings Bank (Asiaweek 1995)

Nick Leeson was an employee of Barings bank, and his responsibilities included managing operations on futures markets in Singapore Monetary Exchange. He

was responsible for both trading and recording his trades, lacking proper segregation of duties; an unusual practice in the banking industry. Leeson gambled on highly speculative markets without authorisation, and at first he was making large profits. However, his luck ran out, and Leeson hid his losses totalling more than $1.4 billion in an account numbered 88888. Although there is not much technical involvement in this case as in Jett's, this example demonstrates the importance of formal segregation of duties and that insider misuse is a management problem as much as technical.

2.7.3 Tim Lloyd vs. Omega (Gaudin 2000)

Tim Lloyd was a trusted employee of Omega for over ten years. He was responsible for planning and building of Omega's first computer network for its Bridgeport manufacturing plant. He was in sole charge of all the network administration duties including backing up data and management of backup tapes. A week before he was fired, Lloyd asked all the users for all the programs stored on local systems and moved to the server. After he was fired, Lloyd planted a time bomb that deleted nearly every program stored on the server. The programs deleted were used for the manufacturing of 25, 0000 different products. Lloyd had also stolen the backup tapes and formatted them. This led to delayed production, and resulted in losses of around $10 million.

This case demonstrates the perils of lack of segregation of duties at both organisation level and within the IT environment. Lloyd did not violate any system level policies, because he made the policies, and he was responsible for making system level policies. His activities could not have been detected, because he was responsible for verifying his own operations. He was only charged for theft of backup tapes.

2.7.4 Robert Hanssen vs. FBI

Robert Phillip Hanssen was an employee of the FBI, and was assigned to the New York Field Office's intelligence division in order to aid establish an automated counter intelligence database in 1979. He was charged with spying for Russia since 1985, and alleged to have given highly classified information including identity of U.S spies, electronic surveillance technology, and nuclear war plans to the Russians in return for $1.4 million and diamonds (Arena 2001). He was authorised to access the FBI's electronic case file system, and it is reported that Hanssen copied classified information on to floppy disks and his Palm handheld computer (Verton 2001). Henssen also accessed the case file system to check whether FBI is investigating him. However, because he was an authorised user his queries were not seen as suspicious.

The case of Robert Hanssen demonstrates the risk of allowing authorised users to carry PDA, and removable media in and out of sensitive office areas. It also

demonstrates the difficulty of determining possible misuse when the operation performed is part of the perpetrator's responsibilities. In the wake of Hanssen's case, a former CIA scientist Allen Thomson suggested the application of two person authorisation rule for all sensitive database searches and system operations in order to limit data compromise.

2.8 Conclusions

In this chapter the meaning of insider, and the notion of misuse within the scope of the research, has been defined. Examples of various insider misuse cases are also presented, and the factors leading to misuse is highlighted. In the next chapter, the existing measures that can be taken to prevent or reduce the risk of misuse occurring will be discussed and evaluated, in order to determine whether they can provide an adequate safeguard against the problems observed.

Chapter 3 An Examination of Existing Safeguards Against Misuse

3. 1 Introduction

Having analysed the nature of insider misuse and classification of cases on the level of the system they may have impact upon, this chapter reviews the current security controls that can be applied against insider misuse. The aim of this chapter is to investigate the weaknesses existing controls have at preventing insider misuse. Those weaknesses will inform the design of the proposed architecture, and enhance the detection of insider IT misuse.

The chapter begins with the discussion of the classic security model for operating systems and evaluation of access control at the operating system level. The analysis of access control mechanisms will aid in identifying activities that can be regulated through access controls and the ones that cannot be regulated by access control and thus require auditing and detection. It then proceeds to consider auditing and intrusion detection techniques.

From the examples given in the previous chapter, it can be noted that some of the misuses could have been prevented through managerial and technical means. Therefore, possible preventive measures need to be evaluated in order to reduce the amount of monitoring needed, and to increase effectiveness and efficiency of monitoring. Preventing when possible reduces monitoring overhead, and increases monitoring efficiency, and most importantly reduces risk. If utilised correctly, many of the preventive mechanisms can reduce the likelihood of misuse occurring.

3.2 Non-technical measures

When considering how to protect systems, it is worth noting that preventative measures need not be technical. Insider misuse is a management problem as much as it is a technical issue. For example, the insider threat study carried out by Carnegie Mellon Software Engineering Institute in conjunction with United States Secret Service revealed that, one third of the insiders responsible for sabotage of employer's IT systems had a previous arrest history (Keeney et. al. 2005), and 65% of the perpetrators did not consider the consequences that may result from their activities (Randazzo et. al. 2004). This emphasise the importance of background checking prior to employment, and security awareness training. As such, formal internal controls are as important as technical controls. Security guidelines, such as the recommendations provided by the ISO/IEC 17799 standard, typically suggest a number of personnel-related measures that, if employed correctly, can reduce some instances of insider misuse:

- Check references of prospective new employees before hiring them;
- Ensure that employment contracts include a clause relating to the acceptable use of IT resources;
- Ensure that adequate reminders about the 'acceptable use' policy are encountered by staff during their day to day use of systems;
- Ensure adequate supervision of staff by line management;
- Provide a means by which staff can confidentially report misuse of IT systems, without fear of recrimination from colleagues.

- Ensure proper division of duties at management level i.e. such that collusion between staff members would be necessary before significant opportunities for frauds could be identified.
- Enforce segregation of duties within system level, i.e. the individual authorising/verifying an operation should not be the same as the person who performed the action, such as the person responsible for administration of the system should not be the same as the person responsible for verification of system security.
- Concerning the access of data, make sure that access control policies resemble organisation's management hierarchy or rules (Ward and Smith 2002).
- Security and access control policies need to be maintained to keep up with the change in organisation's management hierarchy.

In the absence of an automated supervision approach, it would still fall to line managers and the like to enforce and monitor these aspects.

3.2.1 Physical security

Intrusion detection systems are not capable of dealing with issues related to physical security of the systems. However, some of the insiders have physical access to the systems, and physical access can provide means of getting around controls implemented at the system level. Therefore, physical security needs to complement system level security to minimise certain types of insider misuses, such as theft of storage media containing sensitive information. Security checks should be in place to prevent removable

media from being taken in and out of offices containing highly sensitive data, such as server rooms, and backup vaults. Physical locks should be applied to system box, so that hardware may not be easily added without authorisation. Removal of hard disk can result in compromise of data confidentiality. Addition of a network card can result in covert channels for conveying proprietary information, bypassing the channels monitored by the system.

3.3 Technical measures

Once the system is loaded with an arbitrary operating system, the controls enforced by the normal operating system can be bypassed. It is even truer now with the wide availability operating systems that can be loaded from Live CDs and USB keys. (This would fall into the category of clandestine activity). In such cases, the perpetrator would be able to bypass access controls and replicate the information contained on the system hard disk. BIOS level passwords need to be applied to prevent the system from loading unauthorised operating system from unauthorised media. In addition, encryption should be applied to confidential files, in case the files are successfully replicated through such approach, or through physical theft of hard disk.

The security mechanisms and monitoring will not be effective if the operating system and critical applications are compromised prior to installation. Operating systems and application programs need to be verified before being allowed to install on organisation's systems. This requires segregation of duties and multi-person verification to ensure that

the operating system or the application being installed is not compromised. Software installation capability also needs to be limited to a few persons, and verification procedure needs to be enforced. However, some of the standalone applications do not require installation, and the execution of the program file can compromise system security. Executable paths also need to be defined, in order to prevent execution of foreign programs. For example, files within C:\Windows directory and C:\Programs directory can be executed, and prevents execution of files from anywhere within the system.

3.4 Computer Security Model

First of all, basic components of computer security model needs to be examined in order to understand how security is maintained at a system level. The computer security model consists of the identification and authentication module, access control mechanism, and auditing mechanism. The components of the computer security model will be analysed to identify their relevance towards prevention and detection of misfeasance.

Figure 3.1 illustrates the computer security model of many operating systems. Most operating systems include an Identification and Authentication subsystem to identify and authenticate users, an Access Control mechanism to regulate user access of resources, and an Auditing Subsystem to log system events and user access of resources (Escamilla 1998).

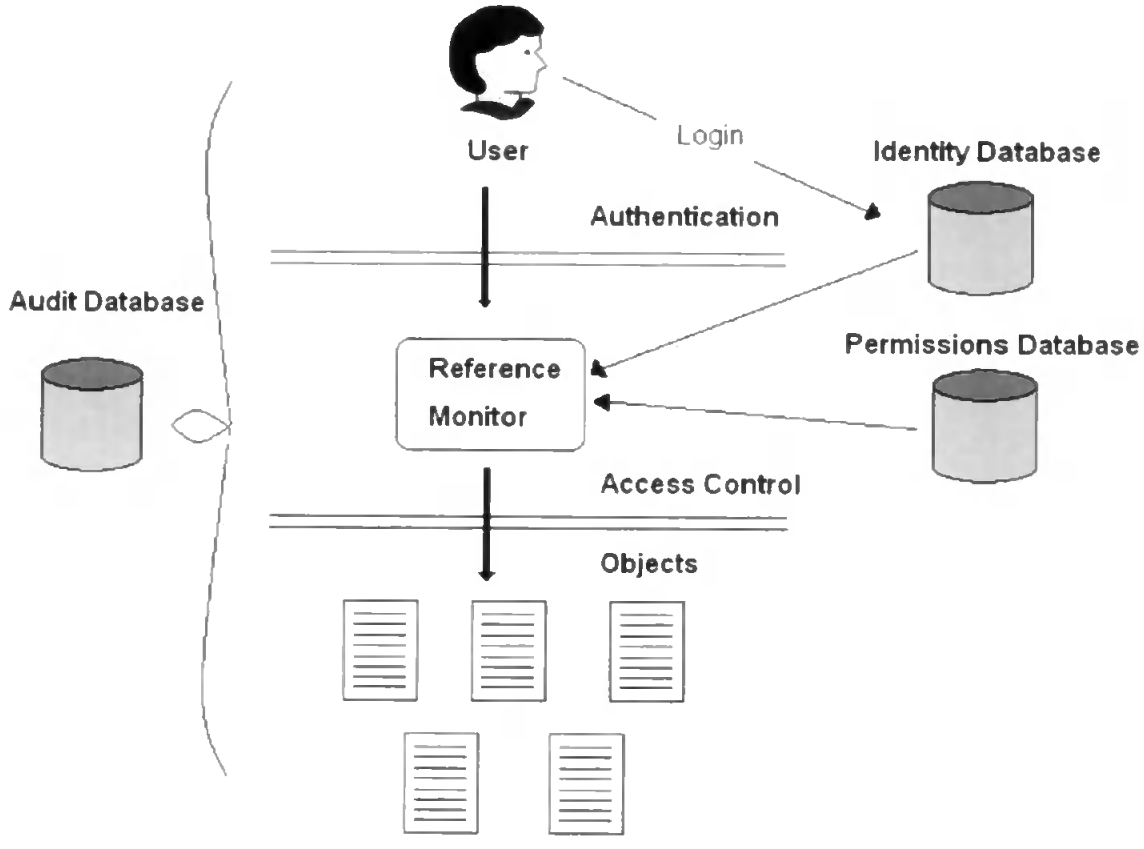


Figure 3.1Computer Security Model

In the computer security model, *subjects* access *objects* and the *reference monitor* makes the access control decisions. The *reference monitor* refers to the *permissions database* to determine permissions associated with each *object* to determine which *subjects* are allowed access to the *object* and the mode of access each *subject* is allowed. Depending upon the level in which the mechanism is implemented, the parameters considered for decision making and available modes of operation may vary. The *subjects* and *objects*

can very depending upon the level of the system in which the access is regulated. For example, at the operating system level files and system services can be *objects*, while records and queries become *objects* within database systems.

3.4.1 Identification and Authentication

It is not possible to regulate access unless the identity of the user (owner of the process requesting access) is known, and authentication procedure is also required to ensure the user is who he/she claims to be. If access is regulated on individual basis, the user's identity needs to be determined, in addition if access is regulated on group/role basis the group/role the user belongs to needs to be determined before identifying permissions associated with the user/role. The ability to establish the identity of the user(s) involved in an activity is also relevant towards detection of misfeasance, because the acceptable usage depends upon the user's responsibility within the organisation and the relationship between the user(s) and the system/data affected/involved.

3.4.2 Security and Usage Policy

Anything that is of value needs to be inventoried (Machines, Operating systems, Applications, Files, Databases etc.) before desired security, and acceptable usage policy regarding each entity can be defined. This requirement is also relevant towards misfeasor monitoring, because acceptable usage policy needs to be referenced for successful detection of misfeasance.

An ideal security policy should to be defined before attempting to implement at the system level in order to understand limitations of the access control technology at each level of the system, and to identify activities that cannot be regulated through access control and thus requires monitoring.

3.4.2.1 White listing and black listing

For any kind of regulation, a reference of what is allowed and disallowed is needed, depending upon the operation and data/system affected/involved. The parameters used or regulation also need to be directly related to the parameters referenced for determining permissions. For example, parameters used for regulation are based on IP address the reference policy must be based on IP address, and if the parameters used for regulation are based on user name the reference policy must be based on user name.

3.4.3 Access control

This section analyses whether confidentiality, integrity, and availability of the system and data can be protected from misfeasor activities through the use of access control. It is clear from previous discussions that proper segregation of duties is a critical issue regarding insider misuse, and thus access control mechanism and relevant access control policies need to be evaluated for their ability to enforce segregation of duties within the system environment. In addition dissemination of data and inappropriate deletion of files are also a concern, and the capabilities and limitations of access controls with regards to these issue need to be evaluated.

Access control mechanism provides means for regulating access of services and data and ensures that only the authorised users are allowed access to the files and only in the mode of access defined by the access policy.

Operating systems utilise file system's file allocation table for regulating file access, and security attributes associated with each file determine how granular the access policy can be. At the operating system level, the available modes of access are read, write, and execute. The mode of access affects the entire file. Depending upon how access permissions are set, two classical access control policies are available.

3.4.3.1 DAC (Discretionary Access Control)

Lampson (1971) introduced the basic ideas that lead to the development of discretionary access control. DAC as the name suggests the discretionary of the contents rest with the owner/creator of the file, and the owner of the file defines which subjects are allowed access to the file, and the mode of access allowed for each subject. However, the system administrator can access every file located within the system and is also able to override the permissions set by the owner, and this presents opportunity for privilege abuse. In addition, managing the ownership of information within an organisation is a challenge, and the issue will be discussed in 3.4.4.

The flow of information within DAC is not regulated i.e. a user having gained access to the file may transfer the contents to someone who is not authorised to access the file. It is not possible to prevent dissemination of data within discretionary controlled system. There is no representation of hierarchy within the DAC model, and thus no control regarding the direction of information flow.

One of the weaknesses of DAC is that the information can be replicated, and the creator of the replica can grant permissions to subjects who would not have access to the original. DAC is sufficient for cooperative environments such as academic research, but does not satisfy the requirements of many commercial enterprises (Ferraiolo et. al. 1993).

3.4.3.2 MAC (Mandatory Access Policy)

Mandatory access control was introduced by (Bell, LaPadula 1975), to enforce lattice-based security policies to thwart Trojan-Horse attacks, and to regulate direction of information flow within the system.

Within MAC model, the system enforces the access permissions overriding the policy set by the creator of the file. Access is regulated using the security label associated with the file, and the clearance level of the subjects. The clearance level of the subjects represents a hierarchical order within the MAC model. For example, the clearance levels may consist of Top Secret, Secret, Classified, Unclassified, and the hierarchy may be in order

of TS > S > C > U. Within this hierarchical order each clearance level dominates itself and those below it.

In addition to the hierarchical clearance levels, categories can also be associated with subjects and objects to create compartmentalisation within the same clearance level. Categories are analogous to departments within an organisation. For example, a user cleared to access classified information from one department (marketing) cannot access the classified information of anther department (payroll) within the same organisation.

Although MAC can regulate flow of information within a mainframe environment where all processing and communication is centralised, it would not fare well in a distributed environment, where the client machines are powerful enough to store, manage, and communicate data through arbitrary channels. However, the idea of hierarchy, categorisation (or compartmentalisation), and information flow control is relevant to insider misuse problems and may be adapted for monitoring of information transfer. The analysis of MAC model has identified the need to consider the access rights of user(s), in relation to data involved, receiving data as a result of data transfer.

3.4.3.3 Privilege and Access Rights Management

Within IT environments with large number of users, it becomes very difficult to manage the access rights and permissions of each user individually. This is especially true within database environments, where a database file can contain a large number of records and

fields each requiring various security requirements. Thus scalability issues can arise when assigning permissions on individual basis. Due to this problem, permissions are associated with role(s), and then the users are assigned to roles to manage scalability. In addition it is desirable to be able to manage access permissions based on the hierarchical structure of the organisation. For the purpose of managing access permissions base upon the organisation structure, Ferraiolo and Khun (1992) proposed role based access control framework RBAC.

3.4.3.4 Role Based Access Control (RBAC)

RBAC framework manages access control policies based upon the functions a user is required to perform within the organization (Ferraiolo et.al. 1995). In order to control transactions based on the role of the user within the organisation, RBAC is mainly employed in database environments (Ramaswamy and Sandhu 1998) although Sun Solaris 8.0 and above supports RBAC. RBAC is policy neutral, it is a framework for managing access rights and permissions, and thus can accommodate both DAC and MAC policies.

RBAC regulates access permissions based on the user's role/responsibilities within the system, rather than the ownership of the objects. A role is defined as a set of job functions that the user is required to perform in order to satisfy his duties. For example, users may take on roles such as clerk, supervisor, manager, and director. Therefore the definition of roles closely resembles organisation hierarchy.

In the RBAC model, roles can have overlapping privileges or responsibilities, meaning users of different roles may need to perform common operations, and some of the operations may need to be performed by all users. It would be time consuming to repeatedly assign common privileges to all the roles. To solve this problem, RBAC allows the hierarchical structure in which a role can be inherited by another role (Moon et. al. 2004). Therefore, common privileges can be assigned to a common role, which can then be inherited by all the roles that exist in the system. If implemented properly, the structure of roles for RBAC can resemble organisational structure, and the hierarchy of roles can reflect the authority, responsibilities or duties of the users associated with each role. Separation of duties can be applied by specifying mutually exclusive operations and mutually exclusive roles, i.e. mutually exclusive operations cannot be assigned to the same role, and a user cannot be assigned to the roles that are mutually exclusive (Kuhn 1997). The concept of separation of duties is the use of processing procedures that require more than one person to complete a transaction or an operation, and it is particularly relevant to insider misuse. The analysis of RBAC has identified how violation of segregation of duties may be detected by providing the detection system with information regarding mutually exclusive operations that should not be performed by the same user. In addition the user's current operating environment is also an important issue related to the operation performed and the affects it can have upon the security of the data accessed (Park and Giordano, 2006). For example, regulating the acceptability of the user's current operation environment, when a system administrator performs important server backups

from home or performs system updates. Another example is data analyst accessing confidential database, from a wireless connection of an internet cafe.

The ability to enforce access controls that resemble organisation hierarchy and the capability to apply the principle of least privilege makes RBAC suitable for commercial and civilian organisations. However, access controls only regulate access permissions and the main problem with insider misuses is that the perpetrators misuse their existing privileges. Therefore, verification of user activity is still required after access has been granted. The idea of roles, responsibility hierarchy, and managing segregation of duties is relevant to insider misuse problem, and thus should be adopted for monitoring purposes.

Although application level access control complements operating system level access control and provides granularity, auditing is still required to monitor abuse of access rights by legitimate users, and verification is needed to ensure integrity of data after legitimate access.

It may be unethical for legitimate users to browse the database as it can result in breach of confidentiality/privacy. Thus, user access needs to be accounted. In addition, to ensure integrity user entries/modifications still need to be verified by a separate entity as fraud can result from unverified entries/verifications.

3.4.4 Ownership of information

In an organisation context, organisation owns the information stored within the file and not the system administrator or the creator of the file. Therefore, no single user should be solely responsible for deciding the deletion of the file, and changing security policy of a file. However, no entity that represents the organisation exists within the system environment to decide the security requirements of a file, and deletion of it. Therefore, deletion of files and policy changes should be regulated by users sharing the ownership of a file. For example, the system administrator of the file server and the business manager may share the ownership of a file, and both users must agree before the file can be successfully deleted, or permission changes can be made. If a third user is added to share the ownership of the file, the first two must agree etc. From then on all the users sharing ownership must agree for successful deletion of the file and file permission changes.

3.4.5 Limitations and challenges of access control

Despite the noted limitations of DAC, most commercially successfully operating systems (Windows, Mac, and UNIX) include DAC as default access control policy. MAC is still not widely employed as many commercial organisations do not have a strict reporting structure as the military. The fact that access control mechanism automates regulation of access can result in accessibility issues when exceptions need to be made and judgement need to be left to the business manager. In addition rigid access control policies like MAC and RBAC haven't been tried and tested across various organisations, and suitability can differ from one organisation to another. By definition of misfeasance as

previously noted before, the acceptability of user actions depends upon the organisation's culture and accepted code of conduct.

At the operating system level, access control only regulate whether a subject can read, write, or execute a file. It does not regulate whether the user can copy the content of the file from within an editor and paste the contents in another file. It does not regulate whether the file accessed may be saved to a removable media, or whether attaching the file with an email is acceptable.

Although access control can regulate access, a user having more privileges than necessary can abuse the privileges, which can result in sabotage or breach of confidentiality/privacy. Therefore the principle of least privilege needs to be applied when assigning permissions, i.e. on the need to know/edit basis.

DAC is associated with super-user, and privilege escalation problems. The owner of the file may pass on access permissions to any subject, the system administrator can override the permissions set by the owner.

The challenges of access control have also highlighted the need for segregation of system administration duties, and requirement for multi-person authorisation of certain activities. Monitoring sabotage by system administrators (by way of deleting critical data files) is difficult, especially when the system administrator has the ability to turn of the

monitoring. Without necessary preventions, monitoring system may only be able to provide prompt alerts and not able to exercise damage limitation.

Depending upon the attributes considered for decision-making, and the attributes that can be involved may be limited by the network, operating system or the application. It is also difficult to add new attributes to be involved in decision-making, and also difficult to add new logic for decision-making.

Access control does not define the application each user may utilise for accessing each file, and this can lead to clandestine operations if the application utilised does not provide sufficient audit data. The application utilised may also provide features for replication of content, and the propagation of confidential data can lead to weakened security of the data concerned. Therefore, the application each user utilises for accessing each file need to be audited, and monitored for activities that may lead to clandestine operations.

Depending upon the environment in which the user interacts, and the features provided by the environment, access control policy can be coarse or too fine-grained. One of the restrictions of access control is that once the access permissions are defined, human judgement is totally bypassed. When access control policy is too rigid, it can affect the functionality of the system and prevent the legitimate users from doing their tasks, because access control does not allow human intervention to accommodate exceptions.

Due to this restriction, sometimes controls have to be weakened than desired for accessibility.

However, in the case of misfeasors the users already have legitimate system and application level access. The misuse is only apparent when considered in the organisation context, code of conduct, or business rules. Auditing user activity does not affect functionality; yet provide avenues for acceptable usage analysis and detection of misfeasance.

Although, access control can minimise the risk of unauthorised users gaining access, and thus minimising risk, no single security technology can be a perfect solution to all the security problems. (Sandhu 1996) Each security technology, authentication, access control, encryption, auditing, and intrusion detection (audit analysis), address only part of the security problem and complements each other. In addition to the limitations of access control, protocol and software vulnerabilities also present opportunity for exploitation. When an application/executable/process containing the vulnerability operates with special privileges, the attacker can bypass access control mechanisms and operate under the identity of the exploited process. Existence of such vulnerabilities emphasise the inadequacy of maintaining system security with access control alone, which highlights the need for monitoring technologies. Vulnerabilities can result from bad software design, or from poor implementation.

3.4.6 Misfeasance vs. Access Control

Considering the list of potential misuses in the previous section, it is possible that appropriate access controls could be used to prevent some of them, but even these will not be sufficient for all contexts (consider, for instance, the case in which the misfeasor has legitimately been granted access). The basic problem with insider misuse is that the person concerned has legitimate access to IT resources of the target organisation. This means that he/she does not need to bypass the authentication mechanisms of the IT infrastructure (no stealing or illegal reproduction of passwords and other forms of authentication tokens). Thus, in an IT context, insider misuse is the act of abusing granted privileges to cause harm. In this context, it can also be observed that users that know more about a system are more likely to abuse their privileges than users who are less knowledgeable (Magklaras and Furnell 2002).

The problem with insider abuse is that, once a user is authenticated to use a system, what he does with the system or the objects he has access rights to is not analysed within the context in which the activity occurred and the system/data affected.

3.5 Auditing

Information security is based on the principles of: Confidentiality, Integrity, and Availability, usually known as the CIA triad. However, establishing accountability through auditing is also important for detection of misuse, and as forensic evidence.

From the preceding discussions, it can be noted that the problem of misfeasance cannot be prevented through access controls, because legitimate insiders already have authorised access, and thus the insiders already have the opportunity factor of the CMO model presented in Chapter 2, section 2.5. The notion of misfeasance is contextual, and in order to identify misfeasance, the user activity and the data or system affected needs to be analysed within relevant context defining acceptable use. Therefore, log data related to the user activity and the system/data needs to be collected for analysis of potential misuse. The following chapter reviews automated log analysers and intrusion detection systems in order to evaluate their relevance towards detection of misfeasance.

3.6 Conclusions

This chapter has reviewed existing controls and countermeasures that are available to tackle insider misuse. However, since these systems were not developed with insider misuse specifically in mind, the preventive mechanism and the logging present in today's commercial systems are not optimized for misuse detection. Access controls cannot prevent authorised users from misusing their granted privileges. For example, a user with administrator level privileges may not have the moral right to access confidential data on the system, but access controls present in today's systems cannot prevent such actions. In addition, if a rogue system administrator is also responsible for verification of security alerts; necessary response actions will be delayed. Access controls cannot prevent a user who has access to a confidential file from transferring the file over a network to someone who is not authorised for access. Access controls prevent unauthorised users from

modifying the data, however the integrity of data is not necessarily guaranteed, merely because an authorised user has entered/modified it. System level access controls cannot regulate the application utilised for accessing the data. The challenges faced by access controls have served to inform the design of the proposed system by highlighting the need to focus on a number of areas, which includes:

- monitoring access of data through arbitrary applications
- monitoring dissemination of data over networks
- enforcing multi-person verification of alerts
- automated verification of new/modified database records

As such, it is considered that some form of supervision system is required to monitor for misuse activity. Audit data can be utilised for determining possible flaws within the security system, and it is essential for detecting the misuse of privileges by authorised users (Sandhu and Samarati 1994). Even if prompt detection is not possible, the fact that accountability is ensured and evidence of activities is collected can discourage potential misfeasors. In addition more intelligence can be gained by collecting information regarding the user activities rather then preventing access. Prevention may sometimes force the perpetrators to find other routes to access, which may not be easily detected by monitoring systems.

Security monitoring tools work in similar manner to access control, except it does not deny access, when the parameters associated with the identified event/activity does not

satisfy defined policy, the event is alerted to the administrator. Such technologies are already available to some extent in the form of Intrusion Detection Systems (IDS) (Amoroso 1999), but as with many other mainstream security technologies, these are geared towards detecting attacks on the system rather than misuse of it by legitimate users. Nonetheless, some of the principles are transferable and these are consequently examined in the next chapter.

Chapter 4 IT Security Monitoring and Detection Tools

4.1 Introduction

Due to technical limitations and practical implementation issues related to access controls, analysis of event/activity logs is needed. However, it is not possible to manually analyse large volumes of logs, and thus Anderson (1980) proposed automated log analysers, and Denning (1987) introduced the main concepts of intrusion detection model. Automated log analysis software is commonly known as Intrusion Detection Systems (Amoroso 1999).

This chapter presents the architecture, the components, and analysis methods employed in intrusion detection systems. The main objective of the research is to be able to employ existing detection technologies for the detection of misfeasance. Thus, the analysis methods employed by current systems in detecting network and system exploits need to be studied, so that suitable approaches may be adopted for misfeasor monitoring. This chapter will identify the conditions facilitating successful detection of outsider attacks, so that they can be referenced to identify the requirements needed to facilitate successful detection of misfeasance. As part of the process, methods employed by outsider attacks will be discussed, although misfeasors do not need to employ similar methods.

4.2 Intrusion detection

Access control mechanism by nature is embedded within the environment in which it regulates access, while IDS can analyse audit data gathered from any environment. The

general architecture of an intrusion detection system is similar to access control mechanism, except that IDS alert a possible intrusion rather than regulate access.

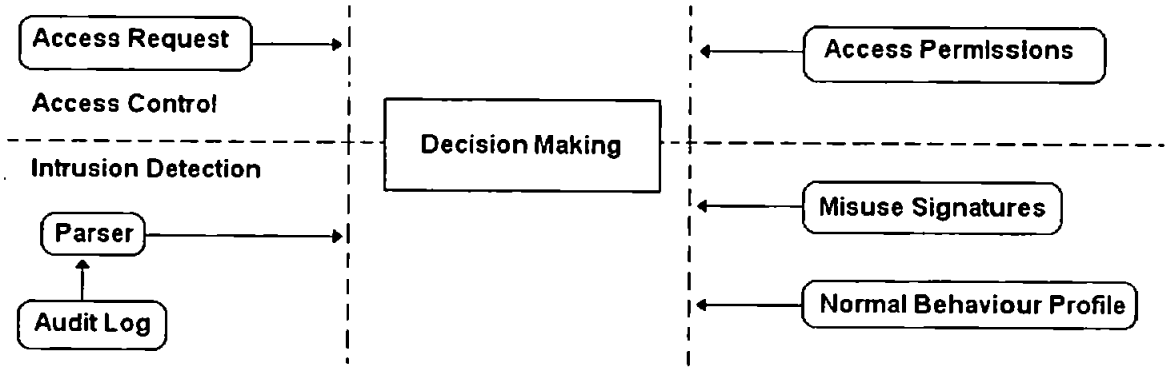


Figure 4.1 Access Control vs. Intrusion Detection

While access control mechanism uses explicit decision making procedure, and include only permission attributes for decision making, IDS can be configured to consider any attribute relevant to an attack during intrusion analysis. Within access control framework, the type of access (read, write, execute) is already associated with permission settings, IDS system needs to identify the event/activity (through parsing audit logs) before referring to misuse signatures or characteristics of norm for the given event/activity. While access control mechanism has embedded decision-making procedure that cannot be change, IDS needs to be provided with or added new rules or decision-making procedure for intrusion analysis. While access control mechanism's decision result in either access being granted or denied, IDS system can be configured to alert the administrator or the output can be directed to a response system so that an appropriate

response can be made rather than simply denying access (Papadaki 2004). While the access control's decision based upon the defined permissions is final, IDS provides room for human judgement and intervention.

4.2.1 Components of Intrusion Detection System

An intrusion detection system is composed of various components each performing a distinct function. Components of an intrusion detection system include (Furnell and Dowland 2000):

- *Sensors*: Sensors collect the data for intrusion analysis. Depending upon the data collected for analysis, sensors may be implemented at the network level, operating system level, and/or application level. Collected data should be directly related to the activity monitored and the data affected. For example, although network packets are directly related to communication activity at the network level, operating system logs are not directly related to user activity within the application environment.

- *Analysis Engine*: Inference engine performs reasoning based upon facts provided by the sensors and reference knowledgebase, and concludes the likelihood of misuse. Data provided by the sensors may need to be processed at various stages in order to provide the facts that can finally be used for the inference of misfeasance. For example, determining whether a user receiving

a file through a file transfer is an insider represents one stage of inference, and whether the receiver considered an insider has appropriate clearance to access the file received is another stage of inference. The facts derived from these various stages may be used to determine whether the activity is acceptable by the policy defined for the given context.

- *Alert Interface*: Alert interface provides the details of possible misuse activities to the IDS administrator. Regarding insider misuse, the interface must be able provide information in such a way that the administrator can understand the context in which the activity occurred, and why scrutiny is required. In addition, the alerts must be sent to the appropriate person who understands the contents of the data, and the acceptable usage policy within the context in which the activity was carried out. As previously identified, the segregation of duties is vital in order to detect privilege abuse. Within the IT environments the system administrators carryout the operations, and if the responsibility to verify the acceptability of the operations is also assigned to the same person who performed the operation, the opportunity to abuse trust arises. This is similar to trading and recording the trades in a business context, in which the two activities must be segregated to prevent fraud. In addition the system administrator may not have adequate knowledge regarding the sensitivity of the contents of each and every file managed by the system, whereas a business manager would have insight knowledge regarding the

sensitivity of the content of a set of files under his/her supervision. The business manager would also be aware of the users needing access to it and acceptable use of the information within the context of the activities each user may carryout. Therefore, mechanisms to distribute alerts to appropriate authority would be desirable.

4.3 Data Collection for Intrusion Analysis

What the detection system can detect depends upon the data analysed for detection. Different types of incidents can manifest themselves at varying levels within the system. Depending upon the nature of data collected for misuse/anomaly analysis, IDS are categorised into Network IDS, and Host IDS, although many of the current IDS are hybrid systems that analyse both the network and host data for indications of misuse.

- *Network-IDS*: analyse data collected at the network level and related to network communications in order to detect network reconnaissance and penetration attacks.

- *Host-IDS*: analyse data collected at the operating level and related to system level operations and activities in order to detect unusual application/user behaviour. Host based intrusion detection systems that analyse application level audit data are sub-categorised as Application-IDS.

- *Application-IDS*: analyse data provided by the application that is related to behaviour of the application/user within the context of the application in order to analyse the activity within the context of the application providing the data.

In order to understand how the existing technology may be employed for misfeasor detection, the application of the detection technologies for detecting network reconnaissance/penetration attacks and privilege escalation attacks need to be studied, so that the requirements of successful detection and suitability of each analysis technique for misfeasor activities may be identified.

Many of the currently available IDS are developed to detect attacks originating from outsiders. The point of entry for outsiders in to the organisation's network is through the Internet connection of the organisation. Accordingly, many intrusion detection systems place *Sensors* at this point of entry to collect data for analysis of network reconnaissance and penetration attacks (Porras and Valdes 1998).

4.3.1 Typical Network Structure of an Organisation

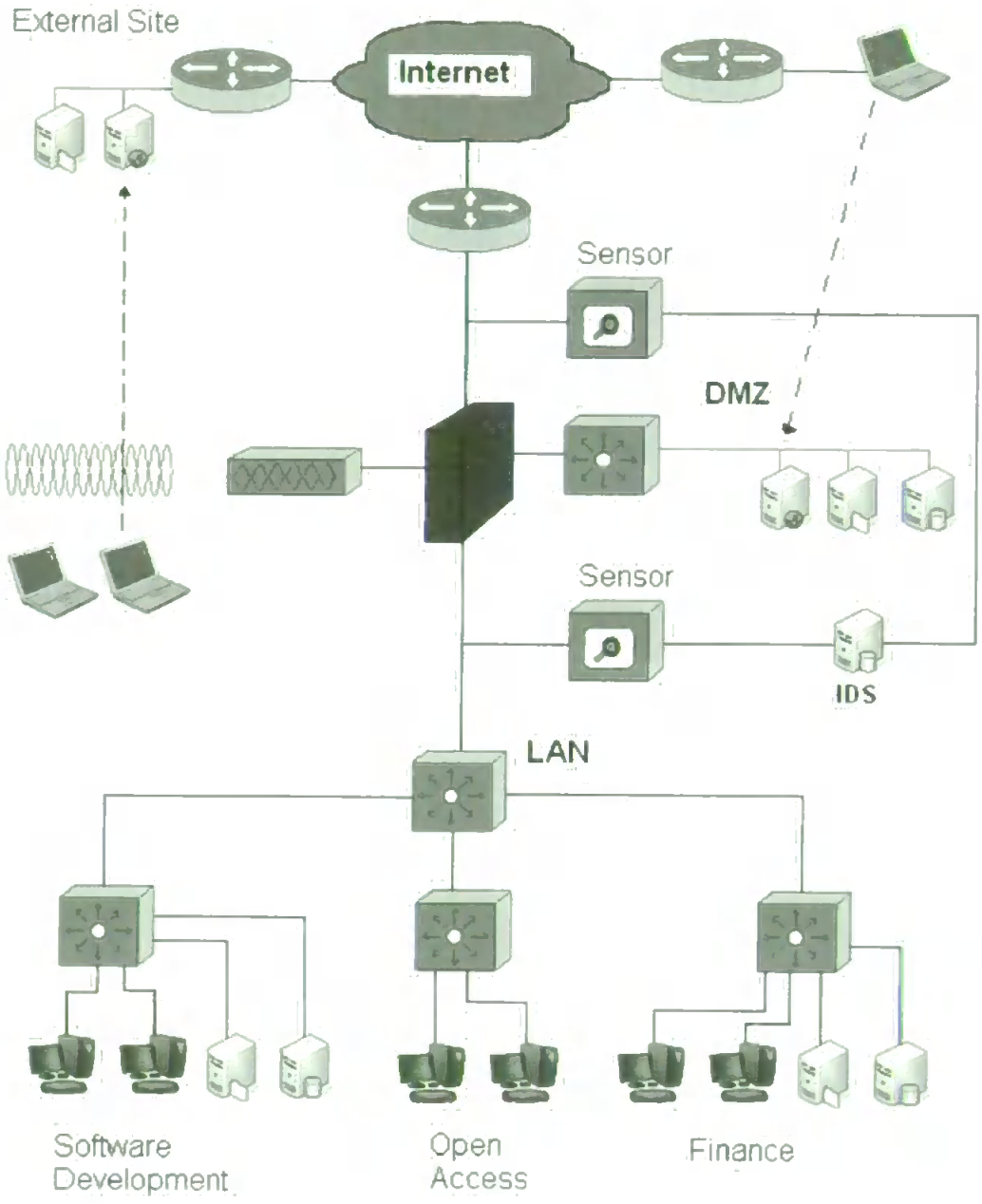


Figure 4.2 Illustration of an Organisation's IT Network

4.3.2 Network-Level Intrusion Detection Systems

NIDS performs detection at the network level, and the network traffic is monitored to look for attacks patterns. Network packets are the main source of data for monitoring. Network packets are captured by placing the network interface cards in promiscuous mode, while some network routers/switches include features for logging network packets.

Usually the system or data collection agent is located at the communication interface of the server, and analyse communication protocol between server and clients. This type of IDS 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. In order to monitor traffic of each host station, Kerschbaum et al (2000) suggested using embedded sensors, where sensors are embedded within the code of operating system that handles network packets.

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. If intrusion analysis is performed only on the data collected at the point of entry to an organisation, it can create an egg-shell affect because only perimeter security is ensured and systems within the network would still be vulnerable.

It is also important that anomalous access of isolated sub-nets is monitored. 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. 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.

The information analysed for identifying possible network attacks include, network packet headers, network packet content, end points involved in network connections, and bandwidth usage. The types of intrusions that can be detected by Network IDS include exploitation of network protocol vulnerabilities, and exploitation of server application vulnerabilities (Koziol 2003). Network IDS can detect denial-of-service attacks, or attempts to exploit server application vulnerabilities in order to gain further access to systems within the organisation's network.

Network protocols are a set of rules designed to accommodate successful communication among entities connected to network media (Duck et. al. 1996). Some of the protocols such as ICMP (RFC 792) are designed for network error detection and troubleshooting, and attackers may exploit the protocol to gain network information. Network reconnaissance involves exploitation of communication protocols, while network penetration involves exploitation of the network service/application.

Network packets are considered suspicious if they match some predefined signatures. Three main types of signatures are header condition signatures, port signatures and string (packet content) signatures. By checking header fields in the packets, the IDS would be able to monitor attacks on the network protocols. The packet headers can be checked to identify indications of impending attack, and the type of service the users is utilising. The packet content can be compared against signatures of known exploits in order to detect network penetration attacks.

Network level intrusion analysis can be broken down into:

- Network packet header analysis
- Network packet content analysis
- Usage statistic analysis

Network packet header analysis method is particularly relevant to misfeasance analysis, because some of the packet headers are legitimate and the analysis is selectively based

upon the contextual parameters of a connection such as source address, destination address, TCP/UDP service, and destination port. Therefore, the same method can be applied to misfeasor analysis, once the contextual parameters associated with each type of activity that may result in misfeasance is identified, and appropriate inference rules/procedures have been developed.

Due to the rigid structure of network packets (RFC 791, RFC 793, RFC 768), the structure of the data, and the meaning of each data field is already known to the analysis engine. The same network protocols are used globally. Therefore, NIDS decision making procedures can work for all organisations employing the same communication protocols. Within the organisation context the structure of the data may differ from one organisation to another, and the meaning of each data field may vary.

Inference rules of the NIDS understand the meaning of the values within each field of the packer headers at each layer of the communication model, and the acceptable values within each field of the header packets are limited. The knowledge of the values within the context of the activity, and the limitation of the variables makes network level intrusion detection possible. NIDS inference rules already understand the characteristics indicative of an intrusion.

Network firewalls also perform the kind of header analysis discussed above, and block specified traffic. The feature that differentiates NIDS from firewall is the ability to

monitor packet content to detect crafted code that may cause the server process to malfunction upon processing by the application layer and result in undesired consequences. Contents of network packets are compared against signatures of known attacks, to detect remote exploitation of application and/or system vulnerabilities.

Although, NIDS can monitor the content of network packets, detection may be avoided if the server process employs application level encryption. In addition, some NIDS are not very good at analysing fragmented packets, and if the code was sent within fragmented packets detection may be avoided (Ptacek and Newsham 1999).

In order to collect unencrypted (and fully assembled) data destined to the application, Almgren et al (2001) suggested application-integrated data collection for security monitoring. This approach uses a data collection module integrated within the application to collect information, providing the IDS with the raw data destined for interpretation by the application. This is different from analysing audit data provided by the application analysis, in the way that it provides the data yet to be processed by the application.

Despite employment of NIDS, some variations of TCP/IP protocol exploits continue to work because the NIDS cannot envisage how the destination operating system may handle the arriving packet, or how the receiving application may interpret the input (Northcutt & Novak 2002). In addition, network-level intrusion detection will not detect

system level attacks, attacks from directly attached terminals or attacks via dial-in modems directly connected to the target computer.

Misfeasors do not need to penetrate the network; they are employees of the organisation and already have legitimate access to systems within the organisation's network. With regards to misfeasor monitoring, Network IDS can help monitor excessive bandwidth usage by legitimate users, and also monitor anomalous connections among machines within the organisation's network. From a misfeasor monitoring perspective, network-level auditing can provide data relating to:

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

Insiders 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.

4.3.3 Host-Level Intrusion Detection Systems

From the discussion in the previous section, it can be noted that some of the input destined for the application may not be easily interpreted/detected by the NIDS, and thus requires detection at the operating system level of the host system for signs of intrusion.

Host IDS analyse data collected at the operating system level. Events or measures that may be indicative of an intrusion at the operating system level are resource (CUP, memory, disk) usage, modification of system files (Kim and Spafford 1993), and access to user files. In order to understand why these measures and events may be indicative of an intrusion, the interaction of hardware, operating system, and applications need to be examined. Operating system is responsible for managing the hardware resources such as CPU, memory, disk drives, network interface cards, and peripheral components; facilitating the user/server applications to read from and write to the resources it manages (Silberschatz et al. 2000). The concept of files and network communications are operating system's abstract presentation of a sequence of zeros and ones at the hardware component level. To facilitate applications to access the resources at abstraction level (files, terminal, etc), the operating system provides application interface through which the applications can read from or write to the hardware components, and request memory allocation and CPU processing cycles. Due to this interaction of application processes

(applications in execution) with the operating system, assumption is made that a process's normal behaviour can be characterised by the nature of its interaction with the operating system through API calls and the process's resource utilisation. The behaviour of a process changes when exploited, and this change in behaviour can be indicative of a buffer overflow exploit (Aleph One 1996).

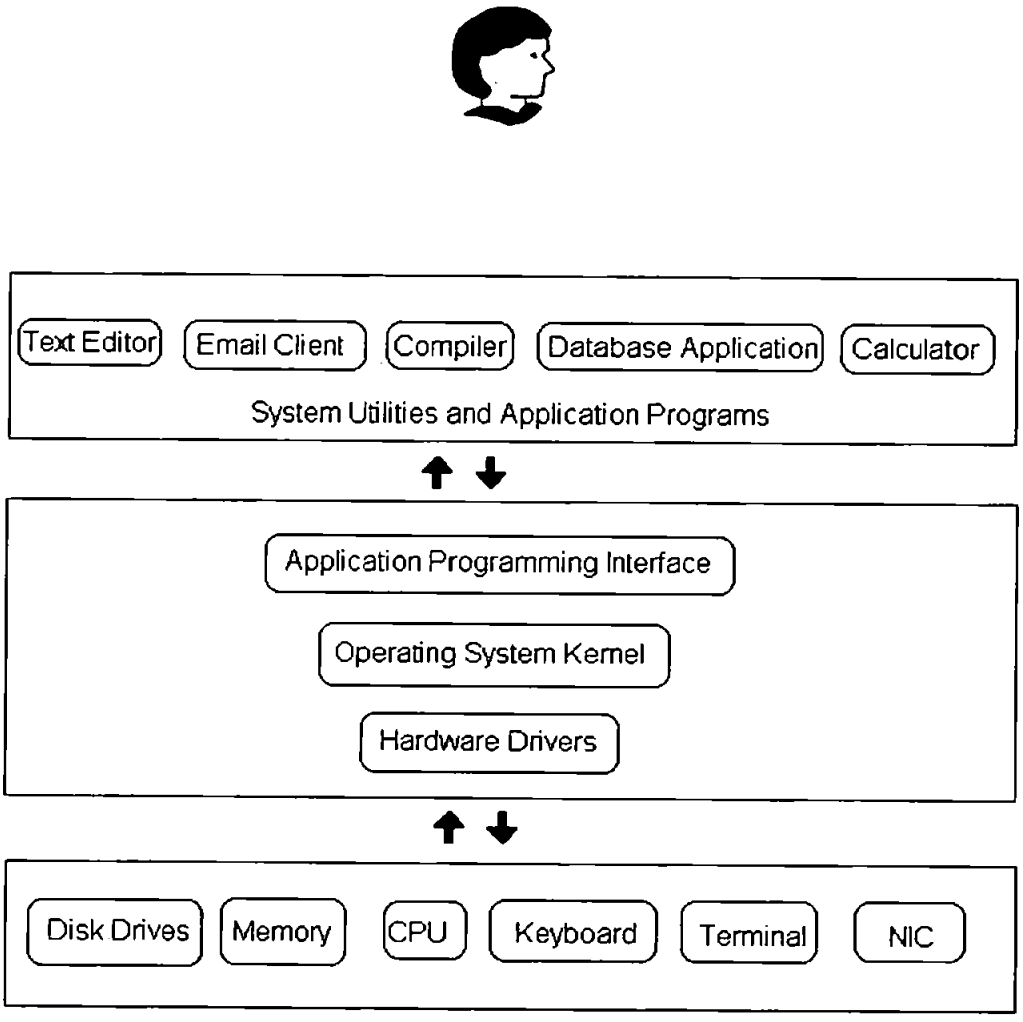


Figure 4.3 Interaction of Entities at Operating System Level

Buffer overflows and privilege escalation attacks are explained to demonstrate why resource utilisation and system calls (Hofmeyr et al 1998) at the operating system level can be used to characterise the normal behaviour of a process. Consequently it helps reason why misfeasance may not be detected by analysis of the same parameters.

Changes made to the system are most evident at this level and the changes would show up in configuration files or the registry. At this level IDS can monitor for the 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. Executions of unauthorised programs are also monitored at this level 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, in order to bypass application level access controls. At this level, atypical usage of I/O and atypical file access can be monitored. Atypical usage of I/O resources by systems may also indicate information leakage such as the backup server establishing connection to the Internet. 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 file containing proprietary content is used as an input to the encryption 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.

Insiders already have legitimate access to systems and misfeasance does not include privilege escalation attacks or system exploits, and thus may not be detected by analysing parameters that may be indicative of system exploits. Therefore, parameters relevant to misfeasance activities need to be identified.

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. However, suitable analysis/inference procedures for each type of misfeasor activities need to be developed.

4.3.3.1 Application-level monitoring

IDS monitor user interactions with the application such as request-response, access patterns, user input, application output, and user utilisation of application functions. For

the purpose of misfeasor monitoring, application level monitoring can provide information regarding the operation performed and the data affected; 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. 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. In order to enable analysis of user activity within the application environment, the knowledge of the application and the context of the activity need to be provided through inference rules.

Due to distributed nature of computer networks, Host-based IDS have evolved into agent-based IDS where the data collection takes place on the machine being monitored while the detection engine resides on a dedicated host (Balasubramaniyan et al 1998).

Host level intrusion analysis can be broken down into:

- Resource usage statistic analysis
- Command/Event pattern matching
- File Integrity monitoring

The fact that misuse is originating from authorised users and the abuse of granted access permissions makes insider misuse more difficult to detect than network protocol

exploitation attacks, and privilege escalation attacks, because the condition of misfeasance depends upon the combination of data affected, action performed, and the user(s) involved, and the security requirements of the data can vary depending upon the action performed and the user(s) involved. In addition, many of the misfeasor activities will only be apparent when monitored at the application level with the knowledge of contextual rules regarding the operation in the context of the application, because the application determines what the user can do with the accessed data/system. However, it will be impractical to monitor every user operation within each and every application. Therefore, there is a need to identify the applications and the functions within such applications that require misfeasor monitoring. In addition, relevant contextual rules that are required to identify misuse, and policies regarding function usage needs to be provided to the system.

4.4 Detection Strategies

Misfeasance is the abuse of access rights and privileges that have been granted legitimately, and misfeasance does not involve privilege escalation attacks that require modification of system/process behaviour. Therefore, any inherent ability to detect the abuse of legitimately granted access rights by the authorised users will be a pure coincidence rather than by design. However, the strategies and technologies employed within currently available IDS can be applied to misfeasor monitoring. Currently IDS systems employ two main strategies (Axelsson 2000):

- *Misuse Detection*: Compares current system/user activities against the database of misuse signature i.e. characteristics of misuse.

- *Anomaly Detection*: Compares current system/user behaviour against the historical/statistical profile of system/user behaviour i.e. characteristics of norm, and if current behaviour deviates from the profiled characteristics that define the norm then misuse activity is considered to be in the process.

Indeed the only difference is the monitoring system's perception of reference data provided for decision-making during analysis, one refers to behaviour considered as misuse, and the other refers to behaviour defined as normal. Therefore, although existing detection strategies can be applied to monitoring misfeasance, it is not possible to reach a satisfactory conclusion without the availability of relevant facts for reference. Thus, in order to achieve accurate detection, it is important that all the relevant data that is required to identify each type of misfeasor activity need to be provided to the misfeasance analysis engine.

Depending upon the detection system's perception of characteristics referenced for decision-making, detection strategies can be categorised into misuse detection, and anomaly based detection (Biermann et al. 2001).

- **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 (Kumar and Spafford 1994), which can then be matched to current activity using a rule-based approach. Attack signatures are usually sequence of events that correspond to an attack such as certain values within network packet headers. Attack signatures are then matched against current activity using rule based approach as shown in NIDS examples. If current user activity matches an attack signature, then the user is suspected to be misusing the system. One of the problems with misuse detection is how to write a signature that encompasses all variations of an attack, and not flag non-intrusive activity as intrusive. This approach is also very reliant upon the database of attack signatures. With this approach the detection system is only as good as the database of attack signatures. For insider misuse detection, the notion of misuse is contextual as the operation itself is legal. Therefore, it is difficult to generate misuse signatures of misfeasance, and detection requires inference rules that include relevant contextual parameters to suit monitoring of each activity. This approach can be applied to detect inappropriate configuration changes, file access through arbitrary applications, and verification of records.

- **Anomaly detection** (heuristics, i.e. trial and error)

  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 (Forrest et al. 1996). 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 (Denning 1987). 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. System/user behaviour may be profiled using statistical approaches, neural networks, or historical profiling of events (Marin et al. 2001). When statistical profiling is used, the measurements taken may be CPU usage, network usage, file access, time of access, and any variable that can be measured. From insider misuse detection perspective, anomaly-based detection seems to be suitable for certain type of insider activities. One of the advantages of monitoring insiders compared to outsiders is that the normal behaviour of insiders can be profiled and established. This advantage of being able to establish a normal user behaviour profile favours anomaly detection for some of the misfeasance activities, such as bandwidth usage, the number of records access. Generally, this approach can be used to monitor any misfeasance activity with measurable characteristics that can be detected through statistical analysis. Although the above descriptions make the concepts sound relatively straightforward, it must be appreciated that neither

technique can be considered 100% reliable, even in the context of traditional IDS. The consequence is that they can lead to false positives (where legitimate activity is believed to be intrusive) and false negatives (where genuine intrusive activities are misjudged as acceptable). The concept of applying the techniques for the detection of misfeasor activity makes the task more difficult, because it involves dealing with legitimate users who are performing legitimate operations which can only be deemed inappropriate when considered within contextual terms.

Anomaly detection is sometimes employed to detect intruders by comparing the current user's activities/behaviour against the characteristics of norm established for legitimate user(s). The reasoning here is that, if the activities/behaviour of the current user differs from the norms of legitimate users, or historical profile of a certain users, then the current user is deemed as an intruder.

4.5 Detection Technologies

There are several techniques to apply each detection strategy (Kumar 1995). Misuse-based detection can be employed using Expert Systems, State Transition Analysis, and Model-based detection, while anomaly-based detection can be employed using Statistical Profiling, Predictive pattern generation, and Neural Networks.

4.5.1 Expert Systems

This approach uses traditional expert system technology where the expert knowledge of the system security officer is coded as rules to identify attacks from the audit data (Lindqvist & Porras 1999). The rules are coded as ***if-then*** conditions. The conditions that constitute an attack are coded and if the audit data matches the conditions specified in the rule, then an attack is recognised. The weakness with this approach is that the system is only as good as the person who coded the rules.

4.5.2 State Transition Analysis

In this approach, the monitored system is represented as a state transition diagram (Ilgun et. al 1995). Here, an intrusion is considered as a sequence of actions performed by the perpetrator, which leads the system from a secure state to a compromised state. These systems usually list key actions that have to occur in order to complete an intrusion. This technique is applied in USTAT (Ilgun 1993).

4.5.3 Model-based detection

In this approach, known misuse scenarios are modelled as sequences of user/system behaviour (Uppuluri, and Sekar 2001). These behaviours are then modelled as events in the audit trail. If current sequence of system events matches with the modelled sequences of events that represent an attack, then the user is in the process on misusing the system.

For the purpose of misfeasor detection, a combination of expert systems and state transition analysis can potentially be used to detect dissemination of confidential data. Expert systems can be used to determine the facts, and the state transition analysis can be used to model the state of a system based on the facts provided by the expert system.

4.5.4 Statistical Profiling

With this approach, user behaviour is profiled using statistical measurements (Singh et al. 2001), (Barbara et al. 2001). A profile can include several types of measures. An intrusion detection measure is an aspect of user or process behaviour. A profile is a description of a user's/process' behaviour with respect to certain measures. This approach keeps statistic of each user or process for each intrusion detection measure (Lunt 1990). These stored statistics form the historical profile of a process or user. The profiles may be updated at regular intervals. In an adaptive system, the profiles are updated based on observed user behaviour. Therefore, the thresholds in the profiles will increase or decrease as the users' behaviour evolves over time. However, users may gradually train adaptive systems to accept intrusive behaviour as normal.

4.5.5 Predictive pattern generation

This approach is based on the hypothesis that the sequences of events are not random, but follow a distinct pattern (Teng et al. 1990). For example, E1 to E5 are security events, the prediction generated would be something like:

E1->E2-> E3=> (E4= 95%, E5=5%)

What the prediction states is that events E1 followed by E2 followed by E3, the probability of seeing E4 is 95% and E5 is 5%. These patterns generated forms the profile of the user. The deviation from normal behaviour is detected if the observed events match the left hand side of the pattern but the right hand side doesn't. A weakness of this approach is that unrecognised patterns of user behaviour may not be flagged as anomalous because they may not match the left hand side of the pattern prediction. In terms of misfeasance detection, analysing events within one environment may not be enough to conclude misfeasance, such as transfer of a confidential file through email, which involves events at the application, operating system, and network level. Therefore, correlation of multiple events at various levels within the system is required before the activity can be analysed in the context of acceptable usage.

4.5.6 Neural networks

This approach trains the neural net on a sequence of information units. The information units may be at a higher abstraction level than audit trails. For example if the user commands are information units, the input to the neural net would consist of current commands and last *n* commands. A number *n* for the past commands need to be defined so that the neural net can take last *n* commands into account when predicting the next command. If n is set too low, the net will perform poorly, if it is set too high, the net will be affected by irrelevant data. Once the neural net has been trained on the set of sequence

of commands that represent the user, the neural net creates the profile of the user. The incorrectly predicted next commands measure the deviance of user behaviour from his historical profile (Ryan et al 1998). Figure 4.4 illustrates the conceptual diagram of a neural net predicting next user commands.

The arrows directed at the input layer are the sequence of last n commands issued by the user. The output layer presents the next command predicted with respect to the sequence of last n command issued by the user. This approach is suitable when events within a single environment are monitored.

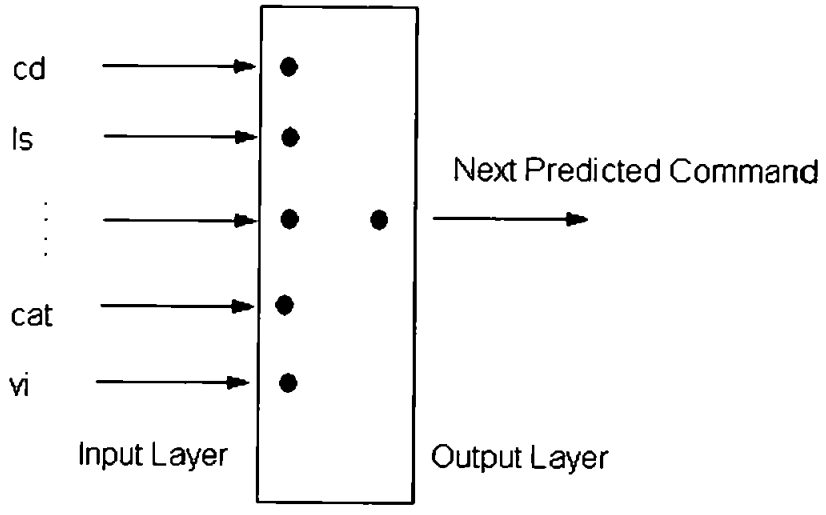


Figure 4.4 Neural Nets in Intrusion Detection

Although, statistical analysis and neural networks may be utilised for characterisation of behaviour/activity of the application/user, the accuracy of characterisation depends upon

the parameters included/considered or available to the application (Lee and Heinbuch 2001). The nature of insider attacks differs, and the relevant parameters also differ. The parameters considered need to be related to the context in which the activity is analysed. Again misfeasance is dependant upon the interpretability of acceptable usage policy, which entails the knowledge of acceptable use.

4.6 Relevant Systems

This section presents existing research that is relevant to misfeasor detection, and some of the technologies identified can aid in monitoring. The systems presented were selected on the basis of their ability to detect particular forms of misfeasance within the environment of their focus, and the relevance of their functions toward development of a comprehensive misfeasor monitoring system. The systems presented here are an indicative set of tools that focus on some of the misfeasor monitoring issues. The list does not represent an exhaustive list of available intrusion detection systems and monitoring tools.

4.6.1 DEMIDS

DEMIDS (DEtection of MIsuse in Database Systems) is proposed by (Chung et. al. 1999). It is a misuse detection system target at detecting misuse in relational database systems, especially abuse by legitimate insiders. DEMIDS uses audit logs to derive profiles of user behaviour in the database environment. The hypothesis is that a user will not normally access all records within a database schema. This approach assumes that

user access patterns in the databases form some *working scopes*. Working scopes are defined by the concept of *frequent itemsets*, which comprise sets of attributes of queries that are referenced together with some value. Frequent item sets, which describe the working scope of the users, are searched with the guidance of *distance measures*, to detect anomalous access. DEMIDS also considers the data structure and schema of a database through the use of distance measure. Distance measure is metric which measure the closeness of a set of query attributes with respect to working scopes.

Of all the IDSs reviewed, DEMIDS is the only one that is mainly focused on detecting insider abuse of privileges. With regards to misfeasor detection, DEMIDS can aid in detecting anomalous browsing of databases. Although DEMIDS can detect user activities deviating from their respective working scopes, it cannot effectively detect fraud, which requires counter verification of data entered, knowledge of organisation hierarchy and separation of duties.

4.6.2 DIDAFIT

DIDAFIT is the detection of intrusions in database through fingerprinting transactions. Low et al (2002) developed a process for fingerprinting SQL statements through the use of regular expressions, so that legitimate queries can be distinguished from malicious queries such as SQL injection.

With regard to misfeasor detection, DIDAFIT can aid in identifying the specific query a user has issued, or detect if the user query differs from the ones deemed acceptable. Although this solves part of the misfeasor problem, it does not include mechanisms for verifying the integrity of modified records, or validity of an access within the context of business process.

4.6.3 eXpert-BSM

eXpert-BSM is a host-based IDS, which analyses Sun Solaris audit trails in real time using forward-reasoning expert system (Lindqvist and Porras 2001). Source of data for intrusion analysis is gathered from Solaris Basic Security Module (BSM) (Sun Microsystems 1998) audit trails. eXpert-BSM utilises knowledge base detection method, which is build upon many years of research in intrusion detection. At the core of eXpert-BSM are an inference engine and a knowledge base built with the Production Based Expert System Toolset (P-BEST). P-BEST is an optimised forward-chaining rule-based system builder for real-time event analysis (Lindqvist and Porras 1999). P-BEST toolset includes a rule translator and a library of run-time routines. P-BEST models utilised in eXpert-BSM can detect intrusive behaviour that may involve complex ordering of events. This ability to comprehend complex event orderings allows the detection of a wide variation of an intrusive activity. At the time of the publication there are 123 P-BEST rules that allow the eXpert-BSM to recognise 46 general forms of misuse. eXpert-BSM excels at detecting security violations at the operating system level.

Although geared towards detecting users who exploit the system and violate security policy, some of the activities it can detect may be performed by legitimate users. For example, when some of the access permissions or privileges, which are not actually required for performing their daily responsibilities, may be granted to legitimate users as a result of erroneous configuration. The attack coverage of eXpert-BSM is categorised into the following broad areas of operating system level misuse:

- *Data Theft* – attempts to read files and devices by non-administrative users that violate security policy, such as accessing files stored in non public directories owned by other users, or read accesses that violates eXpert-BSM's surveillance policy. This category includes detection of opening network interface devices in promiscuous mode in attempt to sniff network traffic.

- *System/User Data Manipulation* – This category covers attempts by users to modify system files where security-relevant configuration data is stored. It also detects attempts to modify UNIX user environment file (eg. .bashrc, .login, .rhosts) and modification of files that violate eXpert-BSM's surveillance policy.

- *Privilege Subversion* – illegal attempts to gain higher privileges through illegal changing of user ID, or by exploiting privileged programs. Intrusion models in this category can detect three variations of buffer overflow attacks: exec argument buffer overflows, environment variable overflows, and data-segment overflows.

- *Account Probing and Guessing* – repeated attempts to gain access to the system via authentication services.

- *Suspicious Network Activity* – various attempts to probe or scan the host. Misuse of host's FTP services to distribute content to external sites. eXpert-BSM can also detect enabled TCP services on the host.

- *Asset Distress* – degradation of a system asset or impending failure of a system asset, such as file system or process table exhaustion, and core-dumps by root-owned services. This category also includes detection of service denials from remote agents and self echo flooding by host processes.

- *User-specifiable Surveillance* – eXpert-BSM allows the creation of site-specific policies, to detect certain activities, such as user defined command arguments that are considered suspicious. This also allows specification of site surveillance policy to monitor user accesses to data, and network ports that should not be accessed by external clients.

- *Other Security-relevant Events* – general security-relevant activities such as backward movement of the clock beyond what is normally performed by clock synchronisation protocols, indicating possible attempt to manipulate file or log state to evade detection. This category also includes creation of symbolic links in

world writable directories, process execution by reserved accounts that should not run applications (e.g, bin, sys), and attempts to modify audit configuration.

eXpert-BSM also provide detailed reports and recommendations of the detected intrusion activities. eXpert-BSM is capable of detecting an extensive range of intrusion scenarios.

4.6.4 Orchestria

Orchestria's Data Loss Prevention solution prevents sensitive data from leaving the organisation's IT network through applications such as e-mail, web-mail, and instant messengers (Orchestria 2007). It employs intelligent agents on the client machines and communication servers to monitor user communications. It allows communications to be monitored based on the meta-data associated, such as addresses or key words. It also monitors the content of file attachments and web uploads.

4.6.5 PortAuthority 5.0

Websense's PortAuthority tool prevents data leakage through the network and replication of files to removable media based on user defined policy (Websense 2007). It also includes digital rights management (DRM) mechanism to encrypt confidential information. It extracts and classifies information from various file formats. It allows the user to define leakage prevention policy, base on the meta data associated with the file, and other parameters such as geographical location of the end points of communication.

4.6.6 NetReplay

NetReplay by Chronicle Solutions captures and archives all user communications from organisation's systems (NetReplay 2007). A feature that stands out from other tools is that it provides CCTV like function, i.e. the security officer can view a particular communication as it would be seen by the user.

4.7 Conclusions

Each system presented focus upon a function that is different from another and for different purpose, and thus cannot be compared to one another. DEMIDS detect misfeasance within relational database environments by detecting users who access data that is beyond their working scope. DIDAFIT presents a method for generating SQL fingerprints so that anomalous queries may be identified. eXpert-BSM detects potential violations of acceptable usage policy at the host system level through providing detection system with the expert knowledge. Orchestria and PortAuthority 5.0 prevent data leakage through the network and removable media, while NetReplay captures and archives network communications from organisation's systems.

Each of systems presented here addresses a distinct issue related to misfeasor activity. However, none of the systems offer a comprehensive solution addressing various forms of misfeasor activity identified in Chapter 2.

From the study of existing IDS systems, it can be noted that network-based IDS are geared towards detecting network protocol exploits and string signatures within network packets. However, misfeasance does not involve exploitations of network protocols or network services. Host-based IDS are designed for detection of system level security violations, and system level anomalies. However, misfeasance activities do no exhibit detectable system level characteristics.

Despite their orientation towards detecting external attacks, the conditions required for detecting attacks can be noted and applied towards successful detection of misfeasance.

In NIDS, the reasoning logic and analysis procedures are tailored for interpreting the meaning of parameters within the structure of each network packet. The data necessary for analysis of an intrusion is available from capturing network traffic.

In HIDS, the reasoning logic and analysis procedures are tailored for interpreting the meaning of parameters available from system audit logs. The data for analysis is provided by the audit mechanism of the system, and sensors placed within the operating system.

The conditions accommodating successful analysis of misuse/intrusions are:

- Knowing the characteristics indicative of an intrusion/misuse
- The availability of data required for comparison against characteristics
- The reasoning logic tailored to detect each type of activity

Therefore, the activities that require misfeasor monitoring and the characteristics that may be indicative of misfeasance need to be identified, and determine the appropriate level of the system from where such data can be collected. In addition, a misfeasor monitoring systems requires tailored analysis procedures and reasoning logic to interpret parameters representing the characteristic of the context in which the operation was performed or the activity occurred. This also means that the parameters relevant to the context in which the activity is analysed must also be made available to the monitoring system. In the next chapter, various types of insider misuse will be categorised based upon the system level from where the data relevant for analysis can be collected.

Chapter 5 A Detection-Oriented Classification of Misfeasance

5.1 Introduction

From the discussions in the previous chapter it can be noted that one of the aspect of successful detection is the availability of relevant data for analysis. Further discussions will focus upon the issue of misfeasance, i.e. performing legitimate operations in inappropriate manner or for unapproved purpose, rather than IT misuse in a more general sense. Therefore, the term misfeasance will also be used instead of insider misuse in the following discussions.

This chapter considers how insider misuse incidents may be classified, giving particular attention to the points in the system at which different forms of misuse would be discernable. The discussion begins with a brief overview of existing approaches to classifying incidents and abusers, some of which already pay specific attention to the role of insiders. From this, the chapter proceeds to propose a detection-oriented approach to classification, and discusses examples of the different forms of insider-sourced incident that would be detectable at network, operating system, application and data levels within the system.

Existing taxonomies focus upon categorising attacks for:

- risk analysis
- describing the nature of attack
- describing the attacker, and type of attack
- incident reporting

- incident response

However, for the purpose of this research, a classification of attacks based upon the level of the system, at which each attack may manifest itself and thus detected is required.

5.2 A review of current intrusion taxonomies

In order to be able to focus on the misuses that may be committed by insiders of an organisation, it is important to understand the type and nature of all kinds of misuses. A number of previous investigations have therefore attempted to classify system attacks and abusers, in order to aid subsequent analysis. Some of these relevant works are summarised in the sections that follow, along with brief commentary in relation to their suitability for classifying incidents relating to insider misuse.

- **Cheswick-Bellovin Classification** divided attacks into seven categories drawn upon their work on firewalls (Cheswick and Bellovin 1994), and the categories are listed along with the nature of attacks belonging to each category.

  Although this approach gives an overview of the attacks, classifies the main categories of attacks and provides the terms to describe the nature of attacks, it is too general and does not give an insight to the characteristics of attacks, which is required for detection.

1. **Stealing passwords** – methods employed to obtain other users' passwords
2. **Social engineering** – deceiving unsuspecting individuals in to providing information that can aid in compromise of targeted systems.
3. **Bugs and Backdoors** – taking advantage of systems that have been poorly designed/implemented/configured, and/or replacing software with compromised versions such as Trojans
4. **Authentication failures** – means of defeating authentication mechanisms
5. **Protocol failures** – exploitation of protocols that have design flaws or are poorly implemented
6. **Information leakage** – utilisation of protocols such as ICMP, Traceroute, DNS, or diagnostic error messages to obtain information that is necessary for system administration and proper operation of the network, and abusing it
7. **Denial-of-service** – attempts to deny legitimate users form utilisation of systems and services

Table 5.1 Cheswick & Bellovin's seven categories of attacks

- **SRI Neumann-Parker Taxonomy** is based upon analysis of security incidents reported over 20 years (Neumann and Parker 1989). It classifies intrusions into nine categories, described by the nature of the misuse within the system environment and does not include external factors such as social engineering

mentioned by Cheswick and Bellovin. This kind of categorisation provides terms that can be used to describe the nature of system level attacks, and can be useful for incident reporting or for communication between detection system and response system. Table 5.2 summarises the overall scheme of the taxonomy.

NP1 External Misuse	Non-technical, physically separate intrusions
NP2 Hardware Misuse	Passive or active hardware security problems
NP3 Masquerading	Spoofs and identity changes
NP4 Subsequent Misuse	Setting up intrusions via plants, bugs
NP5 Control Bypass	Going around authorised protection/controls
NP6 Active Resource Misuse	Unauthorised modification of resources
NP7 Passive Resource Misuse	Unauthorised reading of resources
NP8 Misuse Via Inaction	Neglect or failure to protect a resource
NP9 Indirect Aid	Planning tools for misuse

Table 5.2 SRI Neumann-Parker Taxonomy

- **Lindqvist–Jonsson Taxonomy** is an extension of Neumann-Parker taxonomy, and categories NP5, NP6, and NP7 of Neumann-Parker taxonomy are further classified (Lindqvist and Jonsson 1997).

Extended NP5 Control Bypass	Password attacks, spoofing privileged programs, utilising weak authentication
Extended NP6 Active Resource Misuse	Exploitation of write permissions, resource exhaustion
Extended NP7 Passive Resource Misuse	Manual browsing, automated browsing

Table 5.3 Extension of SRI Neumann-Parker Taxonomy

Although (extended) NP6 and NP7 above do at least recognise the misuse issue, the rest represent the attack methods employed by outsiders, or insiders who utilises the same methods. In addition, the classification of attacks is based on the misuse techniques employed and the consequences of it, and it is not intended for monitoring purposes. However, some other works can also be identified that contain elements more specifically related to insider misuse.

- **Anderson's Taxonomy.** Anderson's early work (Anderson 1980) in this domain classifies system abusers into External Penetrators, Internal Penetrators, and Misfeasors, as previously discussed in Chapter 2.

  Although very useful at a broad conceptual level, the classification does not provide any significant assistance in terms of incident detection, with all insider misuse related incidents being grouped under the single 'misfeasor' heading.

- **Tuglular's Taxonomy** is the first comprehensive taxonomy of misfeasor incidents (Tuglular 2000), and the classification is based upon computer misuse incident in three dimensions: incidents, response and consequences. The Incidents dimension is further classified into target, subject, method, place, and time sub-dimensions. The Response dimension is divided into recognition, trace, indication, and suspect. The Consequences dimension includes disruption, loss, effect, violation, misuse type, misuse act, and result. The sub-dimensions branches into new branches of sub-dimensions and so on until it cannot be further classified. These dimension and sub-dimensions of the scheme are used to characterise each misuse incident. However, the entire taxonomy is orientated towards systematic data collection of insider incidents to provide evidence and incident response.

- **Magklaras-Furnell's Insider Threat Prediction Model** is human centric, and the authors argue that all actions that constitute IT misuse lead back to human factors. The fundamental aspect of this taxonomy is classifying people in three basic dimensions: system role, reason of misuse and system consequences (Magklaras and Furnell 2002). However, while this scheme is intended to assist threat prediction, which can be useful in determining the capability factor of a potential misfeasor. However, it is related more towards risk analysis than detection as it does not describe the specific parameters related to each type of misfeasor activity and the related characteristics.

The above mentioned taxonomies were not developed with the intention of detecting insider IT misuse, and the characteristics that may be indicative of insider misuse activities were not mentioned. A potential approach to this issue is considered in the remainder of the chapter.

5.3 A detection-oriented approach to classification

In determining a means to link classification to the method of detection, it is considered appropriate to classify insider misuses based on the level of the system at which they might be detected. The basis for this is that different types of misuses manifest themselves at varying levels of the system (e.g. some may be apparent at the network level, whereas others are most visible at higher levels, such as the operating system or application levels).

With this form of classification in mind, the concept can be illustrated using a variety of recognised insider misuse activities, and then considering the different levels at which they may be detected. An overall classification is presented in Table 5.4 and then examples of the incidents concerned are considered in the sub-sections that follow. The list of misuses presented is indicative rather than exhaustive. They are presented here on the basis of the potential consequences the activity may generate. These consider what could be monitored, and how this could be used to detect, control and restrict misuse-related behaviour. Lunt (1993) suggested the idea of identifying data that can be statistically measured for detecting possible intrusions. This classification has adopted Lunt's approach and has identified which data need to be evaluated (statistically or otherwise), or referenced to detect possible security violations (Phyo and Furnell 2004).

Misuse	Monitoring Level	Attribute(s) to monitor
Illegal content	Network	Packet content, MIME types
Excessive/anomalous usage	Network	Bandwidth usage
Resource exhaustion	Network	Bandwidth usage
Playing network/online games	Network	Bandwidth usage
Illegal software distribution	Network	Bandwidth usage

Misuse	Monitoring Level	Attribute(s) to monitor
Access to isolated subnets and machines	Network	IP address
Access from unauthorised machines	Network	IP address, MAC address
Access to prohibited online content	Network	URL
Use of web-based email	Network	URL
Recreational surfing	Network	URL
Instant messenger	Network	Service usage, protocol, port
Unauthorised network services	Network	Service usage, protocol, port
Unauthorised file sharing	Network, OS	Service/Bandwidth usage, File attributes
Unauthorised web hosting	Network	Service/Bandwidth usage
Resource exhaustion	OS	CPU, Memory, Disk usage
Storage of image and multimedia files	OS	File types, Number of files for each type
Anomalous command usage	OS, Application	Command utilisation
Anomalous application usage	OS	Application utilisation
Information disclosure	Application, OS, Network	File (read), Record access, Copy/Paste, Network Transfers

Misuse	Monitoring Level	Attribute(s) to monitor
Breach of privacy	Application, OS, Network	File (read), Record access
Data theft	Application, OS, Network	File (read), Copy/Paste, Network Transfers
Alteration of data files	OS	File (write), File Checksums, File Attributes
Alteration of system Files	OS	File (write), File Checksums, File Attributes
Hardware Installation	OS	File (create, write) configuration files
Software Installation	OS	File (execute) unauthorised program
Illegal program execution	OS	File (execute) unauthorised program
Sabotage	OS	File (write, delete)
Privileged Program Exploits	OS	API/System calls, File/ Memory access, I/O usage
Data Hiding	OS	Input files to programs
Encryption	OS	Input files to programs
Program Exploitation	Application	User Input/interaction
Alteration of Input	Application	User Input
Function Usage	Application	Queries, API Calls, Windows Messages
Anomalous Database Access	Application	User Queries, Range of query, Number of records accessed
Configuration Changes	Application, Data	Input flags

Misuse	Monitoring Level	Attribute(s) to monitor
Account creation	Application, Data	Feature usage, Data tables
Inconsistent Data (Fraud)	Data	ID Numbers, Date, Time, Strings, Numbers
Duplicate Entries (Fraud)	Data	Batch Number, Uniquely Identifiable Entities, etc.
Maximum Value	Data	Number of Employees, Bonuses, Extra-time work, gap between payments, etc.
Minimum Value	Data	Hourly pay rate, Work hours, etc.

Table 5.4 Detection Oriented Classification of Insider IT Misuse

5.3.1 Network-level misuses

Given that a great deal of misfeasor activity may relate to the use of network services, several type of misuse would be detectable by monitoring activity at the network traffic level. From a practical perspective, this has the advantage that there is no specific necessity to install monitoring / data collection agents on individual end-user systems. Examples of the misuses that could be identified are discussed below.

- *Access of prohibited content*: User access of prohibited content on the web may be monitored through logging and examination of web addresses accessed. Accessed web addresses may be checked against a database of websites containing inappropriate content, such as pornographic material. Another approach would be

to create a database of websites that the employees may access to perform their day-to-day tasks, then user accessed websites can be compared against the entries. It is not necessary to block the access to the websites that are not in the database; therefore access is not restricted, but monitored. The latter approach is more desirable if the organisations want to discourage recreational surfing.

- *Downloading inappropriate material*: File extensions of the users' network download can be monitored. For example, a user downloading files with image extensions may be downloading pornographic material. Other file extensions that should be monitored include ".mpeg", ".avi", ".mp3", and ".zip" files. Ideally, download rights should be limited to a few users as any type of downloaded material may introduce viruses into the organisation's networked systems. Downloading of large files can also consume valuable bandwidth and delay legitimate work.

- *Use of web-based email*: Many organisations disapprove the use of web-based email, because of the difficulties in monitoring usage. Employees may be circulating inappropriate material, or wasting work hours by sending personal emails through the use of web-based email, especially when the users' email accounts in the organisation are being monitored for usage. User accessed web addresses may be checked against a database of known web-based email sites.

- *Online shopping*: Users may be wasting valuable work hours by shopping online. User accessed websites may be checked against a database of online shopping websites.

- *Spamming*: Users sending more than normal amount of emails may be spamming using company computers. On the other hand, the user's email client might be infected with a worm that mails itself to everyone in the user's contact list. Whatever the case, a closer examination is required, when exceeding number of emails are sent from users.

- *Using chat programs*: Employee utilisation of chat programs such as IRC, ICQ, and instant messengers can affect the productivity of the users. Chat programs can also affect the security of the network as they introduce new services and those services may be exploited. In addition, such applications also provide new channels for unauthorised communications, which may be used to transfer confidential information. Network services utilised by users can be monitored to look out for utilisation of chat programs.

- *Video Conferencing*: Users may be video conferencing with friends or relatives using organisation's computing resources. Network service utilisation and bandwidth usage may be monitored to detect such abuse.

- *Playing network games*: Employees may be playing games on the organisation's local area network. Such activity may consume precious bandwidth. This kind of activity may be monitored through looking out for users with exceedingly high bandwidth consumption.

- *Running servers*: Users may be running personal web-servers from the company network. The motivation of such activity may be for financial gain or for mischievous purposes such as distribution of illegal software. Regardless of the motivation, unauthorised server applications introduce weak links to the organisation's IT security in addition to legal liability.

- *Peer-peer file sharing:* Users utilising file sharing programs may be downloading and sharing inappropriate materials with other internet users. Network service utilisation can be monitored to detect such abuses.

- *Access of isolated sub-networks*: Users accessing sub-networks that are not related to their domain may be suspicious. For example, a software developer establishing a direct connection to the payroll sub-net may have undesirable intentions such as modifying the payroll database to raise one's own wages. Cross network connections may be monitored to detect the access of isolated networks.

Having stated the possible monitoring opportunities for insider misuse at the network level, we should consider the following statement by Schultz (2002), "Insiders do not generally demonstrate the same attack signatures as external attackers". Indeed, 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 exploit the network-level 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.

5.3.2 System-Level misuses

In contrast to detecting network-level incidents, monitoring at the system level necessitates that monitoring activity be conducted upon individual host systems (i.e. some form of data collection agent would need to be present on the user system). If such monitoring is available, then the following list constitutes some examples of the types of incident that could be identified.

- *Storing inappropriate materials*: Users may be storing inappropriate materials on organisation's computers. For example, users may be storing MP3s, movies, illegal software, and pornographic materials. Users' home directories may be scanned to detect files with certain extensions, such as ".jpeg" to detect the content stored. For example a user having a large number of image/media files

may be storing inappropriate materials on the computer. User disk usage may also be monitored for excessive usage. Monitoring excessive disk usage may sometimes lead to the detection of illegal software being stored on company computers.

- *Use of data-hiding programs*: Users may be utilising data-hiding programs, such as steganographic software to hide inappropriate material. Such programs may also be used to disguise proprietary and confidential information before they can be sent out of the organisation. Programs that take file(s) as inputs and produce file(s) outputs should be examined to make sure they are not data-hiding programs, such as encryption and steganographic software.

- *Use of arbitrary programs*: Users may run arbitrary programs to access data. Sometimes when data is accessed through the use of arbitrary programs, application level access controls and auditing may be bypassed. Program executions may be checked against a database of authorised programs. This would require a database of authorised programs along with file check sums to guarantee integrity of the program being executed.

- *Modifying system configuration*: Users may be modifying system configuration files, which may affect the way the system and programs behave; such modifications are undesirable as the system security may be compromised as a

consequence. Monitoring access to vital system and application configuration files can lead to the detection of such abuse. This would require a database of critical configuration files and their check sums.

- *Adding unauthorised hardware*: Adding additional hardware, such as modems can affect the systems' security. For example, the user's communications through the modem will not be picked up by network intrusion detection systems, and the user may be sending confidential information out of the organisation. Addition of unauthorised hardware can be detected by monitoring system settings and configurations.

- *Output redirection*: Output from applications may be redirected to undesired destinations (files, networks, or machines). The output from certain applications may contain confidential information, which should only be sent to appropriate destinations. For example, backup process sending the backup data to a different machine than usual. In this example, the backup operator may be attempting to get proprietary information out of the company. Output destinations of applications processing important information can be profiled to detect anomalous output destinations.

- *Alteration of audit data*: Users may be altering audit and system accounting file to cover up traces of system abuse. Log files and audit trails should not be modified

even by the system administrator, because they contain evidential information regarding system abuses. Modification of log files can be monitored to detect users destroying evidential information.

- *Breach of Privacy*: Users may be accessing other users' files. The perpetrator may be someone with high system privileges or configuration errors may have made the file world readable. This type of incidents can be detected by monitoring users browsing files/directories own by others, and auditing file permission/ownership changes.

- *Batch Deletion*: Users or processes deleting a large number of files may sometimes represent sabotage of system or data. Therefore, users or processes deleting a batch of files can be monitored to detect possible sabotage of system and data. Managerial controls such as separation of duties should also be applied to deletion of files in work folders. For example, a user can be assigned the job of actually deleting the files, while users can mark files that should be deleted.

- *Installation of unauthorised software*: Every software program installed is a link in the security chain of the organisation. The newly installed program may introduce a new vulnerability through which the system may be exploited. The installed program may be a Trojan or viral infected software. In general software installation rights should be limited to a few users and programs should be

verified and authorised before installed on organisation's systems. In order to accommodate this, a list of executable directories needs to be established, and only the authorised programs stored in these directories may be executed. A database of authorised programs with associated check sums is also required. With this approach, users executing unauthorised programs or executing programs from arbitrary directories, such as home or temporary directories can be detected.

- *Copying software programs*: Users may copy customised software programs used in organisation's computers. For example, users can copy executable files, shared library files, and registry entries of a proprietary program for malicious purposes. Users accessing executable files in "Read" mode can be monitored to detect copying of executable programs.

- *Excessive Printing*: Users may be abusing organisation's printer facilities, for personal use and private work. Excessive usage of print services may be monitored to detect this type of abuse.

- *Input to programs*: Files containing confidential data may be passed to encryption/steganographic programs as input. Monitoring input to encryption/steganographic programs can detect users attempting to disguise information before sneaking it out of the organisation. This would require a list of

encryption/steganographic programs installed on the system. Then the file inputs to such programs can be checked if they are important confidential files.

It is clear from the preceding discussion that system-level monitoring gives the potential for a far wider range of misuse activities to be identified. However, some types of abuse will be distinguishable from normal activity only with the knowledge of application-level semantics, and consequently may not exhibit malicious behaviour at the system level. Therefore, to be fully comprehensive, some detection strategies will be necessary at the application and database levels.

5.3.3 Application and data-level misuses

Monitoring at this level must again be focused upon individual host systems, but now at a deeper level, collecting data from within individual applications that might attract misfeasor interest. The list below presents some examples of the general forms that misuse at this level might take.

- *Inappropriate inputs*: Users may type in inappropriate inputs into the applications. Inappropriate inputs can cause the application to crash, behave in an unexpected manner, or result in compromised integrity of the data. Entering a different type/format of data to the type/format expected by the application can result in the application misbehaving and disintegration of processed data. Entering a different range of data can result in fraud. User input could be monitored at the interface

level where the users interact with the application. In a client server environment, user inputs/request (server messages) may also be monitored at the server side.

- *Anomalous access of databases*: Anomalous access of databases can result in disclosure of confidential information and fraud. Insiders may misuse databases containing medical records, criminal records, customer data, personal records, and statistical information relating to businesses. Query requests by users may be monitored to detect anomalous access of databases.

- *Function usage*: Commercial off-the-shelf applications include many features some of which are not easily disabled, and usage of certain functions may result in disclosure of information or compromised data integrity. Monitoring user interaction within the application environment by auditing feature usage can help detect application level abuse.

For the purpose of monitoring misuse in database and transaction systems, it is conjectured 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. 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. Although, potential occurrence of fraud may be detected by monitoring for violation of separation of duties, the actual occurrence of fraud can only be detected by analysing the application data itself within the context of the transaction.

5.4 Conclusions

Existing intrusion taxonomies mainly describe characteristics of various attacks, and not developed specifically for monitoring insider misuse. Anderson was the first person to

classify different types of insiders who misuse the IT systems into, masqueraders, clandestine users, and misfeasors. However, these classifications only characterise the type of users and not the actual misuse or how they may be detected. Tuglular produced the first comprehensive taxonomy of insider misuses. However, Tuglular's taxonomy is primarily aimed for systematic data collection of insider incidents to provide evidence and incident response. This chapter presented a classification of insider IT misuses based upon the level(s) of the system at which each type of incident may be detected or monitored. Internet abuse may be detected at the Network level, while data theft, sabotage, resource exhaustion, process behaviour, and system modification may be detected at the OS level. Anomalous user interaction with the application, anomalous access of databases, and breach of separation of duties may be monitored at the application level and within the context of the organisation. Although, potential occurrence of fraud may be detected at the application level by monitoring violation of separation of duties, the actual occurrence of fraud may only be detected by analysis of the data.

Chapter 6 A Checklist for Identifying Misfeasor Monitoring Opportunities

6.1 Introduction

It may not be practical to comprehensively log all the user interaction and the information affected within each and every application environment. Therefore, applications that require misfeasor monitoring, and certain commands/features that may be subject to abuse within such applications need to be identified. In addition, monitoring every piece of information contained within IT systems may decrease detection efficiency and add undesired overhead. Some of the files, databases, tables, records, and data accessed may have a greater likelihood than the rest to be abused. With that purpose a checklist has been developed for identifying applications, operations and information that requires misfeasor monitoring. The checklist presented here has been developed as part of the research. It is intended as a guide to identify "what" needs to be monitored, and the issue of "how" it should be monitored is considered as part of the next chapter. This chapter discusses why some types of application are more likely to be misused than others and proposes a means by which such applications may be identified.

A data-centric approach is followed in order to develop the checklist. The data-centric approach is followed because without the presence of valuable data, it would not be financially viable to protect a system that holds no valuable data. Therefore, as a starting point applications and commands facilitated by the application in question is evaluated on the basis of whether the execution of the command affects confidentiality, and integrity of user-generated data. Availability of the data and services depends upon the proper functioning of the system. Therefore, the applications and commands facilitated by the

concerned application is evaluated on the basis of whether it affects the proper functionality of the system and the services it provides. In addition, the definition of misfeasance is contextual, and thus the entities relevant for contextual analysis also need to be identified.

6.2 Relevant Entities

The emphasis is placed upon misfeasor activities rather than general insider misuse, because the research focuses upon detecting abuse of legitimate privileges, while general insider misuse can also include insiders performing attacks usually employed by external attackers. Insiders employing the attack methods used by external attackers can be detected by traditional IDS (Bejtlich 2005).

The entities involved in misfeasor monitoring are the user(s), the application utilised, the command executed, and the data affected/involved. The relevant entities have been identified by considering the subjects (user, application, process), and objects (application, command, data) involved in the access and manipulation of data. The relationship between the entities is illustrated in Figure 6.1. The *user* interacts with *application* and executes *commands* facilitated by the *application*, while the *data* is accessed or modified through the application. The details of each entity can be used to determine possible occurrence of misuse. First of all, it is important to identify the type of applications/commands that are most likely to be misuse, and the nature of information that is subject to misuse.

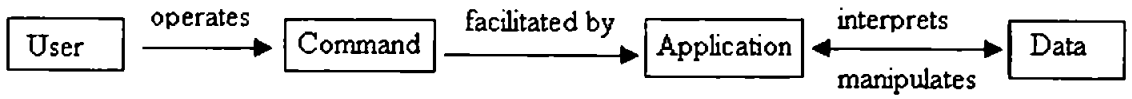


Figure 5.1 Relationship between the Entities Involved

6.2.1 Content

The content can be classified depending upon whether its creation took place within the organisation, or retrieved from external sources.

External content

Content created by external sources may be introduced to organisation's IT network via the Internet, or removable storage media. External content misfeasors may access include pornography (images, video), copyrighted music/video, illegal software, i.e. any content that the organisation can be held liable, or can bring disrepute to the organisation.

Internal content

Internal content may be subdivided into personal (belong to the user), and proprietary (belonging to the organisation). Internal content that belongs to the organisation may include product designs, blue prints, source code, contract details, customer data, marketing data, and supplier details.

6.2.2 Policy

In Chapter 2, it was mentioned that the notion of misuse implies the presence of rules defining the acceptable use of systems and information. Acceptable usage policy determines the security requirements and acceptable use of the information in order to maintain confidentiality, integrity, availability, and accountability. This is important because it is not possible to determine possible misfeasance without a properly defined (security or acceptable usage) policy concerning the data accessed. For example, the policy may indicate whether the data can be replicated, where the data may be replicated to etc. The information may also include the users who have access to the data, and the machines from which the file may be accessed. It may also include who should be informed of the changes made to the data. Details such as the applications that can be used to access the data may be included.

6.2.3 User Details

User credentials determine the relevance of data accessed (or received as a result of a file transfer) to that of user's responsibilities within the organisation. User details may include the department the user belongs to, the user's immediate superior, the role(s) the user has been assigned to, projects the user is a member of, user's email address, messenger addresses, telephone numbers, machines user may utilise, and the servers the user may access to. The identity of the user involved and associated details of each user involved are relevant to determining the possibility of misfeasance, because the

acceptable usage depends upon the user involved, the operation performed, the data/system affected, and the operational frame of reference in which the activity is interpreted (Neumann 1999).

6.2.4 Application/Command Capability

Features and commands facilitated by the application programs determine what the user can do within the IT system, and with the information accessed. Within the context of misfeasor monitoring, the users already have legitimate privileges and carryout abuse through the applications developed for completing their daily tasks. Therefore, the features facilitated by the application environment correspond to the capability factor of the CMO model mentioned in Chapter 2, section 2.5. Applications with distinct capabilities are considered along with possible misuse scenarios. One of the assumptions made is that undesired programs such as file sharing applications and games can be prevented from being installed. Therefore, only the applications that are generally used by many organisations in order to increase productivity and efficiency will be considered for misfeasor monitoring.

The main objective of IT security is to maintain the confidentiality, integrity, and availability of the systems and data while allowing storage, management, and manipulation of organisation's files and databases, and ensuring accountability for user activities. Proper operation of the system/applications relies upon the integrity of executable system files, and configuration files which the system and applications depend

upon. In addition organisation's valuable data is stored within files or databases. If no data exists, then there would not be any need for security in order to maintain confidentiality, integrity, or availability. Based upon this reasoning, the most harmful forms of misfeasance can be considered as data-centric. Thus, the first objective is to protect the files/databases stored on the systems from being destroyed, disturbed, doctored, or exposed. If an application provides access to data, a malicious insider may inappropriately modify, replicate, transfer, or doctor the data. Therefore, any application that can directly access the files, data services, and databases requires misfeasor monitoring. Based upon the reasoning that organisation's data is the main focus of security, the first step for determining what needs to be monitored can be based upon the capability of the application to access files/data.

- Applications that have access to files/databases.
- Applications that do not have access to files/databases.

6.3 Applications with No Direct Access to Files and Databases

Applications that do not have direct access to user-created files/databases, yet may affect the security of the system, need to be considered for monitoring. Examples would include:

- Communication related application/function
- Security related application/function

- Configuration related application/function
- User management application/function

Some of the applications with access to files and databases may also have these functions as built-in features. In addition, security, configuration, and user management functions may not be exclusive of each other.

6.3.1 Communication function

Networked applications with data communication/transfer features allow the user to convey information to other entities over the network, and the facility can be used for inappropriate dissemination of confidential data. Applications within this category include email, instant messengers, and VoIP software. The communication capability through IT systems makes it possible to convey confidential information without passing through physical security checks, and facilitates transfer of confidential data to entities unauthorised for access. Applications that allow the user to transfer files over the network or the Internet are likely tools to be used for dissemination of data, which can result in theft of proprietary information, breach of privacy, or undesired exposure. Therefore, applications that can accommodate data transfer to other machines through the network needs to be monitored for data transfer activities.

6.3.2 User/Registry management function

Adding users who should not have access to a certain system/database can result in sabotage, fraud, or information exposure, and can also result in accountability issues. If a single administrator had total control over the management of users then there is opportunity for the administrator to abuse the trust, such as creation of ghost accounts for use when dismissed by the organisation. Therefore, a form of verification is needed for addition and removal of users. For example, the database admin may have the privileges to add or remove users, and may subsequently abuse the privilege. Therefore, a separate entity or second person should be involved for verification whenever new records are added to important registry databases.

6.3.3 Configuration function

The system and applications need to be configured properly in order to be effectively usable. Depending upon the purpose of the system/application affected, it may affect the security or the service provided. If the system configurations were changed, the system and the services it provides may be inaccessible to authorised users. In some cases, poor configuration settings may result in undesired exposure of confidential/embarrassing information. Therefore, applications/functions that relate configuration should also be monitored for misuse. The importance of application with regards to productivity of the organisation, and the criticality of the settings adjusted regarding the functionality of the applications needs to be considered, and such information needs to be provided to the monitoring system.

6.3.3.1 Security-related functions

Security related applications could be used to harden or weaken the system security. When the security of the IT system is weakened the data stored in the system becomes vulnerable and opportunities for outsider attacks and further insider misuse may arise. Therefore, security related applications require misfeasor monitoring, even if the application does not have access to files and may not affect the data directly.

Security related applications can be further categorised into preventive mechanisms and monitoring mechanisms. If preventive mechanisms such as access control are weakened then the individuals who should not have access to the systems and data may access. If monitoring mechanisms are disabled or weakened then suspicious activities may go undetected. Therefore, applications that are related to system security and monitoring also need to be included for misfeasor monitoring.

Preventive and monitoring mechanisms may not be exclusive of each other, and some applications may offer a hybrid of both technologies.

6.4 Applications with Direct Access to Files and Databases

Applications that have access to the files and databases can be categorised depending upon the nature of access they have regarding the file. Applications such as file managers cannot open the file and access the content within it, but can copy, move or delete the files. Applications such as, Word processors, spreadsheet, and presentation software

accommodate the access to entire content of the file i.e. provide facility to view/manipulate the contents. However, applications such as database clients may be configured to accommodate access to only a small part of the file.

Applications with data access capability also can be checked whether they have functions related to security, configuration, user management, and communication. Applications that have data access and are capable of communicating the data accessed to another location are likely tools to be used for information theft.

6.4.1 File managers

File manager applications allow the user to access the files but not able to view or manipulate the contents directly. However, such applications may be used to replicate, move, or delete critical files. Therefore, applications with capability to manage files require misfeasor monitoring.

6.4.1.1 Propagation

Features such as Copy/Cut/Paste provide the capability to replicate and propagate content.

If the replicated file ends up in the wrong hands, it may result in information theft or undesired exposure without the organisation being aware of it. Therefore, this type of activity needs to be monitored.

Backup operations are one of the most important contingency plans for organisation's data. Therefore, these operations need to be monitored to keep track of where the backed up data is located, when it was made, and who is responsible for backing up the data. Note: segregation of duties should also be applied to performing backups and verifying the backed up data.

6.4.1.2 Move

When a file is moved, there may be consequences regarding the confidentiality of the information contained, and the accessibility of the data by regular users or applications that utilise the file. Therefore, this type of activity needs to be included for misfeasor monitoring.

6.4.1.3 Rename

When a file/database is renamed, accessibility issues may arise. Therefore, regular users of the file/databases should be informed, and configuration option of applications that utilise the file/database in order to provide services may need to be updated. In addition, if the monitoring of the file security is regulated on the basis of filename then renaming a file may be one of the steps in disguising information prior to theft.

6.4.1.4 Delete

It will have disastrous consequences if the only copy of a critical file/database is deleted, or if the backup of the critical file is deleted then a false sense of assurance may be created and contingency plans may be affected. When a file/database that is needed for day-to-day functioning of the business applications is deleted, productivity may be lost as a result. Therefore, file delete actions need to be monitored, and the appropriate authority needs to be alerted promptly.

6.4.2 Databases

Most of the valuable, confidential, and propriety information are stored, managed, and accessed through this type of application. Therefore, database applications require misfeasor monitoring. However, within database applications the value of underlying data determines whether the application requires monitoring. Thus, a number of questions concerning the importance of data to the organisation need to be asked while evaluating the value of data contained within the database system. Additionally, the databases themselves can be categorised into static and dymanic approaches, which alters the nature of the analysis that would be performed.

6.4.2.1 Static databases

Within this type of database, records are not added, updated, or deleted on regular basis. Usually historical records are stored in this type of databases for future analysis and

decision-making. If the answer to one of the questions listed is a "Yes", then the database requires misfeasor monitoring.

- *Is the database part of a decision making system?*

  If the information contained within this kind of database is exposed, competitors may understand the organisation's decision-making process and future strategies, which can eventually result in financial loss.

- *Is the database part of performance analysis system?*

  If the information contained within this kind of database is exposed, organisation's reputations and or share prices may be affected.

- *Does the database contain confidential records?*

  If information contained within this type of database is exposed, it may result in breach of privacy for individual or persons, for which the organisation may finally be held liable. Examples of this type of databases include health records, criminal records, financial records, student records, customer records, trade records, etc.

- *Is the database part of marketing system?*

  If information contained within this kind of database is revealed, the competitors may understand the organisation's marketing strategy and take advantage of such valuable knowledge.

- *Does the database contain valuable research data?*

  If information contained within this kind of database is exposed, the competitors may gain competitive edge over the organisation.

- *Does the database contain customer data?*

  If information contained within this type of database is exposed, competitors may have the opportunity to prise away organisations' existing and potential customers.

- *Does the database contain information that can lead to identity theft?*

  Today, identity theft has become a great issue (Kotadia 2003). For example, if the records contain the name, address, national insurance number, driver's licence information, date of birth, mother's maiden name etc. then such information may be used to create false identities, which can result in the organisation being held liable. In addition the information may be used to create false identities for attacks against the organisation, and for fraud.

6.4.2.2 Dynamic databases

Within this type of databases, records are added, updated, and sometimes deleted. Dynamic databases can again be categorised into transaction databases and non-transaction databases.

The checklist for static databases also applies to dynamic databases, and the subsequent points apply in addition.

- *Is the database part of delivery system?*

  If goods are delivered to the wrong address or at a later date, it may result in fraud, or delay production on the part of the customer, resulting in the organisation being held liable.

- *Is the database part of booking/reservation system?*

  If the information contained in this type of database is revealed, customer's privacy may be breached and the organisation may be held liable. For example, if the flight destination of the customer is revealed, the customer's privacy may be inadvertently breached.

- *Is the database part of pricing system?*

  If the wrong pricing information for goods and services are entered, it may result in fraud or financial/customer loss.

- *Is the database part of ordering system?*

  If the quantities ordered are inaccurate it may delay production. If the goods are ordered to be delivered to an address other than that of the organisation's then it may result in fraud. There may also be opportunities for employees to generate kickbacks from suppliers. Statistics may be needed to determine and compare the quantities ordered from each supplier for each type of product.

- *Is the database part of an inventory system?*

  If incorrect information is entered within this type of database fraud may result as a consequence, such as recording non-existent items, orders that did not arrive, or a different quantity than that actually arrived or ordered.

- *Is the database part of payroll system?*

  If information contained within this type of database is revealed, it may expose financial details of employees, which can result in low morale among employees or result in identity theft. There may also be opportunities to create ghost employees or dubious pay calculations in order to generate financial gain.

- *Is the database part of invoice system?*

  There may be opportunities to commit fraud within this type of databases, if the data entered is inaccurate, such as the billing address and the registered address of the credit/debit card.

- *Is the database part of a claimant system?*

  Within this type of databases, false claims may be made to generate financial gain.

- *Is the database part of a trading system?*

  Fraud may result within this type of databases if the contextual rules or business controls regarding each trade are not satisfied, or if the values entered are inaccurate.

- *Is the database part of a manufacturing system?*

  If the data/information entered within this type of databases is incorrect, it may delay production. For example, ordering less quantity of raw material than needed. Ordering more quantity then needed may also result in fraud. Another example is, ordering the product to be delivered at a later date than required.

- *Is the database part of an accounting system?*

  This type of databases usually contains sensitive financial information, which if revealed may affect share prices. If inaccurate data is entered within this type of databases, it may also result in fraud.

- *Is the database part of a transaction processing system?*

  There may be many steps involved in a single business transaction, and if the contextual rules or business controls for each step are not complied, fraud may result as a consequence.

❖ Non-transaction databases

Within this type of databases, modification of a record will not have direct effect upon another unrelated record. For example, modifying the marks of a student will not affect another student's record directly. For example, entering a type of allergy to a patient's record will not affect another patient's record directly. However, if the patient is prescribed with medicine, it will also affect the inventory database.

1. *Are there pre-requisite conditions to be satisfied?*

   For example, a student's record must exist in the registry, and the student must have registered with the institution from a certain date, and the student must have achieved required marks for specified modules before a certificate of graduation can be issued.

2. *Are there post-requisite conditions to be followed through?*

   For example, when an order is placed through the stock ordering system, the inventory system needs to be checked for the receipt of goods after a specified time (expected date for receipt of goods). Appropriate example in the banking scenario will be the Account Receivable, and the Cash Receipt Account.

3. *Does verification depend upon the values entered?*

   For example, if the student did not obtain pass marks, then the student cannot be graduated. In addition, the student cannot be still registered for the same course after a number of years.

   For example, the expected date for receipt of goods may be entered while placing orders. The date entered may be used to check for the arrival of goods, and if the date entered is unusual it may postpone verification.

❖ Transaction databases

Within transaction systems, modification of a record may have knock on effect upon another record. For example, when a fund transfer is carried out, the transferred amount will be deducted from the source account and added to the destination account.

1. *Are there conditions that must be satisfied for the operation to be legitimate?*

   For example, when a reimbursement operation for overdraft charges is carried out, the system needs to check the conditions qualifying a reimbursement, and also calculate that the amount reimbursed is correct.

2. *Are there contextual conditions that must be checked?*

   For example, when an order is placed through the stock ordering system, the inventory system needs to be checked for the receipt of goods after a specified time (expected date for receipt of goods). Appropriate example in the accounting scenario will be the Account Receivable, and the Cash Receipt Account to monitor cash flow.

3. *Is there a possibility of verification being deterred/delayed as a result of this operation?*

   For example, bank customers verify their transactions through bank statements. If the customer's address is changed without the knowledge of the customer, then the verification of transactions carried out on that customer's account would be deterred or delayed.

4. *Are there codes/ID/batch numbers that can be related to an operation?*

   For example, a range of batch numbers may be used only for claiming travel expenses, and a different range numbers may only be used for reimbursing equipment purchase etc.

5. *Are there matches to be verified?*

   For example, expense claims having the destination account number different from that of the claimant.

6. *Is the value entered related to calculation of loss/profit, bonus, charges, and interest rates?*

   For example, calculation of interest payment for each type of accounts can be complex, and an employee who understands how interest rates are calculated may defraud the organisation, the customer, or both. An appropriate example within the business environment will be the Accounts Payable fraud where the date for the payment to be made is significant in calculation of loss and profits, in addition it may also relate to prompt payment discount.

However, in order to detect possible occurrence of fraud, the detection system needs to be provided with the business/application specific contextual conditions, which can be provided through appropriate analysis procedure (inference rules) for a given operation.

If the activities deviate from the defined norm of contextual conditions, then there is the possibility of fraud in progress. Someone who has in-depth knowledge of the application or the business may provide such contextual conditions. Example of such personnel includes fraud auditors, and business managers. This highlights the fact that insider misuse is very much a management problem as much as a technical problem.

6.4.3 Access to entire files

While utilising applications that access the content of the file as a whole, the user may view and/or edit any part of the file. Such applications include word processors, e-book readers, spreadsheet programs, image viewer/editor, audio software, and video editors. Some research is being carried out to implement page level control within document management systems (Garg et al. 2004).

Within these types of applications, content propagation activities need to be monitored. Therefore, policies regarding data propagation also need to be defined before monitoring can detect misfeasance. Printed documents are beyond the scope of monitoring software, and monitoring shifts into physical realm. Due to lack of defined structure within such files, automated monitoring of data integrity is difficult, especially when users are authorised to modify.

The main concern with accessing valuable files is the dissemination of confidential information by legitimate users. Therefore, acceptable usage policy needs to indicate:

- whether replication of the file is acceptable
- whether replication of the file's contents is acceptable
- whether saving the file to a removable media is acceptable
- to which users the file/contents may be transferred
- the machines from which the file may be accessed
- the machines to which the file may be transferred
- who should be alerted if the acceptable usage policy is violated

6.5 The special case of browsers

Although editing capability is limited, Web browsers provide an interface that may be used for communication, file management, database access, viewing documents, viewing images, listening to audio files, and viewing video content.

6.5.1 External Content

1. What is the nature of content being accessed?
    - Text
    - Image
    - Video
    - Audio

Graphic, video, and audio downloaded from dubious web sites may affect the organisations reputation, and some may have legal liabilities. Therefore, web pages containing highly visual content needs to be monitored.

6.5.2 Internal Content

1. If the user is accessing files, then apply checklist provided for access to entire files.

2. If the user is accessing databases, then apply checklist provided for access to databases.

6.6 Software development tools

Software development tools should not be installed on operational systems, because the compiler can be used to execute malicious code. Machines installed with software development tools should be segregated from machines containing sensitive/critical data.

6.7 Conclusions

A methodical process for identifying applications/operations, and data that requires misfeasor monitoring has been presented in this chapter. For effective misfeasor monitoring, details regarding the security requirements of the files/databases, the role of the user within the organisation, and the capability of the applications are needed. Applications can first be categorised into those that have direct access to files and

databases, and those that do not have direct access. Whether an application has direct access to data or not, the functions related to security, configuration, communication, and user management need to be monitored for possible misuse. Within applications that have direct access to files and databases, user operations performed upon data need to be monitored to ensure that the security and acceptable use policy are satisfied. Download of video, audio, and images from the Internet should also be accounted and monitored. For fraud monitoring, additional verification process needs to be included in the transaction applications.

Merely having data indicative of misuse is not enough to detect insider misuse. Due to the fact that misfeasance is a contextual perception, appropriate inference/analysis procedures need to be developed for each activity considered, so that the data analysed may be interpreted within the context of the activity. The next chapter presents a generic conceptual architecture of a misfeasor monitoring system and appropriate analysis procedures that considers relevant data identified here for the analysis of misfeasance.

Chapter 7 Conceptual Architecture for a Misfeasor Monitoring System

7.1 Introduction

This chapter describes the architecture of a generic misfeasor monitoring system, and the processes involved in detecting potential misfeasance resulting from operations and activities identified. Concluding from the analyses made in the previous chapters, the activities that can result in misfeasance include:

1. Application Utilisation

   The application the user is actively interacting with can be used for performance and productivity monitoring.

2. Internet access

   a. URL/IP address

      The address of an Internet server can be used to determine the nature of content available to the user accessing it.

   b. Bandwidth

      i. Total bandwidth consumption attributed to each user

      ii. Bandwidth consumption attributed to each user by media type

         - Images
         - Video
         - Audio

iii. Bandwidth consumption attributed to each user by service type (e.g. Web, Email, IM, etc.)

3. Bypassing application level controls
    a. File access through anomalous application

4. Compromise of availability
    a. File deletion
    b. Configuration changes

5. Compromise of confidentiality
    a. Database access (read)
    b. Dissemination of confidential data
        i. Partial replication of contents
        ii. File replication
        iii. Transfer of file/data through network applications

6. Compromise of integrity
    a. Database access (Insert/updates, accounting)
    b. Registry Management (record addition, account creations)

Therefore, a misfeasor-monitoring tool should include features that can analyse such activities to determine possible misfeasance.

7.2 Overview of the conceptual monitoring tool

In order to accommodate analysis of various activities through a single analysis engine, it will be appropriate to identify and differentiate each user activity depending upon the suitable method of detection i.e. behaviour-based (statistical) or knowledge-based (inferential). Figure 7.1 illustrates the basic components of an intrusion detection system (Denning 1987), upon which the components needed to facilitate misfeasor monitoring will be added in Figure 7.2. Sensors provide the detection engine with the audit data related to the activity being monitored. Inferential component refers to knowledgebase during analysis of intrusion while the statistical component refers to the profiles of normal behaviour.

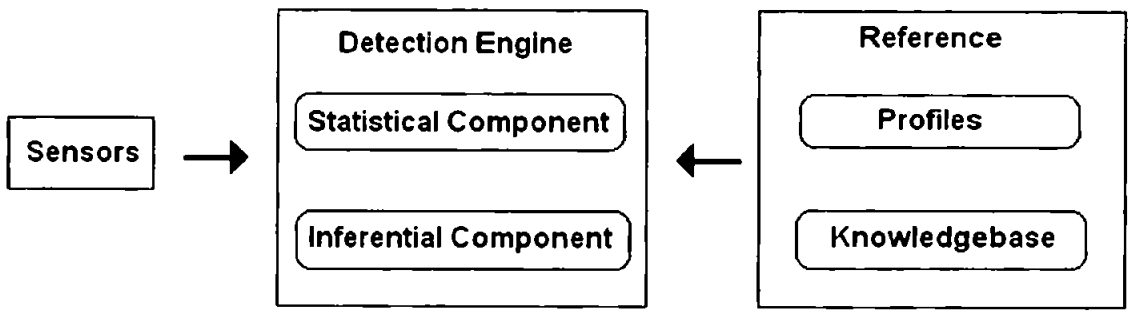


Figure 7.1 Basic Components of IDS

7.2.1 Statistical Analysis

Activities that may be monitored through statistical analysis for detection of misfeasance include:

- Amount of time spent interacting with each application environment
- Amount of network bandwidth utilised by each user
- The number of database queries issued by each user
- The number of records returned per query
- The number of records accessed per user within a time frame
- The percentage of a data table accessed per user within a time frame
- The percentage of a database accessed per user within a time frame

For the monitoring system to be effective the activity monitored and the acceptable usage policy needs to be closely related. For example, in order to detect the bandwidth usage abuse, the policy regarding bandwidth usage thresholds needs to be provided for automated decision-making. The values needed for reference can be manually defined by the assigned authority, or characterised through usage patterns of an individual, all system users, or users with similar responsibilities.

7.2.2 Inferential Analysis

Activities that may be monitored through inferential analysis for detection of misfeasance include.

- URL/IP address accessed
- File access through arbitrary applications
- Compromise of confidentiality through replication and dissemination of data
- Compromise of confidentiality through unethical access of database records
- Compromise of availability through deletion of critical data
- Compromise of availability through inappropriate configuration changes
- Compromise of integrity through inappropriate configuration changes
- Compromise of integrity trough inappropriate modification of data

The inference rules, the complexity of the rule and parameters needed by each rule differs for each user activity monitored. Therefore, the next step would be to define appropriate decision making rules, and identify the set of knowledge/facts/thresholds needed as reference to determine violation of acceptable usage policy associated with each activity. This will be discussed later in detail during analysis of decision making rules and reference data needed for each activity.

7.3 Components of the Misfeasor Monitoring System

Audit logs generated by various systems and applications may differ in format and the number of parameter logged. This varied format and parameters need to be parsed in order to generate a standardised format so that log data from heterogeneous systems can be processed. Appropriate analysis procedure and associated log data also need to be identified in order to accommodate monitoring of various user activities. In addition,

some of the log data from various level of the system or dispersed databases may need to be correlated or processed (such as add, subtract, compare etc.) in order to derive usable information that can be provided as facts. Segregation of duties and involvement of users who have intimate knowledge of the content and context is also an important issue related to misfeasor monitoring. The aforementioned factors were considered upon deciding the components needed by the misfeasor monitoring system, and determining which functions need to be decoupled. The additional components needed to facilitate misfeasor monitoring is combined with the basic components of an intrusion detection system to form the conceptual architecture of a misfeasor monitoring system as shown in Figure 7.2. The basic components of an intrusion detection system in Figure 7.2 are shaded.

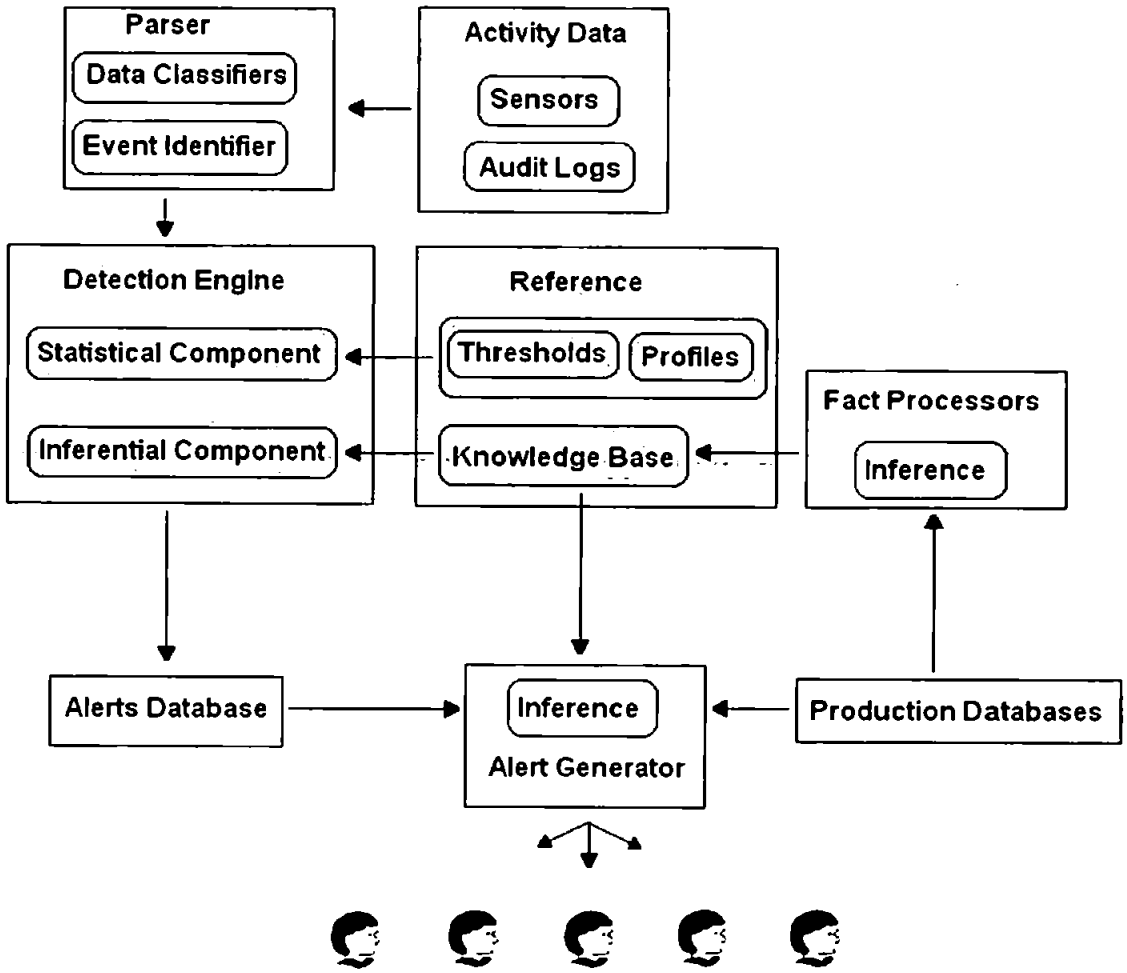


Figure 7.2 Overview of Misfeasor Monitoring System Components

7.3.1 Parser

Users may carryout numerous operations that may lead to misfeasance and a number of activities have been identified for monitoring. Each type of activity requires an appropriate analysis procedure. Therefore, the nature of the activity must be identified

prior to analysis of misfeasance. The *Parser* component performs pre-processing of audit data, classification of log data and identifies the nature of activity so that an appropriate analysis procedure can be chosen by the detection engine.

7.3.2 Fact Processors

In some cases such as dissemination of data, unethical access of records, and verification of database updates, the facts needed for inference of misfeasance, and determining the authority for verification can only be derived from live databases of the organisation. Therefore, *fact processors* are needed to infer the data from the organisation's databases and provide to the knowledgebase as facts. The *fact processors* also need to be provided with the inference rules regarding how each fact may be determined.

7.3.3 Alert Generator

One of the aspects presenting the opportunity for misuse is the lack of segregation of duties between the person responsible for the activity and the person verifying the activity. Therefore, if all alerts are sent to one person, segregation of duties will not be enforced. In addition the notion of misfeasance is contextual dependent, and a single person may not know all the contextual conditions relating to acceptable usage of the system/application/data involved.

Within the conceptual misfeasor monitoring system, the process of analysing events to determine whether an event should be alerted, and the process of determining the person

to be alerted are treated as two separate processes. The alert process is treated separately, so that rather than alerting all the events to the same administrator, alerts can be sent to the responsible authority depending upon the affected machine/application/file/record, and assigned authority. This approach allows segregation of duties to be enforced between the user performing the activity, and the user verifying the activity.

Alert generator determines the responsible authority by checking the knowledgebase directly, or derives the responsible authority from organisation's live databases according to the inference rules defined within the knowledgebase. Therefore, if the responsible authority cannot be determined directly from the facts within the knowledgebase, but can only be derived from organisation's live databases, inference rules relating to identification of responsible authority needs to be provided within the knowledgebase.

The architecture presented here differs from the one described in (Phyo and Furnell 2004b). The framework presented in (Phyo and Furnell 2004b) relies upon the ability to characterise user behaviour based upon the role of the user within the organisation and the user's daily responsibilities. However, the ability to characterise user behaviour based upon the role of the user within the organisation can differ from one organisation to another. Therefore, the framework presented in (Phyo and Furnell 2004) was abandoned and the architecture presented in this chapter was developed.

7.4 Application Utilisation Monitor

With the exception of dedicated terminals, many of the organisations computers include applications that can be used for recreation purposes. Users may not be able to carry out productive work while utilising such programs for long periods. Therefore, the amount of time each user spends actively interacting with certain application environments need to be checked to ensure that organisation's IT systems are used mainly for productivity purposes, and that the users are not wasting time surfing the web, chatting on messenger, writing personal emails, or playing computer games.

Monitoring of this can be accomplished through statistical analysis of the amount of time each user spent interacting within each application environment. The data for analysis can be collected through logging the focus window, i.e. the application the user is actively interacting within, and analysing the amount of time spent using the application either cumulative or per session.

Logging this information can also help determine the exact time at which the user was interacting with a particular application environment, and this can later be used in conjunction with other log data for misfeasor analysis. For example, when determining whether the user is/was utilising a communication application while accessing confidential information, or to determine the application that is/was in focus when screen capture operation was carried out.

The logged data may also be used for characterisation of each user's application usage patterns. For example, chronologically the user checks emails after logging in, then reply emails, access word documents, write emails, then utilise web browser etc.

Note: The values within the various log tables presented in this chapter were created artificially in order to illustrate the principle.

Log ID	Date	Time	Window Handle	Application
38	20/05/2007	10:40	3736472	Internet Explorer
39	20/05/2007	10:42	SystemIdle	
40	20/05/2007	10:55	3736472	Internet Explorer

Table 7.1 Active Window Log

From the collected data further characteristic and be derived, such as the amount of time each user spent utilising each application for a particular day.

Date	User Name	Application Name	Time Spent Interacting
20/05/2007	A.Phyo	Internet Explorer	3:27hr
20/05/2007	F.Steve	Internet Explorer	1:12hr
20/05/2007	A.Phyo	Visual Studio	2:50hr
20/05/2007	F.Steve	Visual Studio	0hr

Table 7.2 User Application Usage Log

The data can then be used to calculate the averages of all users within the organisation or users belonging to similar responsibilities. These characteristics can be used as reference in order to detect anomalies.

Date	Characteristic Category	Application	Time Spent
20/05/2007	Employee Average	Internet Explorer	2:17hr
20/05/2007	Role 9A Average	Internet Explorer	2:45hr

Table 7.3 Application Utilisation Characteristics

Alternatively the organisation policy may explicitly state thresholds of acceptable usage for reference.

Threshold Category	Application	Acceptable Time
Employee	Internet Explorer	1:30hr
Role 9A	Internet Explorer	3:00hr

Table 7.4 Application Utilisation Reference Thresholds

7.5 Internet Access

Users may spend great amount of time surfing the web, or downloading media unrelated to work. This not only affects their productivity, but also the productivity of the organisation as a whole, because it may delay access (how significant depends on the total bandwidth available to the organisation and the load the user is utilising) for users performing legitimate work, and it may also limit the ability to provide Internet-based services. In addition, the media downloaded by the user may be inappropriate such as

pornography, which can tarnish the organisation's reputation, or copyrighted material for which the organisation may be held liable.

User Name	Date	Time	URL/IP Visited
A.Phyo	20/5/2007	10:57	http://www.plymouth.ac.uk

Table 7.5 Internet Sites Accessed Log

The log of URL/IP visited by each user needs to be referenced against a list of addresses deemed acceptable and a list of addresses deemed unacceptable to facilitate automated detection.

Acceptable Addresses	Unacceptable Addresses
http://www.plymouth.ac.uk	http://www.pornography.com
http://www.network-research-group.org	http://www.warez.com

Table 7.6 Address Reference

In addition to the URL/IP visited, each user's bandwidth consumption can also be monitored for indications of possible misuse.

User Name	Date	Interval	Category	Mega Bytes
A.Phyo	20/05/2007	10-11am	Total Download	100
A.Phyo	20/05/2007	10-11am	Total Upload	50
A.Phyo	20/05/2007	10-11am	Image download	10
A.Phyo	20/05/2007	10-11am	Video download	20
A.Phyo	20/05/2007	10-11am	Audio download	70

Table 7.7 Bandwidth Usage Statistics

The monitoring approach for this is to either define thresholds for bandwidth usage limit, or to determine the normal bandwidth usage of a user for each role, then comparing it with the actual bandwidth usage of each user to identify those who may be abusing the bandwidth usage.

Norms of bandwidth usage for all employees of the organisation, those of a particular user group can also be derived from the collected data. This can be used as reference when detecting anomalous bandwidth usage.

Date	Interval	Application Name	Category	Mega Bytes
20/5/2007	10-11am	Internet Explorer	Average Download	50
20/5/2007	10-11am	Internet Explorer	Image Download	10
20/5/2007	10-11am	Internet Explorer	Video Download	0
20/5/2007	10-11am	Internet Explorer	Audio Download	25

Table 7.8 Bandwidth Usage Norms

7.6 Configuration Changes

Proper configurations need to be made in order for an application (service/security related) to function as desired. Therefore, configuration changes/updates need to be monitored to ensure that the application functions as expected. In order to enable this type of monitoring, the monitoring system needs to know the required configuration settings for each application within each and every monitored system [Table 7.11].

The data needed to log are:

Required Parameters	Example Values
Event ID	9
User Name	A.Phyo
Machine Name	PSQ_A304_WS1
Application Name	Firewall

Table 7.9 Configuration Changes Log

Event ID is needed to correlate the user input values associated with the event, and for chronological ordering of events.

User Name is needed for identification of the user responsible for the activity.

Machine Name and *Application Name* are needed to identify the configuration policy associated with the application for the given machine.

Event ID	Flag
9	Turn On
9	Limewire
9	MSN Messenger

Table 7.10 Flags Associated With Configuration Change Event

Event ID is needed to correlate the Flags associated with the each configuration change event.

Flag attribute is needed to describe the user input associated with each configuration change event.

Standard or normal configuration settings are required as a reference to determine whether the user actions are acceptable. The required application settings for one web/database server may be different to another, while settings for workstations in one network may be different to those in another network.

The data needed for reference are:

Machine Name and *Application Name* are needed to describe the associated configuration policy of each unique Machine-Application combination.

Required Flag attribute is needed to define the acceptable user inputs for the configuration changes made to each unique Machine-Application combination.

Machine Name	Application Name	Required Flag
PSQ_A304_WS1	Firewall	Turn On
PSQ_A304_WS1	Firewall	Don't Allow Exceptions

Table 7.11 Application Configuration Policy

The main purpose is to identify if the changes will affect functionality of the system/application. A system with weak controls will be vulnerable, however if the controls are too tight accessibility may be reduced, and if legitimate uses cannot access the services then it may reduce productivity.

Monitoring Process:

1. The "Machine Name" and "Application Name" values from the event log are compared against the "Machine Name", and "Application Name" values within the data table containing the required settings for each machine-application combination. This step is needed for identifying the acceptable configuration policy of each unique Machine-Application combination.

2. The monitoring system notes the "Flag" value of each and every matched entry of the required settings data table. This step identifies the flag values that the user input must correspond, in order to satisfy configuration policy.

3. The noted "Flag" values are compared against the list of "Flag" values associated with the event. This step determines whether current user input conforms to the configuration policy.

4. If all the required "Flag" values do not appear in the "Flag" values associated with the event, the event is logged into the "Configuration_Changes" data table of the alerts database. This step determines whether current user input includes all the flags required to satisfy the configuration policy. If all the required flags are not included, functionality, availability and security of the system may be affected.

5. If all the "Flag" values associated with the event do not appear in the list of required "Flag" values, the event is logged into the "Configuration_Changes" data table of the alerts database. This step determines whether current user input exceeds the requirements defined in the configuration policy. If new flags are added, the functionality, availability and security of the system may be affected.

Alert Process:

1. The "Machine Name" value is used to locate the record containing the details of the machine, so that the "System Administrator" can be identified and alerted. This step determines the appropriate person to be alerted, depending upon the machine affected, so that the alerts can be distributed correctly.

2. Perhaps file custodians of all the listed files located on the affected machine should also be alerted?

Information systems security officer (ISSO) should be assigned responsibility for defining security policies regarding the network and computer systems, and to ensure that the security is implemented as defined. The ISSO will only be responsible for defining system security, and network security, including communication and data transfer within internal sub-networks. While developing security policy for communication and data transfer, the ISSO may also have to arrange discussions with personnel from various

departments, so as to reach a balance between security and accessibility. One of the focuses of these discussions should be to define communication/transfer from one geographical location (building/room/department) to another, which the system administrator can later map to IP addresses.

System administrator's responsibility should only be to maintain functionality of the network, machines, and network services, and to configure the security applications as defined by the security policy.

7.7 Issues Regarding File Usage

One of the problems of access control is that it cannot determine what the user does with the file after the user has gained access to it especially with regards to dissemination of data. The user may create a copy of the confidential information and transfer/take it out of the organisation. The user may encrypt the data so that other legitimate and authorised users of the file cannot access it. The user may delete critical files in order to delay productivity or to cause sabotage. The user may move the confidential file to another location so that the information may be exposed. In order to monitor such misuse of privileges, the security requirements of the system/application/file need to be defined and the system needs to provide mechanism to create an acceptable usage policy for critical files, which will indicate what the user can do with the system/application/information the user has gained accessed to. However, it is not practical to include each and every file stored on organisation's computers for misfeasor monitoring, since the users may also store personal files on the workstations or their personal folders on the network drive.

Therefore, critical files that are considered as part of organisation's intellectual property needs to be listed for misfeasor monitoring, and the security policy for each file needs to be defined in order to enable misfeasor monitoring [Table 7.12].

A reference required by the misfeasor monitoring tool is the list of files that need to be included for misfeasor monitoring in the file inventory table of the knowledgebase and specify policy regarding dissemination of data.

Once a file is included in the list, the user will be asked to answer a number of questions with regards to the file's acceptable usage policy. The user listing the file for monitoring and answering security requirements may be someone responsible for watching over the file, "file guardian/custodian". System administrator is not responsible for listing files and answering the questions regarding what the user can do with the information accessed.

Required Parameters	Example Values
File ID	9
Machine Name	PSQ_A304_FS1
File Path	S:\Surveys\Misuse.doc
File Custodian	F.Steve
File Description	Misuse survey report
Application for Access	MS Word
Allow Save to Removable?	No
Allow File Replication?	No
Allow Partial Replication?	Yes

Table 7.12 File Inventory Table

For example, a head of Human Resource will be responsible for listing files containing employee details, and answering questions related to the security requirements of the file. The advantages of this approach is that the file guardian will have better knowledge than the system administrator regarding the sensitivity of the information contained in the file and which personnel need access to it, in addition it will also reduce workload for the system administrator. The novelty lies in getting "file custodians" involved in the security process. The anticipated drawback will be the need to provide training for users i.e. file guardians/custodians. File custodians should be those who understand the sensitivity of the content, and also (partly/wholly) responsible for deciding who needs access to the content and maintaining its confidentiality.

- *Which users need direct access to the file?*

  This policy is to be defined in the OS/Application level access control permissions.

- *Which users need indirect access to the file?*

  These are the users who do not have permission to read or write at the OS/Application level yet may need access to a copy of the file and thus can request from those who have direct access. Therefore, the monitoring system needs to be provided with information regarding which users may have access to replicated data [Table 7.20, and Table 7.21]. Within the misfeasor monitoring

system the policy can be defined within the FileReceivableRole, and FileReceivableUsers data tables in the knowledge database.

- *From which computers can the recipient retrieve the transferred file?*

  Some of the sub-networks within the organisation may be segregated to prevent computers outside the sub-net from directly accessing those within. However, some of the users from outside the sub-net may need access to some of the files hosted on the server within the subnet. As noted previously, a user who has direct access may transfer the file to the person needing access to a replica. However, the security of the data may be compromised if the machine the recipient uses to access the replica does not meet certain standards. Therefore, the machines that meet the standards required to host the replicated files from each server also need to be defined within the knowledgebase so that it can be referenced during inference [Table 7.23].

7.7.1 File Access

When a user accesses a file, the entities involved are the user, the application, and the file. Every entity involved is a link in the security chain, and thus the monitoring system needs to verify that all entities involved conform to the security (or acceptable usage) policy. If a users utilises an application different to the one normally used for accessing a classified file of certain type, the user may bypass application level controls, and may also evade auditing. Therefore, the monitoring system needs to determine not only

whether the user has the access rights to a file, but also whether the application utilised for accessing the file is acceptable. However, this type of regulation should only be applied to the files considered as the organisation's intellectual property. Therefore, an inventory of files considered as organisation's property is needed as reference [Table 7.14].

The data needed to log from the user activity are:

User Name is needed to identify the user responsible for the activity.

Machine Name is needed to determine where the activity was carried out from.

Application Name is needed to identify the application utilised for accessing the file, and to determine whether the application utilised is acceptable for accessing the file in question.

File Server and *File Path* are needed to identify the usage policy associated with the file involved.

Required Parameters	Example Values
User Name	A.Phyo
Machine Name	PSQ_A304_WS1
Application Name	Windows Explorer
File Server	PSQ_A304_FS1
File Path	S:\Security\Survery04.doc

Table 7.13 File Access Log Table

The data needed for reference are:

Required Parameters	Example Values
File ID	9
Machine Name	PSQ_A304_FS1
File Path	S:\Surveys\Misuse.doc
File Custodian	F.Steve
File Description	Misuse survey report
Application for Access	MS Word
Allow Save to Removable?	No
Allow File Replication?	No
Allow Partial Replication?	Yes

Table 7.14 File Inventory Table

File ID is needed to represent the unique combination of *Machine Name* and *File Path* of inventoried files.

File Custodian is needed to enforce multi-person verification of alerts for the file concerned.

File Description is needed to describe the nature of the content within the file when alerting.

Application for Access is needed to define the application that can be used to access the file concerned.

Allow Save to Removable is needed to define whether it is acceptable to save the file on to a removable media.

Allow File Replication is needed to define whether replication of the file is acceptable.

Allow Partial Replication is needed to define whether partial replication of the file's contents is acceptable.

Monitoring Procedure:

1. The "File Server" and "File Path" or "File ID" from the user activity log are compared against the entries within the file inventory table to determine whether the accessed file requires monitoring. This step is needed to identify the security policy associated with the file involved.

2. If a match is found in the file inventory, the application utilised to access the file is compared against the application defined as normal. This step is needed to determine whether the application utilised for accessing the file is acceptable according to the file's security policy. If the application utilised differs from the one defined, application level controls may be bypassed and application level auditing may be avoided.

3. If the application utilised for access is different from the one defined as normal, the event is logged in the database of alerts. This step logs the event for alerting, if the activity violates the acceptable usage policy associated with the file.

Alert Procedure:

1. The "File Server" and "File Path" from the alert log are used to locate the record containing the details of the file, so that the "File Custodian" can be identified and alerted. This step is needed to identify the person responsible for verifying the alerts associated with the file affected, so that the alerts can be sent to the appropriate person for multi-person verification purpose.

2. The "File Server" value is used to locate the record containing the details of the machine, so that the "System Administrator" can be identified and alerted. This step is needed to determine the administrator of the file server on which the affected file is hosted, so the appropriate administrator can be alerted.

7.7.2 File Deletion

When a user deletes a file, the monitoring system needs to determine whether the file being deleted is a personal file, or an organisation file. Therefore, the files regarded, as organisation's intellectual property need to be listed in a file inventory [Table 7.14]. In addition, the monitoring system also needs to know who is responsible for watching over

the security of the file so that the appropriate authority can be alerted promptly [Table 7.14].

The data needed to log are:

User Name is needed to identify the user responsible for the activity.

Machine Name is needed to determine where the activity was carried out from.

Application Name is needed to identify the application utilised for accessing the file.

File Server and *File Path* are needed to identify the usage policy associated with the file involved.

Required Parameters	Example Values
User Name	A.Phyo
Machine Utilised	PSQ_A304_WS1
Application Utilised	Windows Explorer
File Server	PSQ_A304_FS1
File Path	S:\Surveys\Misuse.doc

Table 7.15 File Deletion Log Example

Monitoring process:

1. The "File Server" and "File Path" or "File ID" from the event log are compared against the files listed within the file inventory table to determine whether the

accessed file requires monitoring. This step is needed to determine whether the file involved is intellectual property of the organisation.

2. If the file is listed, then the event is logged into the alerts database. This step logs the event if the user deleted the file considered as intellectual property of the organisation, because deletion of information regarded as intellectual property of the organisation can result in sabotage.

Alert Process:

1. The "File Server" and "File Path" or "File ID" from the alert log used to locate the record containing the details of the file, so that the "File Custodian" can be identified and alerted. This step determines the person responsible for verifying alerts associated with the file involved, so that multi-person verification can be enforced.

2. The "File Server" value is used to locate the record containing the details of the machine, so that the "System Administrator" can be identified and alerted. This step is needed to determine the administrator of the file server on which the affected file is hosted, so the appropriate administrator can be alerted.

7.7.3 File Replication

When a user replicates a file containing sensitive information, the replicated file also needs to be applied identical security policy in order to maintain confidentiality of the contents. However, in order to enforce the security policy, the replicated file also must reside on one of the computers monitored by the monitoring system. If the file containing sensitive information is replicated to a system that is not monitored, or on to a removable disk, the security of the information is compromised. Therefore, the monitoring system needs to know whether it is acceptable to replicate a file listed for monitoring, and whether replicating the file to a removable disk is acceptable. Policy regarding replication of content and replication of the file needs to be provided in the file inventory table for reference [Table 7.14].

The data needed to log are:

User Name is needed to identify the user responsible for the activity.

Machine Name is needed to determine where the activity was carried out from.

Application Name is needed to identify the application utilised for accessing the file.

File Server and *File Path* are needed to identify the usage policy associated with the file involved.

Destination Machine and *Destination File Path* are needed to identify the location in which the replicated file is saved.

Required Parameters	Example Values
User Name	A.Phyo
Machine Utilised	PSQ_A304_WS1
Application Utilised	Windows Explorer
File Server	PSQ_A304_FS1
File Path	S:\Surveys\Misuse.mdb
Destination Machine	PSQ_A304_WS1
Destination File Path	D:\Documents\Misuse.mdb

Table 7.16 File Replication Log

Monitoring Process:

1. The "File Server" and "File Path" from the event log are compared against the listed files to determine whether the accessed file requires monitoring. This step is needed to determine the security policy associated with the file involved.

2. If the file is listed, then the system checks whether replication of the file to a removable disk is acceptable. If it is, no further analysis is made, and no alerts generated. This step determines whether the security policy allows the file to be

saved to removable media. If saving the file to removable media is acceptable then it is assumed that replicating the file to any machine (insider or outside the organisation) is acceptable. Therefore, no further analysis needs to be made.

3. If the replication of file to removable disk is not acceptable, the system checks whether file replication is acceptable. If it is, the system logs the event to the File_Replications table of the alert database, with the alert value set to false. This is logged in order to keep track of the number of copies made, where the copies are saved, and to enforce security policy on the replicated files. In addition, in case of the original file being deleted, replicated files can be used for data recovery.

4. If the replication of the file is not acceptable, the system logs the event to the File_Replications table of the alert database with the alert value set to true. This step logs the event and associated data to a relevant log table, if replication of the file is not acceptable.

Alert Process:

1. The "File Server" and "File Path" or "File ID" from the alert log used to locate the record containing the details of the file, so that the "File Custodian" can be identified and alerted. This step determines the person responsible for verifying

alerts associated with the file involved, in order to enforce multi-person verification.

2. The "File Server" value is used to locate the record containing the details of the machine, so that the "System Administrator" can be identified and alerted. This step is needed to determine the administrator of the file server on which the affected file is hosted, so the appropriate administrator can be alerted.

7.7.4 Partial replication of file content

In some scenarios, a misfeasor may not copy an entire file through a file management application, but copy majority of the information through applications that have direct access to the content of the file, which can still compromise the confidentiality of the information. Therefore, the monitoring system needs to know whether partial replication of file contents is acceptable, and where the contents have been copied.

The data needed to log are:

User Name is needed to identify the user responsible for the activity.

Machine Name is needed to determine where the activity was carried out from.

Application Name is needed to identify the application utilised for accessing the file.

File Server and *File Path* are needed to identify the usage policy associated with the file involved.

Destination Machine and *Destination File Path* are needed to identify the location of the file in which the replicated content is saved.

Required Parameters	Example Values
User Name	A.Phyo
Machine Utilised	PSQ_A304_WS1
Application Utilised	MS Access
File Server	PSQ_A304_FS1
File Path	S:\Surveys\Misuse.mdb
Destination Machine	PSQ_A304_WS1
Destination File Path	D:\Documents\Misuse.mdb

Table 7.17 Content Replication Log

Monitoring Process:

1. The "File Server" and "File Path" from the event log are compared against the files listed in the file inventory to determine whether the accessed file requires monitoring. This step is needed to determine the security policy associated with the file involved.

2. If the file is listed, then the system checks whether replication of the file to a removable disk is acceptable. If it is, no further analysis is made, and no alerts generated. This step determines whether the security policy allows the file to be saved to removable media. If saving the file to removable media is acceptable then it is assumed that replicating the file to any machine (insider or outside the organisation) is acceptable. Therefore, no further analysis needs to be made.

3. If the replication of file to removable disk is not acceptable, the system checks whether file replication is acceptable. If it is, no further analysis is made, and no alerts generated. This step determines whether the security policy allows the file to be replicated. If replication of the file is acceptable, then it is assumed that partial replication of contents is acceptable. Therefore, no further analysis needs to be made.

4. If the replication of the file is not acceptable, the system checks whether partial replication of contents is acceptable. If it is, no further analysis is made, and no alerts generated.

5. If partial replication of contents is not acceptable, the system logs the event to the Partial_Content_Replication table of the alerts database. This step is needed to log information regarding the event and where the replicated content is saved, and for alert generation.

Alert Process:

1. The "File Server" and "File Path" or "File ID" from the alert log used to locate the record containing the details of the file, so that the "File Custodian" can be identified and alerted. This step determines the person responsible for verifying alerts associated with the file involved, and for multi-person verification purpose.

2. The "File Server" value is used to locate the record containing the details of the machine, so that the "System Administrator" can be identified and alerted. This step is needed to determine the administrator of the file server on which the affected file is hosted, so the appropriate administrator can be alerted.

7.7.5 File Transfer

In some scenarios, a misfeasor activity may result from a user transferring a file containing sensitive information to another individual who may or may not have legitimate access to the concerned file. In order to understand the situation, the entities involved in the data communication/transfer needs to be discussed and analysed.

The entities involved in this type of activity are shown in [Figure 7.3], and include the user sending the file, the machine utilised by the sender, application utilised by the sender, the file server on which the file involved is stored, the file path of the file

involved on the file server, the server mediating the communication/transfer (if it is not peer-peer transfer), the user receiving the file, the machine utilised by the user to retrieve the file, and the application utilised by the receiver to retrieve the file. Therefore, the monitoring system needs to be provided with the security related contextual information regarding all entities involved in [Figure 7.3], and stored in the *Knowledgebase* [Figure 7.2]. *Sensors* [Figure 7.2] also need to be placed within the systems and applications involved in the file transfer activity [Figure 7.3], so that the log data related to the activity and relevant for analysis of misfeasance can be fed to the *Detection Engine* [Figure 7.2] for identification of potential misfeasance.

Assuming the sender has legitimate access to the file, and the machine utilised by the sender has direct access the server hosting the file concerned. Other factors to consider in the context of data transfer activity include:

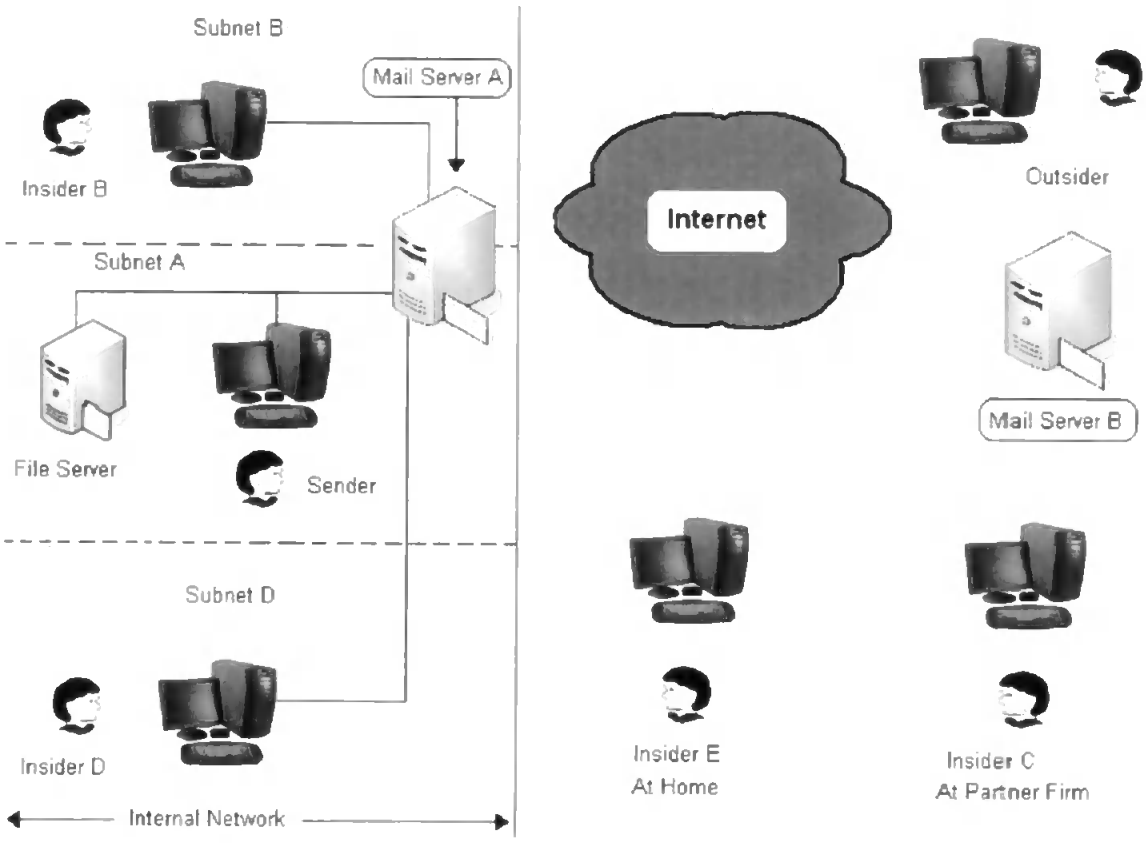


Figure 7.3 Illustration of the entities involved in data transfer

- The server mediating the transfer

  The first point of concern is the server mediating the communication or the data transfer. If this server is not managed by the organisation's IT department, the file would have left the organisation's IT boundaries and further monitoring would not be possible. Therefore, the monitoring system needs to determine whether the communication server involved is internal or external [Figure 7.3], by referring to the contextual information contained within the *Knowledgebase* described in

[Figure 7.2]. Due to this requirement the knowledgebase should contain a data table with the list of machines considered to be internally managed by the organisation [Table 7.22].

- The recipient of the data

    - Insider or outsider?

      Another fact that needs to be determined by the monitoring system is whether the recipient is an insider. The knowledgebase should contain a data table with the list of users considered as insiders of the organisation [Table 7.19]. Within the context of the research only employees of the organisation are considered as insider. In real life cases, contractors, customers, or suppliers may also be considered as insider. In such case the notion of insider would depend upon the context of the data contained within the file, and it may not be possible to provide a single list of users considered as internal. Due to this problem, the inference rules for determining the contextual insider for each unique scenario should be provided in the knowledge base, so that *Fact Processors* [Figure 7.2] can extract the information from organisation's databases. Email or communication address can be used as a parameter to determine whether the recipient is an insider [Table 7.19].

- If the recipient is an insider, does the data security policy allow the recipient to access the data?

  If the recipient is considered as an insider within the context of the file involved, the monitoring system needs to determine whether the recipient is authorised for accessing the contents of the file. Assuming that the recipient does not have required permissions to access the file at the OS and application level, the monitoring system needs to determine whether the user is authorised to receive a copy of the replicated file. Therefore, the monitoring system needs a data table containing the list of users authorised to access a replica of each inventoried file [Table 7.20].

- The machine the recipient utilised to retrieve the data

  As mentioned previously some of the machines may be prevented from accessing a file server within the subnet. Referring to Figure 7.3, due to security reasons machines from subnet B and subnet D do not have direct access to the file server within the subnet A. The security policy may allow an authorised user utilising machines within subnet B to receive files from the file server of subnet A, if transferred by a user with direct access permissions. However, the security policy may also state that machines within subnet D should not be used to retrieve mail containing files originating from the file server of subnet A. Therefore, the monitoring system needs to determine whether the machine, utilised by the

recipient of a file transfer, is authorised to receive the file. In order to infer this fact, the knowledgebase should contain a data table listing the list of machines that may receive files originated from each file server [Table 7.23].

- The application the recipient utilised to retrieve the data

  In order to accommodate further monitoring the application utilised by the recipient needs to provide audit data. For example, the recipient of a file transfer decides to forward the file to another user. Therefore, it is also important to determine the application utilised by the recipient to retrieve the transferred file.

The data tables needed to accommodate successful monitoring would look similar to those described below.

Sender Address is needed to identify the sender of the file transfer.

Machine Utilised by Sender is needed to determine where the transfer is conducted from.

File Server and *File Path* are needed to determine the source of the file involved, and the security policy associated with the file.

Communication Server is needed to determine whether the communication server mediating the file transfer is an internal machine managed by the organisation [Table 7.22].

Receiver Address is needed to determine whether the recipient of the file is an insider and whether the recipient should have access to the file involved [Table 7.20 and Table 7.21].

Machine Utilised by Receiver is needed to determine whether the machine used at the receiving end is secure enough to host the file involved [Table 7.23].

Required Parameters	Example Values
Sender Address	aung@plymouth.ac.uk
Machine Utilised by Sender	PSQ_A304_WS1
File Server	PSQ_A304_FS1
File Path	S:\Surveys\Misuse.mdb
Communication Server	192.168.3.5
Receiver Address	r.shukor@plymouth.ac.uk
Machine Utilised by Receiver	192.168.21.59

Table 7.18 File Transfer Log

The data needed for reference are:

Required Parameters	Example Values
User Name	a.phyo
Email Address	aung@plymouth.ac.uk

Table 7.19 List of Insiders

Required Parameters	Example Values
File ID	9
User Name	r.shukor

Table 7.20 Users allowed to receive each inventoried file

Required Parameters	Example Values
File ID	9
Role ID	3

Table 7.21 Roles allowed to receive each inventoried file

Required Parameters	Example Values
Machine ID	6
Machine Name	PSQ_A304_MS1
Machine Type	Mail Server
IP Address	192.168.3.5
System Administrator	D.Paul

Table 7.22 List of internal machines

Required Parameters	Example Values
Server Name	PSQ_A304_FS1
Allowed Machine	PSQ_B201_WS9

Table 7.23 List of machines allowed to receive files from each server

Monitoring Process:

1. The "File Server" and "File Path" from the event log are compared against the listed files to determine whether the accessed file requires monitoring. This step is needed to determine the security policy associated with the file involved.

2. If the file is listed, then the system checks whether replication of the file to a removable disk is acceptable. If it is, no further analysis is made, and no alerts are generated. This step determines whether the security policy allows the file to be saved to removable media. If saving the file to removable media is acceptable then it is assumed that replicating the file to any machine (insider or outside the organisation) is acceptable. Therefore, no further analysis needs to be made.

3. If replication to a removable disk is not acceptable, the system checks whether the server mediating the communication/transfer is an internal machine by checking in the list of internal machines. If the server mediating the communication/transfer is not an internal machine, the event is logged in to the File_Transfers table of the alerts database with the alert status set to true. If replication of the file to a removable media is not acceptable, it is assumed that the file must remain within the organisation's internal machines. Therefore, this step determines whether the server mediating the transfer is an internal machine. If the server mediating the

transfer is not an internal machine the security of file cannot be further managed/regulated by organisation, and thus need to be alerted.

4. If the server mediating the communication is an internal machine, the system checks whether the recipient is an insider by checking the recipient's address in an appropriate registry (Employees, Students, Contractors, Customers etc.). If the recipient's address is not found in the registry, the event is logged into the File_Transfers table of the alerts database with the alert status set to true. This step determines whether the recipient of the transfer is an insider, because if the recipient is not an insider then the confidentiality of the file will be compromised.

5. If the recipient's address is found in the registry, the system gets the name (or user name), and assigned role of the recipient. It then checks whether the recipient's role is authorised to access the concerned file by checking in the data table containing the roles allowed access to each listed file. This step determines whether it is acceptable for the recipient to have access to the file, based on the role the recipient belongs to, because if the recipient does not have necessary clearance for access, the confidentiality of the data will be compromised.

6. If the recipient's role is not found in the list of roles allowed to access the concerned file, the system checks whether the recipient is allowed access to the file by checking the data table containing the list of users allowed access to each

listed file. This step determines whether it is acceptable for the recipient to have access to the file, because if the recipient does not have the necessary clearance for access, the confidentiality of the data will be compromised.

7. If both the user and the user's role are not allowed access to the concerned file, the event is logged in to the File_Transfers table of the alerts database with the alert status set to true. This step logs the event if the recipient does not have necessary clearance to access the file involved, because the confidentiality of the file is compromised as a result of the activity.

8. If either the user or the user's role is allowed access to the concerned file, the system checks whether the machine utilised by the recipient to retrieve the file is allowed access to the file server from which the file originated by checking the data table containing the list of machines allowed access to each file/database server. This step determines whether the machine utilised by the recipient is secure enough to host the file involved in the transfer, because if the machine utilised by the recipient does not meet the security requirements (not managed by the organisation, or does not have security controls) then the security of the file can be compromised inadvertently.

9. If the machine utilised by the recipient is not allowed access to the file from which the file originated, the event is logged in to the File_Transfers table of the

alert database with the alert status set to true. This step logs the event if the machine utilised by the recipient does not have appropriate security to host the file involved in the transfer.

10. If the machine utilised by the recipient to retrieve the file is allowed access to the file server from which the file originated, then the analysis process logs the event in to the File_Transfers table of the alert database with the alert status set to false. This step merely logs the file transfer to keep track of the files for further monitoring, and does not generate any alerts because current activity satisfy all security requirements.

Alert Process:

1. The "File Server" and "File Path" from the alert log used to locate the record containing the details of the file, so that the "File Custodian" can be identified and alerted. This step identifies the appropriate person to be alerted for multi-person verification purpose.

2. The "File Server" value is used to locate the record containing the details of the machine, so that the "System Administrator" can be identified and alerted. This step identifies the administrator of the system on which the originating file is hosted, so that the appropriate administrator can be alerted.

7.8 Database Access

Issues concerning database access by insiders include.

- The number of database queries issued by each user
- The number of records returned per query
- The number of records accessed per user within a time frame
- The percentage of a data table accessed per user within a time frame
- The percentage of a database accessed per user within a time frame

Many database systems include statistical analysis features to monitor information such as the number of records viewed by a user, the number of records updated by a user, and the number of records returned by a user query. However, such techniques will not be able to detect a query affecting a single record. For example, a police officer checking the criminal records of a neighbour may constitute misuse if there was no valid reason for access although the operation itself is legitimate and part of the job. Confidentiality or integrity of a record(s) may be compromised if each access is not verified. When a user updates or views a record, the monitoring system needs to verify the validity of the access. However, in order to accommodate this kind of verification, the monitoring system needs to be provided with information regarding how the validity of the access or integrity of the record may be verified.

Assuming the user has authorised access and that the query issued is legitimate, other factors to consider in the context of a database access and the following question needs to be answered.

- If it is read access, is the access valid i.e. is there a valid reason for access?
    i. Can the validity of the access be checked through referencing another record?

- If it is write access, the integrity of the record needs to be verified.

- Each attribute within the affected record needs to be verified for contextual integrity.
    i. Can the integrity of each data field within the affected record be checked by referencing another record?

Assuming the answers to the questions asked is positive. The successful monitoring depends upon being able to identify

- The database, data table, and the record affected

  The monitoring system may need to monitor a number of databases, and each database may contain numerous data tables containing a large number of records.

- An appropriate record required for reference

  The reference record depends upon the affected database, data table, and record. The monitoring system must be able to establish a unique link between the affected record and the reference record. Therefore, the monitoring system needs to be provided with information regarding the identification of a reference record.

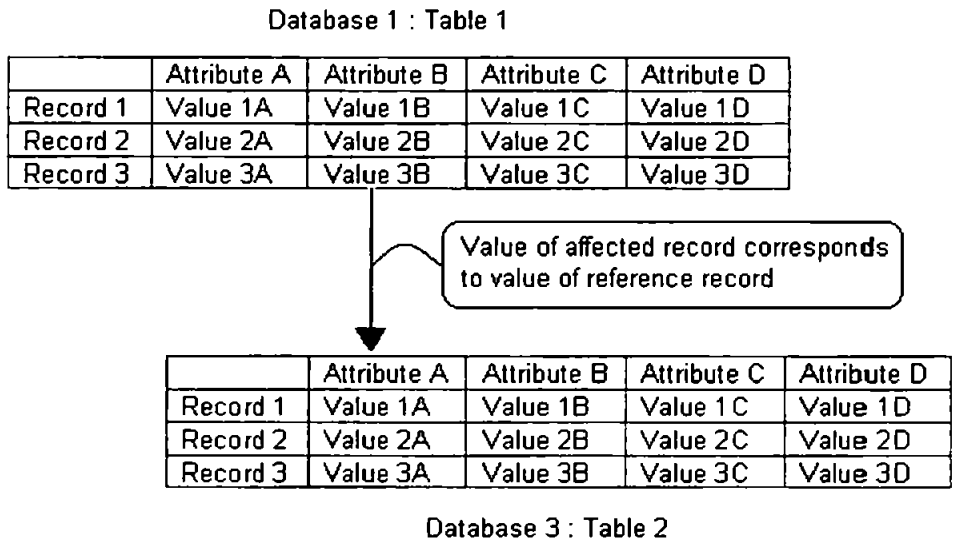


Figure 7.4 Relationship between affected record and reference record

- Under certain circumstances, validity of the access can be verified by ensuring existence of a reference record. In some cases the monitoring system may need to test a value within the reference record or compare a value of affected record and that of the reference record. In such cases, the monitoring system needs to be provided with information regarding the value(s) that need to be tested, and the conditions of a successful test.

The data needed to log are:

Required Parameters	Example Values
User Name	A.Phyo
Machine Utilised	PSQ_A304_WS1
Database Server	PSQ_A304_DBS1
File Path	D:\CustomerRecords.mdb
Table Name	Account Details
Affected Record ID	7

Table 7.24 Database Access Log

User Name is needed to identify the user responsible for the activity.

Machine Utilised is needed to identify where the activity was carried out from.

Database Server and *File Path* are needed to identify the database affected, and to identify the person responsible for verifying the integrity of the database. *Table Name* is needed to identify the data table affected, and to identify the person responsible for verifying the integrity of the data table. A unique combination of *Database Server*, *File Path*, and *Table Name* are needed to identify the appropriate reference data required for automated verification of the integrity of the record affected.

Affected Record ID is needed to identify the record that is affected as a result of the activity, and to identify the appropriate reference data required for automated verification of the affected record.

The data needed for reference are:

- List of monitored data tables (or queries) requiring verification
- List of associated reference data for each listed query

List of monitored data tables (or queries) requiring verification

Required Parameters	Example Values
Query Name	Detail Update
Database Server	PSQ_A304_DBS1
File Path	D:\CustomerRecords.mdb
Table Name	Account Details
Table Custodian	J.Jones
Primary Key	Account ID
Link Key	Account ID
Attribute to be Tested	Update ID

Table 7.25 Data required for identifying the affected record

Query Name is needed to identify the reference data required for automated verification of the affected record. There may be more than one query available for each data table. Therefore *Database Server*, *File Path*, *Table Name, and Query Name* combination is used as a unique identifier.

Table Custodian is needed to identify the person responsible for manual verification of the data table when the reference data needed for automated verification of the affected record cannot be located by the system.

Primary Key is needed to identify the primary key attribute of the affected data table, so that the affected record can be located.

Link Key is needed to identify the attribute that can link the affected record to the reference record, so that an attribute from the affected record can be tested against an attribute from the reference record.

Attribute to be Tested is needed to identify the attribute of the affected record that needs to be tested against the reference record.

List of associated reference data for each listed query:

Required Parameters	Example Values
Query Name	Detail Update
Database Server	PSQ_A304_DBS1
File Path	D:\CustomerService.mdb
Table Name	Account Update Requests
Table Custodian	B.Marley
Link Key	Account ID
Attribute to be tested against	Update ID
Condition for testing	Equals

Table 7.26 Data required for identifying the reference record

Query Name is needed to represent the corresponding record that contains information (*Database Server*, *File Path*, Table Name, *Link Key*) to identify the reference data for automated verification. The *Query Name* for identifying the reference record corresponds to the *Query Name* for identifying the affected record.

Database Server and *File Path* are needed to identify the database server and database file containing the reference record.

Table Name is needed to identify the data table containing the reference record.

Table Custodian is needed to identify the person responsible for manual verification of the data table containing the reference record. In case the reference record needed for automated verification cannot be located, the *Table Custodian* of the affected record and the *Table Custodian* of the reference record may need to communicate for manual verification.

Link Key identifies the attribute within the reference record that can be used to link the affected record and the corresponding reference record.

Attribute to be Tested Against identifies the attribute within the reference record that needs to be tested against an attribute of the affected record.

Condition for Testing defines the condition that must be hold true, when testing the attributes from the affected and the reference record, so that the validity/integrity of the affected record is ensured.

Monitoring Process:

1. The "Database Server", "File Path", and "Table Name" values from the event log are compared against the listed data tables to determine whether the affected datable requires monitoring. In future developments query ID may be used instead of the "File Path" and "Table Name" in order to identify affected data table(s). This step is needed to identify the contextual rules that must be conformed in order to ensure the validity/integrity of access.

2. If the query or data table is not listed, then no further analysis is made, and exist the analysis procedure without generating any alerts.

3. If the data table is listed. The monitoring system notes the "Query Name" and locates the corresponding entry in the list of associated reference data, so that the data required for verification can be located. This step is needed to identify the record that contains information for locating the reference data

4. Now, the value of the "Affected Record ID" from the event log, and the value of "Primary Key Attribute" from the listed data tables are used to locate the affected record in the data table indicated by the values of "Database Server", "File Path", and "Table Name" of the event log. This step is needed to identify the affected record, so that its integrity can be verified.

5. Once the affected record is located, the value stored within the attribute indicated by the "Link Key" is used to search the corresponding reference record. This step is needed to identify the reference record, so that an attribute from the affected record can be compared against an attribute from the reference record.

6. If the condition of testing is "Exist" and a corresponding reference record is not found, the event is logged into the "Database_Access" table of the alerts database. This step checks whether a reference record exists. If a reference record does not exist the integrity of the affected record is in doubt, and thus requires manual verification by the Table Custodian of the affected data table.

7. If the condition of testing is "Equals", the value stored within the attribute indicated by "Attribute_To_Be_Tested" of the affected record is compared against the value stored within the attribute indicated by "Attribute_To_Be_Tested_Against" of the reference record, to determine whether the condition is satisfied. If the condition is not satisfied, the event is logged into the Database_Access table of the alerts database. This step compares a value from the affected record against a value from the reference record, and the two values must be equal in order to ensure the integrity of the affected record.

8. If the condition of testing is "True", the value stored within the attribute indicated by "Attribute_To_Be_Tested_Against" of the reference record is checked. If the value contained within the attribute indicated by "Attribute_To_Be_Tested_Against" is false, the event is logged into the Database_Access table of the alerts database. This step test whether the identified attribute in the reference record is "True", to ensure the validity of the access.

Alert Process:

1. The "Database Server", "File Path", and "Table Name" from the alert log are used to locate the record containing the details of the file, so that the "Table Custodian" can be identified and alerted. This step identifies the person responsible for manual verification of the data table affected, so that when the integrity/validity of the access is in doubt, manual verification can be requested.

7.8.1 Registry Management

To an information system, whether a user should have access or not depends on whether the concerned user exists in the registry referred to by the access control (or authentication and identification) system. Therefore, anyone who can add an entry to the registry may abuse the privilege if the additions of new records are not verified. Within an organisation, more than one registry may exist for various purposes, and thus the person responsible for each registry may also vary. Therefore, it is critical that the right

personnel are alerted so that the verification will be authentic, and prevent abuse by privileged users. However, in order to accommodate this facility, the monitoring system needs to know which registries/lists (data tables, or categorised lists within each table) need to be monitored, and how to determine the person responsible for verification. The monitoring approach is similar to verifying the validity of database access. The difference is that the reference record is used for identification of the person responsible for verification.

Data needed to log are:

Required Parameters	Example Values
User Name	A.Phyo
Machine Utilised	PSQ_A304_WS1
Database Server	PSQ_A304_DBS1
File Path	D:\UserAccounts.mdb
Table Name	UserRoles
Affected Record ID	7

Table 7.27 Registry Update Log

User Name is needed to identify the person responsible for the activity.

Machine Utilised is needed to determine where the activity was carried out from.

Database Server, *File Path*, and *Table Name* are needed to identify the data table affected.

Affected Record ID is needed to identify the record affected as a result of the activity and require verification.

Data needed for reference are:

Required Parameters	Example Values
Registry Name	User Roles
Database Server	PSQ_A304_DBS1
File Path	D:\UserAccounts.mdb
Table Name	User Roles
Primary Key	Record ID
Link Key	Role Name

Table 7.28 List of registries to be monitored

Registry Name is needed to locate the record containing information to identify the person responsible for verification of the affected record.

Database Server, *File Path*, and *Table Name* are needed to identify the data table affected.

Primary Key is needed to locate the affected record within the affected data table.

Link Key is needed to identify the attribute within the affected record that can be used to locate the corresponding reference record.

Required Parameters	Example Values
Registry Name	User Roles
Database Server	PSQ_A304_DBS1
File Path	D:\UserAccounts.mdb
Table Name	Roles
Link Key	Role Name
Attribute Containing Authority to be Informed	Role Manager

Table 7.29 Information to locate appropriate authority to be alerted

Registry Name is needed to represent the record containing information for identifying the person responsible for verification of the affected record.

Database Server, *File Path*, and *Table Name* are needed to locate the data table containing the identity of the person responsible for verification of the affected record.

Link Key is needed to locate the corresponding reference record containing the identity of the person responsible for verification of the affected record.

Attribute Containing Authority to be Informed is needed to identify the attribute that contains the identity of the person responsible for verification of the affected record.

Monitoring Process:

1. The "Database Server", "File Path", and "Table Name" values from the event log are compared against the listed registries to determine whether the affected

datable requires monitoring. This step is needed to determine whether the validity/integrity of the access needs to be verified.

2. If the registry is not listed, then no further analysis is made, and exist the analysis procedure without generating any alerts.

3. If the registry is listed. The event is logged into the "Registry_Management" table of the alerts database. This step logs the events as requiring verification by appropriate authority.

Alert Process:

1. The monitoring system notes the "Registry Name" and locates the corresponding entry in the list of associated reference data, so that the data required for identifying the authority to be alerted can be retrieved. This step locates the record containing information for identifying the record that contains the identity of the person responsible for verification of the affected record.

2. Now, the value of the "Affected Record ID" from the event log, and the value of "Primary Key Attribute" from the listed registries are used to locate the affected record in the data table indicated by the values of "Database Server", "File Path", and "Table Name" of the event log. This step locates the affected record so that

the corresponding reference record containing information for identification of the person responsible for verification can be located.

3. Once the affected record is located, the value stored within the attribute indicated by the "Link Key" is used to search the corresponding reference record containing the authority to be alerted, stored in the data table indicated by "Database Server", "File Path", and "Table Name" of the "Registry Custodian" list. This step uses a value from the affected record in order to locate the corresponding reference record containing the identity of the person responsible for verification of the affected record.

4. Once the corresponding reference record is found, the authority indicated by the value stored within the attribute indicated by "Attribute_Containing_Authority" is alerted. This step requests the responsible person identified in the previous step to manually verify the affected record in order to ensure integrity.

7.8.2 Fraud Detection

Although ensuring integrity of data by verification of each record after an update minimise the risk of fraud occurring as a result of corrupted database, it does not guarantee detection of fraud. Fraud can result from creative accounting, and loopholes within reward policy, which requires human judgement to determine occurrence of fraud.

Fraud occurs mostly in areas where some kind of purchase, inventory, sales, payment, and accounting or performance analysis is involved, and usually related to some type of accounting or recording of trade/transaction. Therefore, performance/accounting data is required for fraud monitoring. Computer aided fraud monitoring tools already exist (Coderre 1999), and the methods applied are fairly trivial such as indexing, sorting, sequencing, stratifying, classifying, counting, counter verifying calculations, performing statistical analyses, finding gaps and duplicates. However, it is critical that the correct analysis method is applied to the relevant data, and the difficulty lies in identifying the relevant data within the application and associated contextual rules for fraud monitoring. Therefore, the most important factor in detecting fraud is the in-depth understanding of the business application and contextual conditions related to each business transaction.

Since business applications and contextual conditions differ from one organisation to another, it would be fair to say that it would be extremely difficult if not impossible to develop an automated fraud-monitoring tool that will suit all organisations. A generic fraud-monitoring tool will work only if the data structures for business applications are identical and contextual rules related to each and every business transactions are identical. Since businesses try to make money by operating differently to their competitors in one way or another, it would be naïve to assume that it will be possible for one business to have the same contextual rules as another in the same line of business. In addition, contextual rules related to transactions for today may not be the same in a

week's time as the business strive to be competitive and develop creative/attractive offers to their potential and existing customers.

At most the tool will be able to provide the auditor with the charts and graphs comparing the performance statistics of an employee to the rest in the same role. However, it depends upon the auditor to decide the parameters that need to be included for analysis, and human judgement is required to determine whether fraud has occurred by applying the knowledge of the application and related contextual conditions.

7.9 Conclusions

This chapter highlighted the activities that may lead to misuse and presented a misfeasor monitoring system to detect misfeasance that may result from the identified activities. Each activity monitored requires varying analysis procedure and relevant parameters. Misfeasance is a contextual perception as explained in Chapter 2, section 2.3.1, and thus information related to the context in which the activities are analysed need to be included for analysis of possible misfeasance. Within the architecture (Figure 7.2), the contextual information is fed to the *Knowledgebase* of the monitoring system through *Fact Processors*. The context in which the activities are considered needs to be provided to the monitoring system through appropriate analysis/inference procedures. It is also important to segregate duties by distributing alerts to authorities who have intimate knowledge of the content and the context in which the activity is acceptable.

The chapter presented a misfeasor monitoring architecture that accommodates analysis of various user activities, allowing the utilisation of audit logs from heterogeneous systems. The architecture facilitates, and also shows methods for, extraction of relevant contextual information from organisation's various databases for analysis of misfeasance. Suitable analysis/inference procedures were also developed and presented, so that the activities can be analysed within the context of acceptable usage.

Within the presented architecture, the process of distributing alerts have been decoupled from the analysis procedures, so that the alerts can be sent to the authority with the most intimate knowledge of the system/content affected. This approach consequently provides the mechanism to implement segregation of duties for authorities verifying the alerted activity. The proposed architecture accommodates detection of some of the misfeasor activities identified in Chapter 5, and in particular:

- File access through arbitrary applications
- Information disclosure/theft
- Dissemination of data
- Unauthorised file sharing
- Access from unauthorised machines
- Inappropriate modification of system settings/configurations
- Account creation
- Inappropriate modification/access of database records

- Inconsistent database records

In the next chapter, a prototype system built upon the architecture described in this chapter is used to demonstrate the detection of misfeasor activities considered here.

Chapter 8 Prototype Misfeasor Monitoring System

8.1 Introduction

This chapter presents a prototype misfeasor monitoring system built upon the architecture described in the previous chapter, utilising the methods of data extraction, and analysis procedures presented. The aim is to provide proof that existing detection technology and strategies can be employed in a novel way for misfeasor monitoring, and that components proposed in the previous chapter are relevant for extracting contextual data and distribution of alerts to appropriate authority. This chapter also aims to validate the requirement of data identified in Chapter 5 for monitoring of misfeasance, emphasising the need to provide the system with data relevant to misfeasance for successful detection. The prototype system also evaluates the applicability of analysis procedures suggested in the previous chapter, and demonstrates how the contextual conditions governing the acceptable usage can be implemented through appropriate inference/analysis procedures.

The prototype system is evaluated on the basis of its ability to detect a number of activities within a generic organisation that can result in misfeasance as identified in Chapter 5, which includes:

- File Access through an arbitrary application
- Deletion of files considered as property of the organisation
- Replication of confidential files
- Replication of confidential content
- Dissemination of confidential files

- Modification of database records
- Addition of users or new records to organisation's registry databases
- Modification of system/application settings

The activities included here is only a subset of the activities listed in Chapter 5, because other forms of misuse excluded from the above list can be detected by existing detection and monitoring tools.

Visual Basic was used for the development of the prototype system, and Microsoft Access was used for creating databases and data tables required by the monitoring tool. Visual Basic and Microsoft Access were chosen due to the fact that they provide all the features needed for the validation of the concept, development of graphical interface and rapid development of the prototype system. Programming language C/C++ was not used, because system level programming was not needed in order to validate the relevance of data identified as a requirement for successful monitoring. Although the system developed in Visual Basic and Microsoft Access would run only on Microsoft Windows systems, it can analyse data collected from heterogeneous systems, because the data identified for analysis is generic across all systems and not Microsoft Windows specific. Source code of the prototype system and databases are included in the accompanying CD at the back of the thesis.

8.2 Overview

The prototype system consists of three modules: *Event Generator*, *Event Analysis Engine*, and *Alert Generator*.

The prototype monitoring system architecture (Figure 8.1) resembles the conceptual system described in the previous chapter (Figure 7.2), although the conceptual version is more modularised. *Fact Processor* and *Parser* components described in the conceptual version are embedded within the *Analysis Engine* in the prototype version. *Analysis Engine* within the prototype version includes only the *Inferential component* and does not include *Statistical component*.

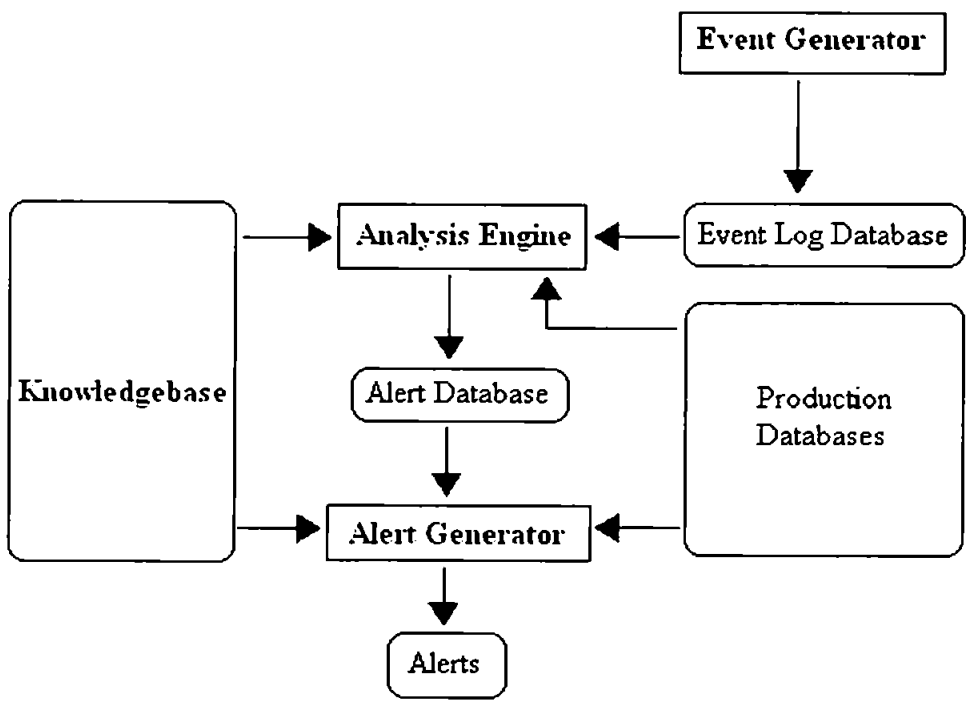


Figure 8.1 Components of the Prototype Misfeasor Monitoring System

The prototype system makes use of three databases, *Events Database*, *Knowledge Database* (acceptable usage policies are also stored here), and *Alerts Database*, for the analysis and alerting of possible misfeasor activity to the appropriate authority. The functionality of the monitoring tool depends upon the three databases mentioned, and thus it would be appropriate to explain the purpose and details of each database to discuss how each user activity is analysed to determine whether possible misuse activity is in progress.

8.3 Event Generator

Due to the absence of fully functional data collection agents, and lack of application level user activity logs, an event generator is required to simulate user activities for the scenarios considered. *Event generator* simulates the logging of user activity and relevant data associated with each activity within the concerned application environment. The logs generated for each activity will be analysed by the *Event Analysis Engine* (decision making logic/rules are embedded within it) in order to determine possible misuse. *Event generator* is used because the log data required for misfeasor analysis is not provided by currently available desktop applications, and to actually get the required data from a commercial application will take enormous efforts to modify the application without the availability of the application source code.

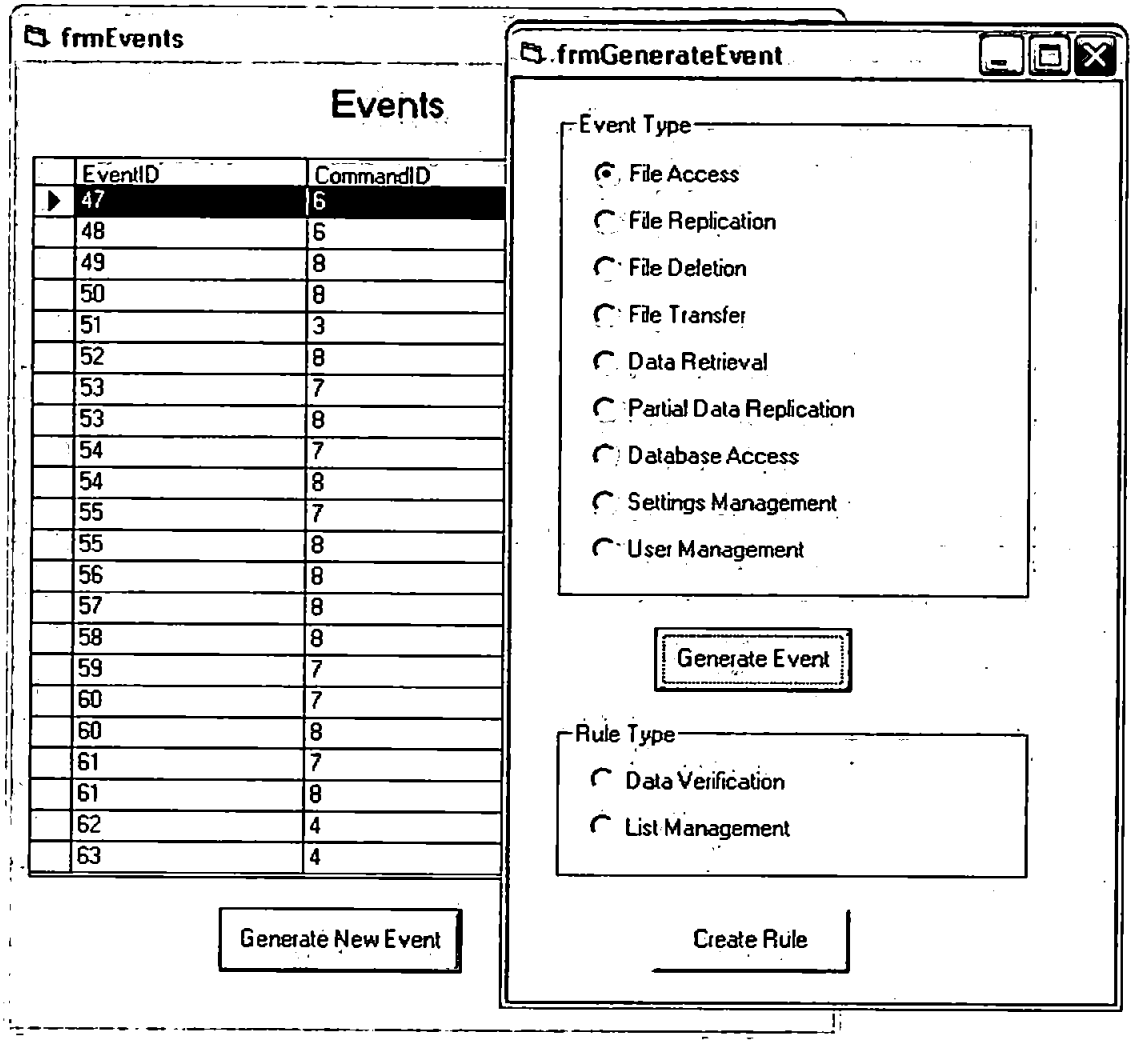


Figure 8.2 Event Generator Main Interface

By demonstrating the relevance of the data within certain applications, it will also emphasise the need for application vendors/developers to include features to provide log data outlined in each scenario, which is required for misfeasor monitoring. *Event generator* logs relevant data in the *Events Database*. *Events Database* consists of a data table listing all events and separate data tables to log the relevant details regarding each type of user activity. The data tables within the *Events Database* are:

8.3.1. Events

The *Events* table is used, so that all identified activities that may lead to misuse can be logged in chronological/sequential order. The *Events* table contains three attributes, *EventID*, *CommandID*, and *EventType*.

EventID: is used for identification of each unique event, and for sequential ordering of events.

CommandID: is used to identify the appropriate analysis procedure for misfeasor analysis of the concerned event.

EventType: describes the nature of the event and is related to the *CommandID*.

The code responsible for interpreting the values within this table, which identifies an appropriate analysis procedure for each event, corresponds to the Parser component described in the contextual architecture. Other relevant details regarding each event type are logged in their respective log tables within the *Events* database.

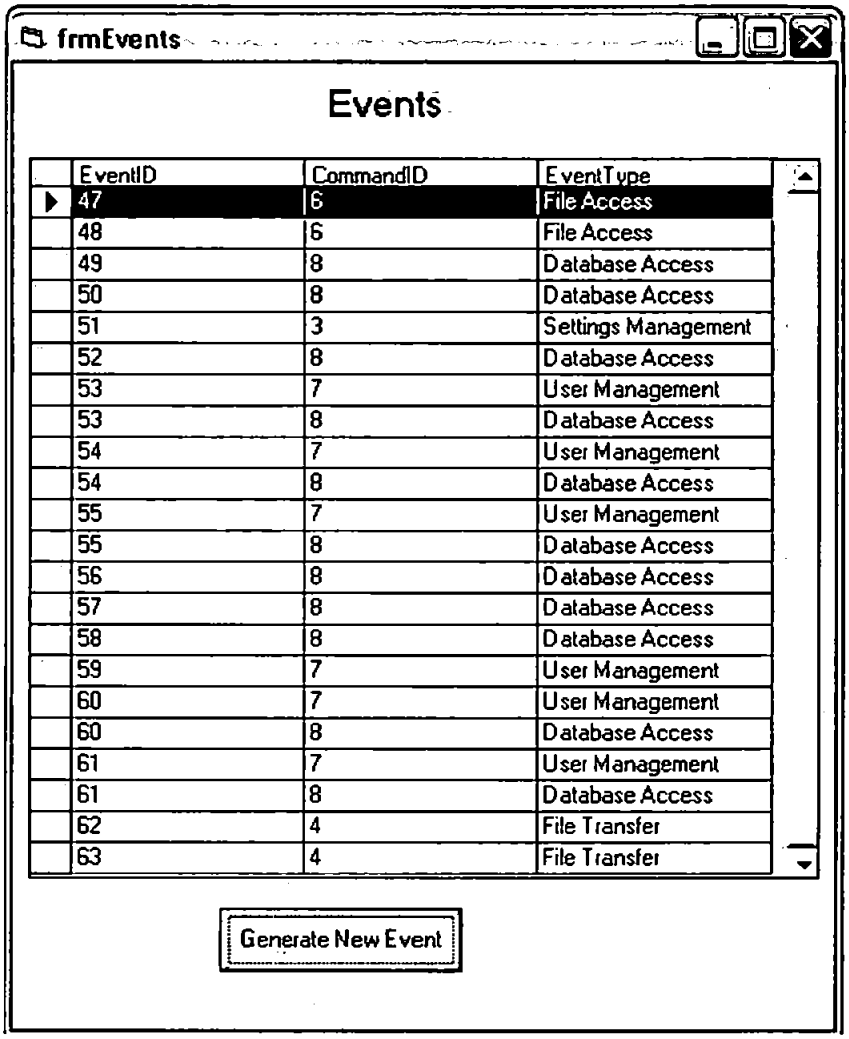


Figure 8.3 Events Interface

8.3.2 File Access

The *File Access* table is used for logging data regarding each user's file access details. The *File Access* table contains six attributes, *EventID*, *UserName*, *MachineName*, *ApplicationName*, *ServerName*, and *FilePath*.

EventID: The *EventID* is related to the *EventID* attribute in the *Events* table. This is used, so that the analysis procedure will be able to link the two records together, i.e. the *Events* table and the *File Access* table.

UserName: This attribute is used to store the name of the user responsible for the activity.

MachineName: This attribute is used to store the name of the Machine from which the user performed the activity.

ApplicationName: This attribute is use to store the name of the application the user utilised for accessing the file.

ServerName: This attribute is use to store the name of the machine on which the accessed file is stored.

FilePath: This attribute is used to store the exact file path of the concerned file on the server.

The combination of *Server Name* and *File Path* is used to identify each unique file within the IT network.

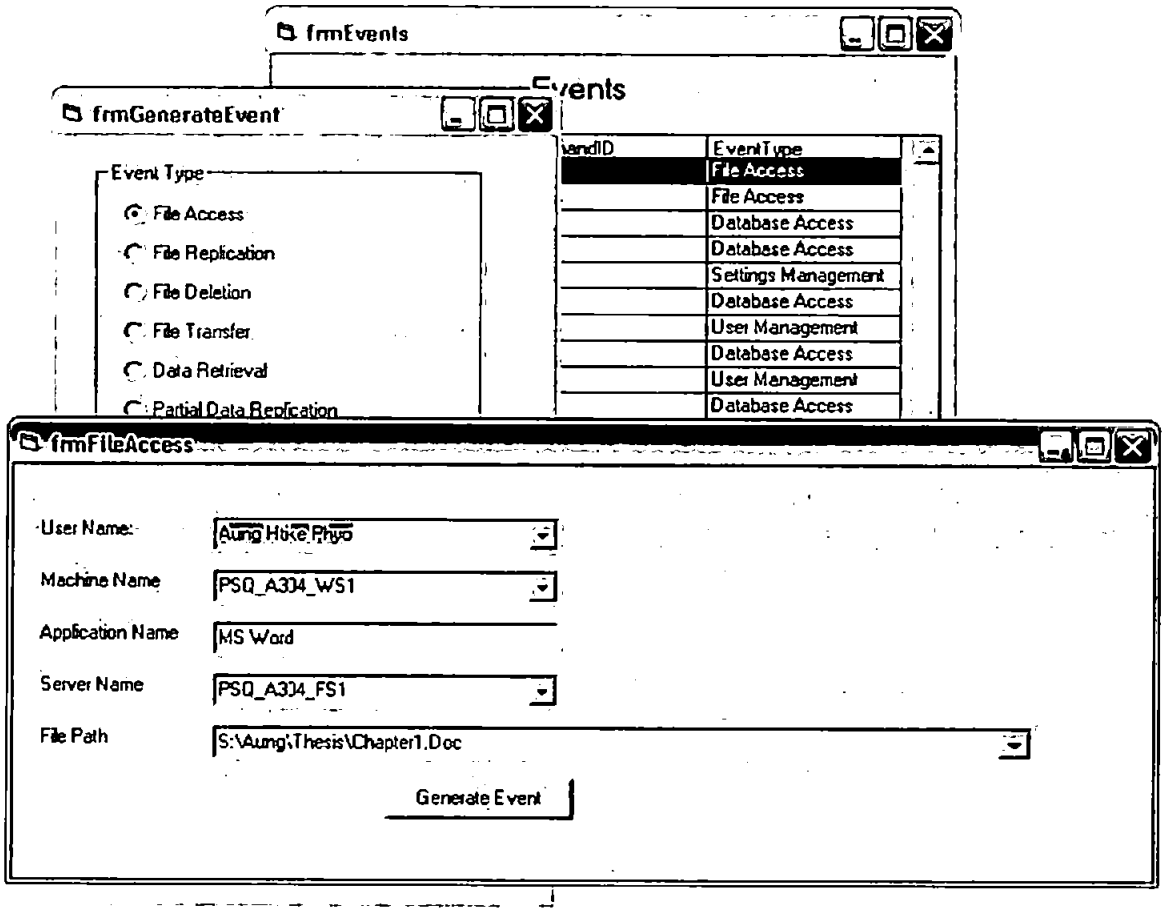


Figure 8.4 File Access Log Generator Interface

8.3.3 File Deletions

The *File Deletions* table is used for logging data related to file deletions by each user. The *FileDeletions* table contains six identical attributes to *FileAccess* table.

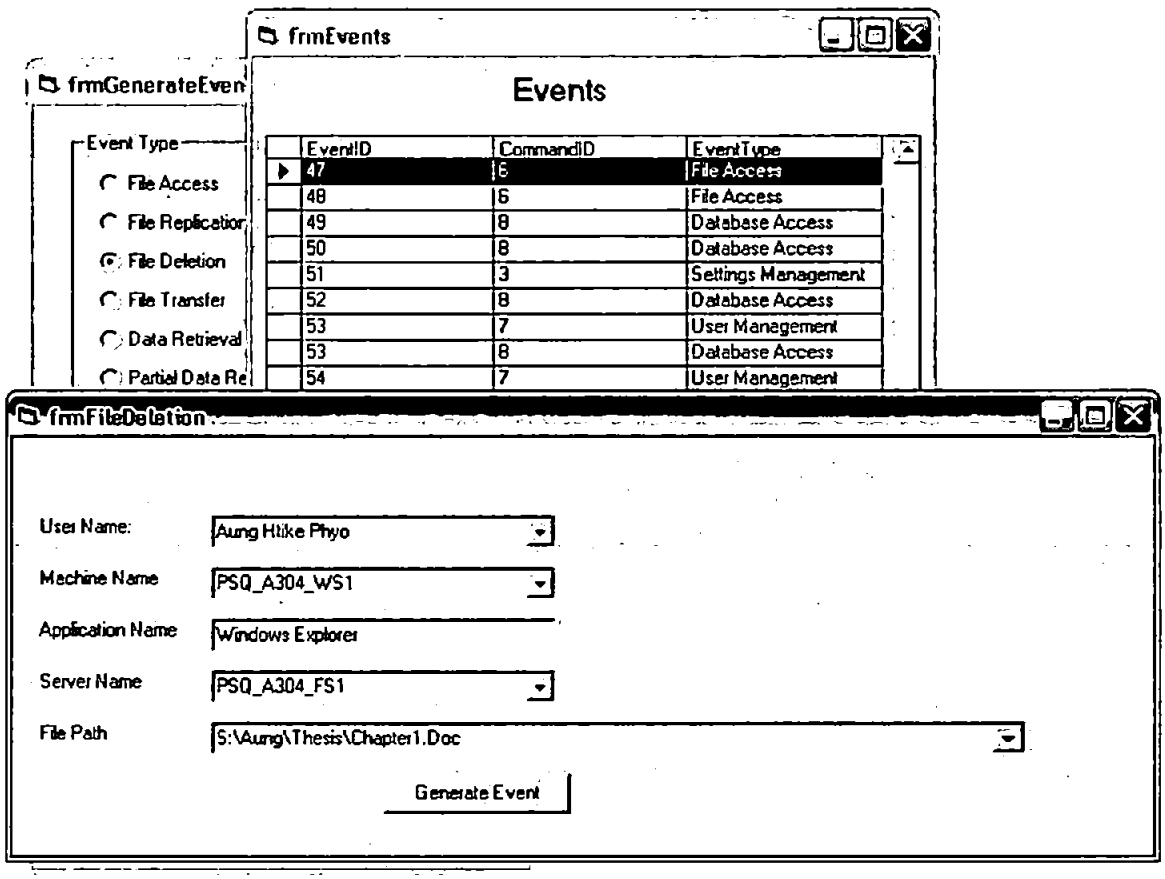


Figure 8.5 File Deletion Log Generator Interface

8.3.4 FileReplications

The *FileReplications* table is used for logging data related to replication of files by each user. The *FileReplications* table contains seven attributes, six of them are identical to those in *FileAccess* and *File Deletions* tables, and a *CommandID* is added to separate capturing of data (Copy/Cut events) from reproduction of captured data (Paste events). For the data capture (Copy) events, the *Server Name* indicates the machine the source file is located on, and the *File Path* indicates the location of the file on the machine. For the

reproduction (Paste) events, the *Server Name* indicates the machine to which the file is copied, and the File Path indicates the location of the replicated file.

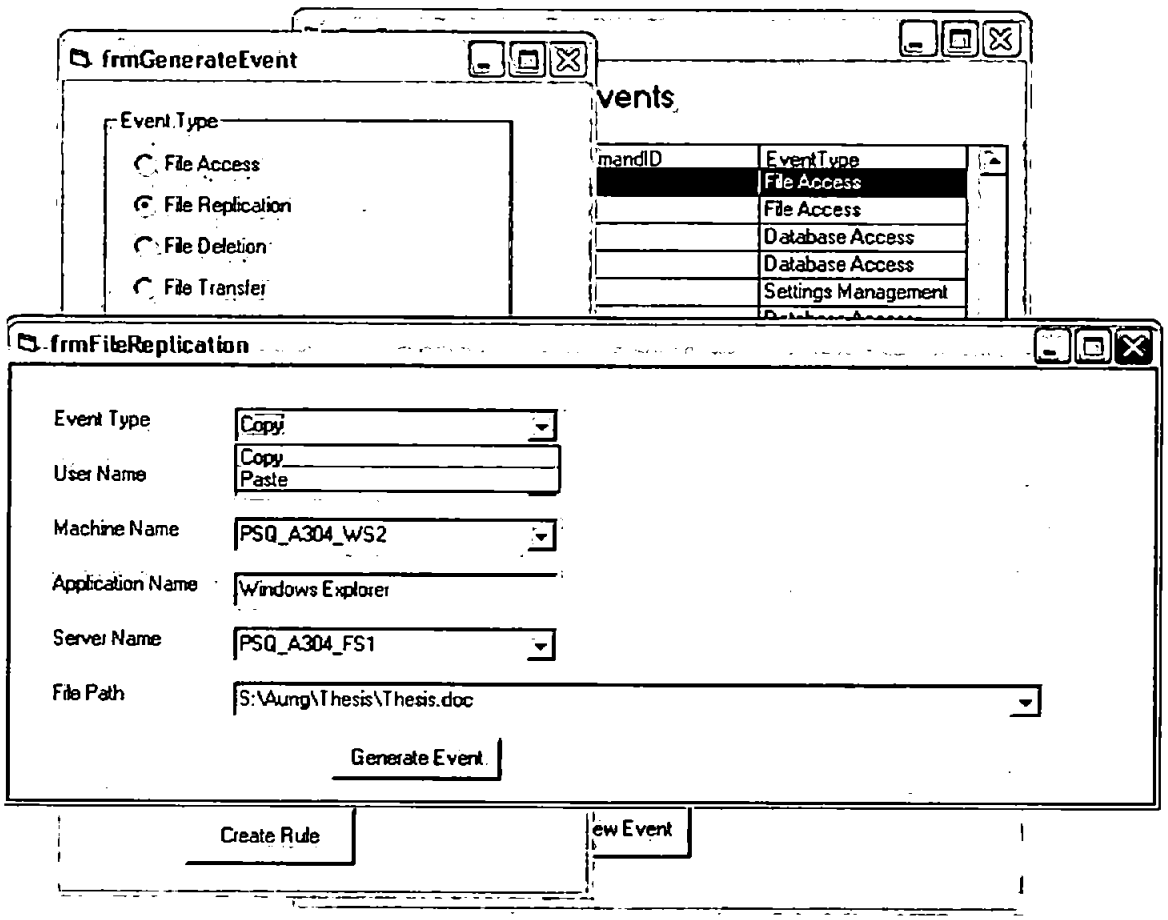


Figure 8.6 File Replication Log Generator Interface

8.3.5 DataReplications

The *DataReplications* table is used for logging data related to partial replication of information by each user through applications (such as word processors) that provide direct access to the contents of a file. The *DataReplications* table contains seven

attributes identical to those in *FileReplications* table. Again the capturing of the data and reproduction of the captured data are treated separately.

8.3.6 Clipboard

The Clipboard table is used for logging clipboard data resulting from data capture (Copy/Cut) events for use in analysing file and data replication activities. The two activities are treated separately because capturing the data does not necessarily mean it will be reproduced. The Clipboard table contains five attributes, *EventID*, *CommandID*, *UserName*, *MachineName*, and *SourceFileID*.

SourceFileID: This attribute is used to store the *FileID* (a unique identifier within the file inventory table of knowledge database) of the file from which the file/data is copied.

8.3.7 DatabaseAccess

The *DatabaseAccess* table is used for logging data related to database access by each user. The *DatabaseAccess* table contains eight attributes; six of them are identical to those in *FileAccess* table, with the addition of *TableName*, and *RecordID*.

TableName: This attribute is used to store the name of the data table affected.

RecordID: This attribute is used to store the identifier of the record affected. This value is returned by the database application. The example in the figure, adds a new employee to the employees data table of the ContextualInfo database located on the PSQ_A304_FS1 server, and returns the affected record's ID to the event generator. For the purpose of the demonstration, it is assumed that only one type of query is available to the user for each data table and that each query affects only one data table. In future developments of the system, layered analysis may be employed to detect various queries, such as utilising query signatures to identify the query prior to associating each query with an analysis query or a counter verification query.

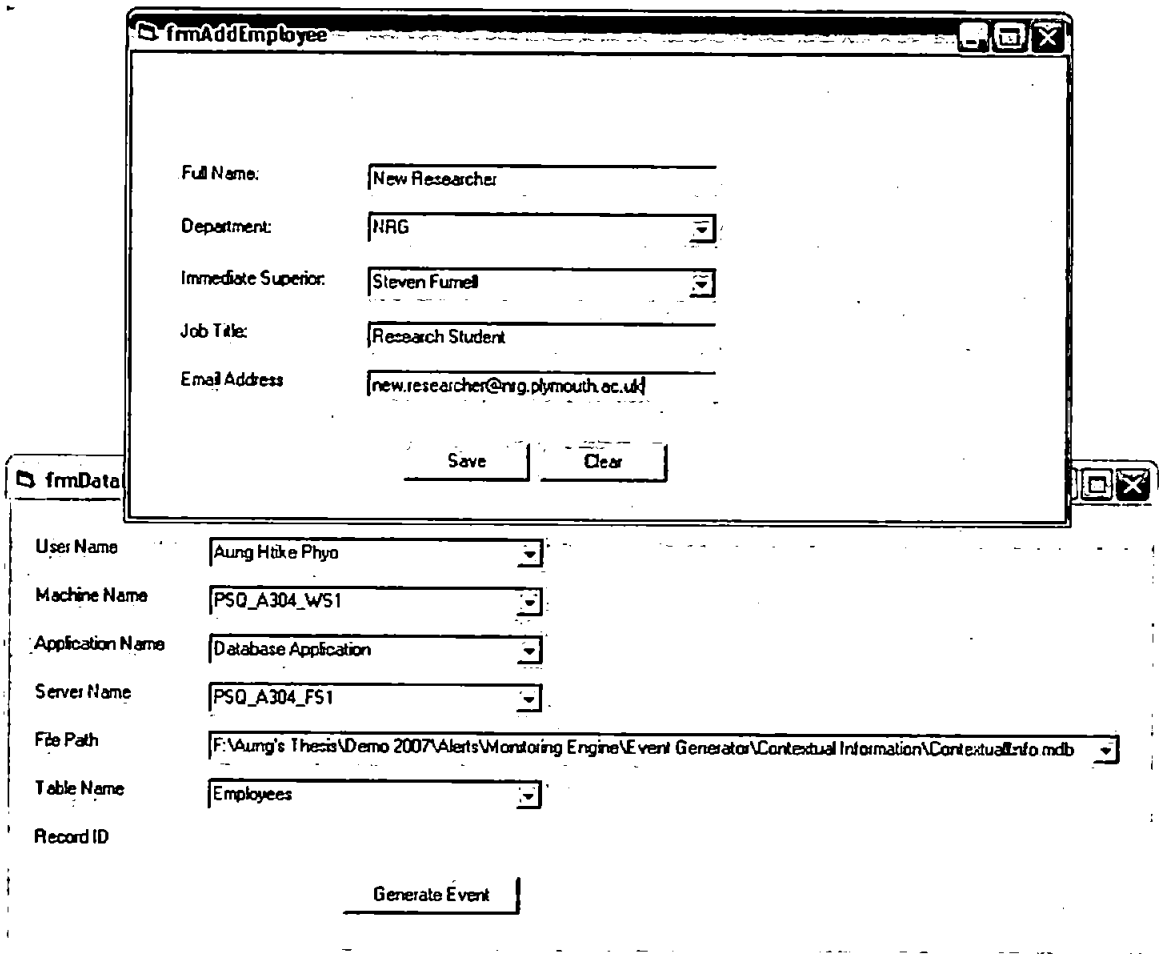


Figure 8.7 Database Access Log Generator Interface

8.3.8 Registry Management

The *Registry Management* table is used for logging data related to addition of individuals to lists of employees/users/students etc. Each registry represents a white list for relevant activities within the context of the prototype system. The *Registry Management* table contains eight attributes identical to those in the *Database Access* table.

8.3.9 Settings

The Settings table is used for logging data related to updating/changing of system/application configurations/settings. The Settings table contains four attributes, *EventID*, *UserName*, *MachineName*, and *ApplicationName*.

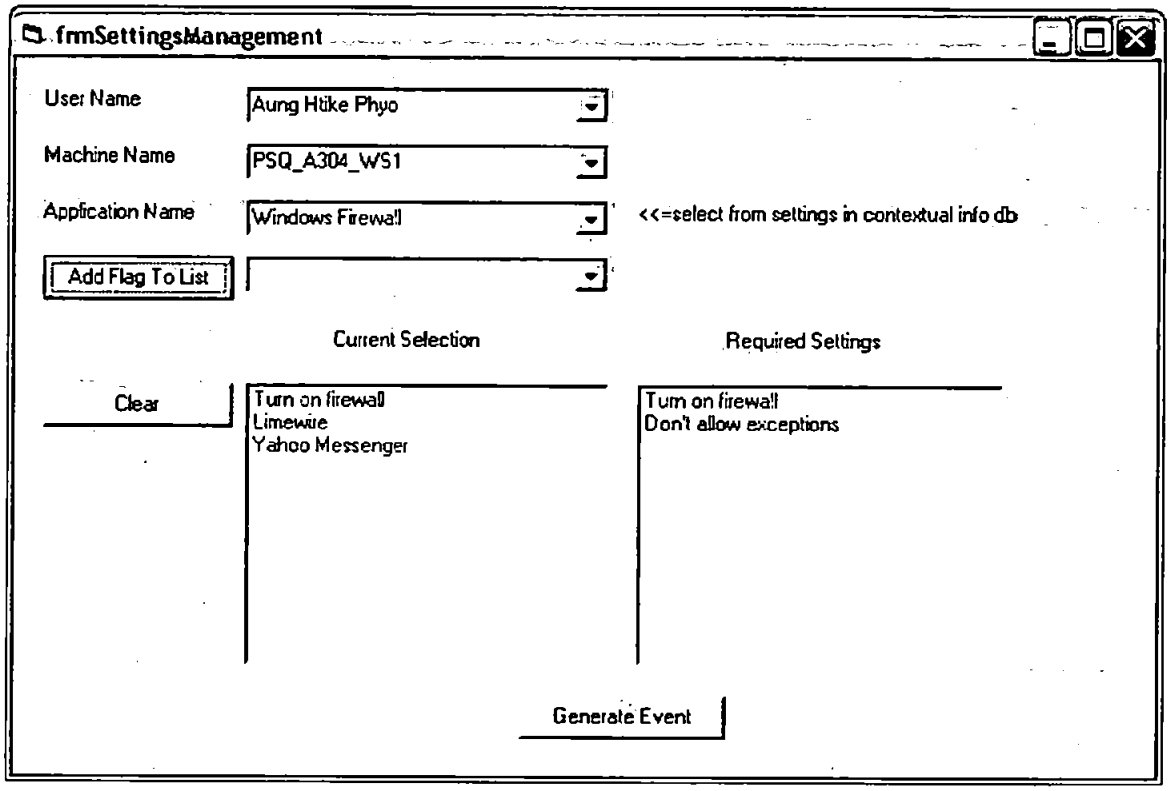


Figure 8.8 Configuration Change Log Generator Interface

UserName: This attribute is used to store the user responsible for the activity.

MachineName: This attribute is used to store the machine affected. It is assumed that the user also used the same machine for carrying out the activity.

ApplicationName: This attribute is used to store the application affected.

8.3.10 Flags

The *Flags* table is used in conjunction with the Settings table, in order to store flags associated with each event. The *Flags* table contain two attributes, *EventID*, and *FlagDescription*.

EventID: This attribute indicates the event to which the associated flag belongs. There may be more than one flag associated with each event.

FlagDescription: This attribute is used to store the description of the flag used for the event. This attribute stores only one flag for each record. Thus if there are more than one flag associated with a unique event, each associated flag will be paired with the same *EventID*.

8.3.11 Data Transfers

The *Data Transfers* table is used for logging data regarding the transfer of files/data through network/communication applications with file transfer capability. The *DataTransfers* table contains eight attributes, *EventID*, *SenderName*, *SenderMachineName*, *ApplicationName*, *CommunicationServer*, *FileID*, *SenderAddress*, and *ReceiverAddress*.

SenderName: This attribute is used to store the name of the user sending/transferring the file.

SenderMachineName: This attribute is used to store the name of the machine the user transferred the file from.

ApplicationName: This attribute is used to store the name of the application utilised for transferring/sending the file.

CommunicationServer: This attribute is used to store the IP address of the server involved in mediating the transfer of the file.

FileID: This attribute is used to store the identifier of the file transferred.

SenderAddress: This attribute is used to store the communication address of the sender. For the purpose of the demo, email address is used.

ReceiverAddress: This attribute is used to store the communication address of the receiving user. For the purpose of the demonstration, email address is used.

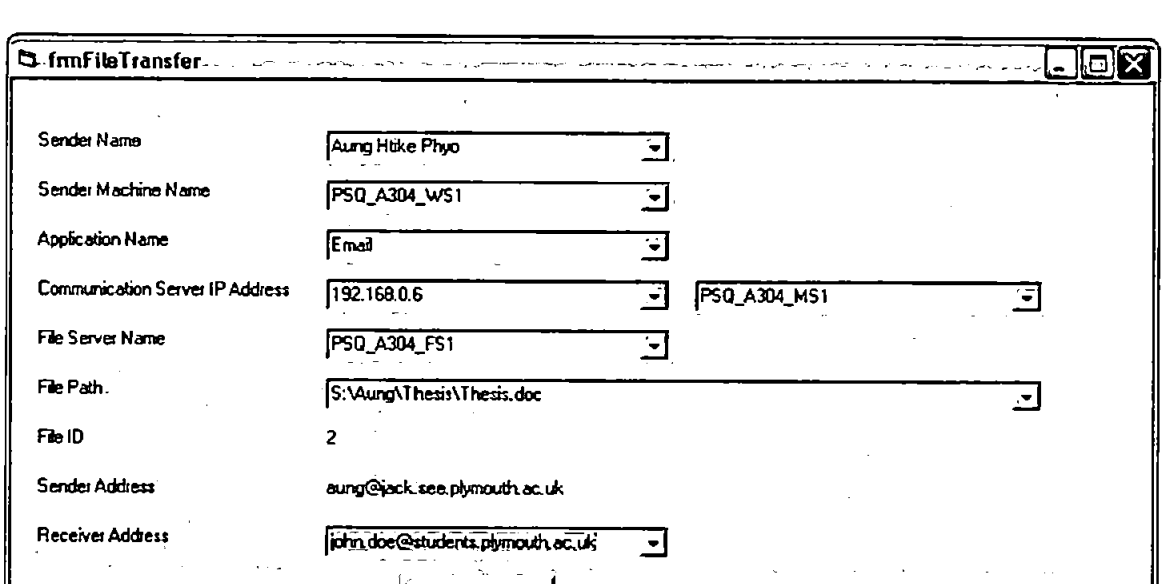


Figure 8.9 File Transfer Log Generator Interface

8.3.12 Data Retrievals

The *Data Retrievals* table is used in conjunction with the *DataTransfers* table for logging related data regarding retrieval of transferred file/data. The *DataRetrievals* table contains four attributes, *EventID*, *ReferenceEventID*, *ApplicationName*, and *ReceiverMachine*.

ReferenceEventID: This attribute is used to store the *EventID* of the data transfer event that appears in the *DataTransfers* table, so that the two events can be linked.

ApplicationName: This attribute is used to store the name of the application the receiving user utilised for retrieval of the file/data. Note: This may not be available if the receiving user was accessing from an external machine.

ReceiverMachine: This attribute is used to store the IP address of the machine from which the data was retrieved.

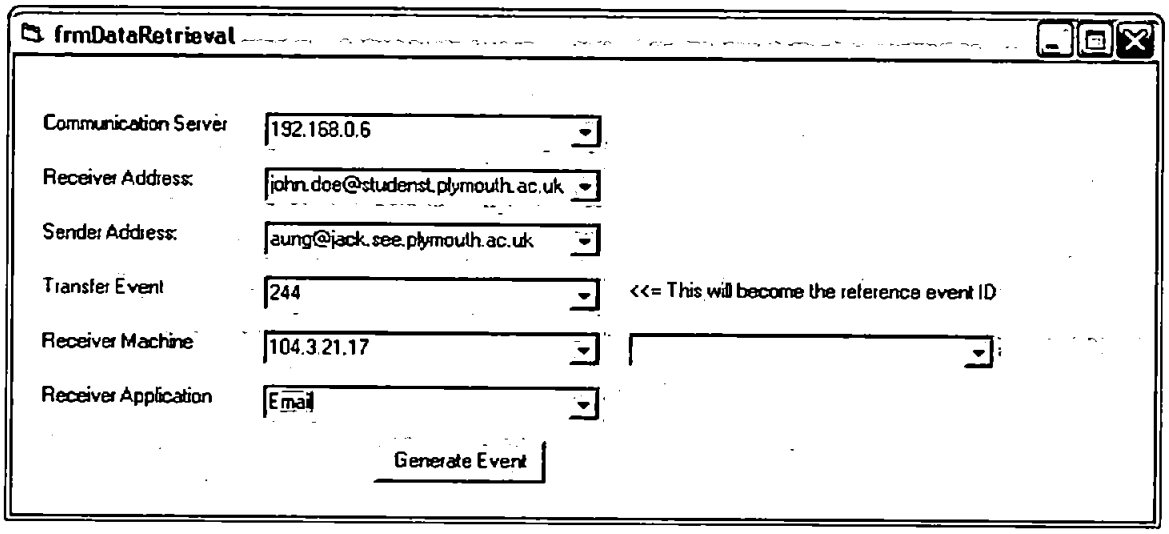


Figure 8.10 Data Retrieval Log Generator Interface

8.3.13 Add Query

The *Add Query* table is used for adding database queries that need to be verified. The *Add Query* table contains four attributes, *UserName*, *MachineName*, *QueryName*, and *QueryType*. This data table is used as a reference by the analysis engine. Thus addition of new entries to this data table needs to be verified by the monitoring tool administrator and the custodian of the database affected by the query.

QueryName: This attribute is used for storing the name/identification of the query.

QueryType: This attribute is used for describing the type of the query. Currently there are only two available query types, a general *Query*, and *Registry Management* queries.

In order to accommodate misfeasor monitoring, the operating system and applications must be able to provide the parameters needed for analysis of activity.

8.4 Knowledge Database

Knowledge Database is utilised by the *Analysis Engine* and *Alert Generator* as a reference in order to determine whether the user activity may be deemed misuse in the context defined by the application and the organisation, and to provide information regarding the appropriate authority to be alerted for each suspicious activity. *Knowledge Database* contains thirteen data tables, *Employees*, *File Inventory*, *Roles*, *UserRoles*, *FileAllowedRoles*, *FileAllowedEmployees*, *Machines* (Considered as Internal), *Settings*, *ServerAllowedMachines*, *Queries*, *QueryVerificationReference*, *Registries*, and *RegistryCustodians*.

8.4.1 Employees table

This table contains list of employees of the organisation, in the context of the demonstration, each record represents an insider. For future developments of the system, it may also include contractors, and customers, or devise a way of adding data tables

containing details of perceived insider. The *Employees* table contains six attributes, *EmployeeID*, *FullName*, *Department*, *ImmediateSuperior*, *JobTitle*, and *EmailAddress*.

EmployeeID: This is an automatically generated number, used to represent each unique record.

FullName: This attribute stores the full name of the employee.

Department: This attribute stores the department to which the employee belongs.

ImmediateSuperior: This attribute stores the name of the person responsible for supervision of the employee within the organisation. This is used to present knowledge of hierarchy to the monitoring system. For example, if the person transferring sensitive information to someone outside the organisation happens to be the custodian of the information, the immediate superior may be informed of the activity as a precautionary measure. *Immediate Superior* may also become the temporary custodian of the data managed by the employee in the event of redundancy.

JobTitle: This attribute stores the job title of the employee for descriptive purpose.

EmailAddress: This attribute stores the email address of the employee. In the context of the prototype system, this attribute is used to determine whether file transfer/communication is between insiders or insider-outsider.

8.4.2 File Inventory

It may not be practical to include all files for misfeasor monitoring. The purpose of the *File Inventory* table within the *Knowledge database* is used for listing files that require misfeasor monitoring. The *File Inventory* table contains nine attributes, *FileID*, *MachineName*, *FilePath*, *FileCustodian*, *FileDescription*, *ApplicationForAccess*, *PartialReplicationAllowed*, *WholeReplicationAllowed*, and *SaveToRemovableDisk*.

FileID: The value of this attribute is unique, and automatically generated by the database. The number associated represents each file listed, i.e. the unique combination of the machine on which the file is located (*Machine Name*), and the exact file path (*File Path*).

MachineName: This attribute stores the name of the server machine on which the listed file is stored.

FilePath: This attribute stores the exact file path of the listed file on the server machine.

FileCustodian: This attribute stores the name of the person who understands the sensitivity of the contents and responsible for the security of the file listed. In the context of the prototype system, the alerts related to the file will be sent to the associated file custodian.

FileDescription: This attribute contains a short description of the file.

ApplicationForAccess: This attribute contains the name of the application defined as the norm for access by majority of the users. In the context of the prototype system, the value is used as a reference for determining whether the user is attempting access to the file through an arbitrary application.

PartialReplicationAllowed: This is a Boolean value, and this attribute indicates whether partial replication of the file's contents is acceptable.

WholeReplicationAllowed: This is a Boolean value, and this attribute indicates whether the replication of the file is acceptable.

SaveToRemovableDisk: This is a Boolean value, and this attribute indicates whether replicating the file to a removable disk is acceptable.

8.4.3 Roles table

The purpose of the *Roles* table is to group uses with similar responsible responsibilities, in order to simplify management of user access rights. The *Roles* table consists of five attributes, *RoleID*, *RoleName*, *Department*, *RoleManager*, and *RoleDescription*.

RoleID: This attribute is the primary key and contains automatically generated number to represent each unique record.

RoleName: This attribute stores the given name of the role.

Department: This attribute stores the name of the department to which the role belongs. Currently, the prototype system does not utilise this value in the decision making process.

RoleManager: This attribute stores the name of the personnel responsible for deciding which users should belong to the role, and managing the access rights required for the role. In the context of the prototype system, this is the person responsible for verifying the addition of new users to the role. If the role manager adds a user to the role under his management, then the custodian(s) of the file(s) accessible by the role should be informed.

RoleDescription: This attribute stores a short description of the role.

8.4.4 UserRoles table

UserRoles table list the users and the role they are associated with.

RecordID: This is an automatically generated number to represent each unique record.

UserName: This attribute stores the name of the user.

RoleName: This attribute stores the name of the role the user belongs to.

8.4.5 FileAllowedRoles table

This table list the roles and the files allowed access to the role members. *FileAllowedRoles* contains two attributes *FileID*, and *RoleName*.

FileID: This is the number representing the record in the Files table, from which the machine the file is located on and the exact path to the file can be derived.

RoleName: This attribute stores the name of the role allowed to access the file.

8.4.6 FileAllowedEmployees table

This table lists individual users who may not belong to the role allowed access to a particular file, yet may still be allowed access to the file on individual basis. This table contains two attributes, *FileID*, and *FullName*.

FileID: This is the number representing the record in the *File Inventory* table, from which the machine the file is located on and the file path can be derived.

FullName: This is the name of the individual allowed access to the file.

Note: *FileAllowedRoles* and *FileAllowedEmployees* are used to determine the acceptability of data dissemination and do not represent access control policies at system or application level i.e. the users defined here may not actually have the access rights at system/application level to read or write the contents, but may receive if a user who has access transfers the file/contents.

8.4.7 Machines table

The *Machines* table list the details of organisation's internal machines. It contains seven attributes, *MachineID*, *MachineName*, *MachineType*, *SubnetDescription*, *IPAddress*, *SystemAdministrator*, and *MachineDescription*.

MachineID: This attribute is an automatically generated number to represent each machine. In future developments of the system it may be more practical to utilise this value for identifying machines, although the current prototype system uses the machine name for identification.

MachineName: This attribute stores the given name of the machine. In the prototype system a naming convention is used, so that the machine's physical location can be derived from the machine name. The ability to derive the physical location of a machine is important if the acceptable usage policy include the physical location of a machine, and the physical security of each location is considered for analysis.

MachineType: This attribute stores the machine type, i.e. File Server, Database Server, Mail Server, Web Server, Print Server, User Workstation, etc. Current prototype system does not utilise this value during decision-making process.

SubnetDescription: This attribute stores a short description of the sub-network the machine belongs to.

IPAddress: This attribute stores the IP address assigned to the machine. In the context of the prototype system, the values of this field are used for reference in

determining whether communication occurs between machines managed by the organisation or those outside the organisation.

SystemAdministrator: This attribute stores the name of the person responsible for the security and proper functioning of the machine. In the context of the prototype system, alerts related to the security and functionality of the system are sent to the administrator associated with the machine. (If formal segregation of duties are defined, the alerts should be sent to the System Security Officer, and not the system administrator.)

MachineDescription: This attribute stores a short description of the machine's purpose, such as backup server, mail server, etc.

8.4.8 Settings table

The *Settings* table list the required settings of each application on each machine. The *Settings* table contain three attributes, *MachineName*, *ApplicationName*, and *FlagDescription*. Within this table a compound key (*MachineName* & *ApplicationName*) is used, so that the same application on different machines can have different settings.

MachineName: This attribute stores the name of the machine the policy is associated with.

ApplicationName: This attribute stores the name of the application the policy is associated with.

FlagDescription: This attribute stores the flag/option, required for proper functioning and security of the associated machine/application. This attribute accommodates only one option/flag each. Therefore, if an application requires more than one flag/option, multiple records need to be created in this table.

Within the prototype system, flags are associated with machine-application combination. In the future developments a compound key (*Machine ID*, and *Application ID*), may be created in a separate table, and flags may then be associated with the compound key.

8.4.9 ServerAllowedMachines table

ServerAllowedMachines table list the pairing of server and the machine allowed to access the associated server. This table contains two attributes, *ServerName*, and *MachineName*. In the context of the prototype system, this table is referenced to determine whether the machine utilised by the recipient of a file transfer is also allowed to access the server from which the file originated, in order to detect indirect bypass of controls.

ServerName: This attribute stores the name of the server that may be accessed by the associated machine.

MachineName: This attribute stores the name of the machine allowed to access the associated server.

Note: The associated machine may not actually have direct access to the file server, but have adequate security to receive the file(s) through a file transfer activity. Therefore, the pairing here does not represent firewall rules.

8.4.10 Database Access

In order to accommodate monitoring of database access, the monitoring system needs to identify databases that require monitoring, and the associated reference data needed for decision making during analysis.

Within the prototype system, two monitoring approaches are available for database access. One approach is for monitoring databases that are considered part of registries, and the other is for general-purpose databases. The aim of registry monitoring is to identify the appropriate authority to be alerted so as to accommodate a form of segregation, whereas the aim of general-purpose database monitoring is to ensure integrity of the records by way of counter verification according to the data integrity policy.

The prototype system identifies database that require monitoring through the use of *Queries* Table, and *Registries* Table, and determines the associated reference required for

decision-making through the use of *Query Verification Reference* Table, and *Registry Custodian* Table. This approach allows the monitoring system to utilise data from dispersed databases for decision-making during analysis. The code responsible for performing this function within the prototype system corresponds to the *Fact Processor* component of the conceptual system.

8.4.11 Queries table

This table list the queries that need to be verified for integrity and validity of access. This table contains eight attributes, *QueryName*, *ServerName*, *FilePath*, *TableName*, *TableCustodian*, *PrimaryKeyAttribute*, *ForeignKeyAttribute*, and *AttributeToVerify*.

QueryName: This attribute stores the name of the user queries that need to be verified.

ServerName: This attribute stores the name of the server on which the database file is stored.

FilePath: This attribute stores the path to the database file on the server.

TableName: This attribute stores the name of the data table affected by the query.

TableCustodian: This attribute stores the name of the person responsible for the security of the data table.

PrimaryKeyAttribute: This attribute stores the unique identifier of each record in the concerned data table.

ForeignKeyAttribute: This attribute stores the foreign key, i.e. the name of the attribute containing the value that should also appear in the reference record needed for verification, so that the relevant reference record can be identified and linked with the affected record.

AttributeToVerify: This attribute stores the name of the attribute in the affected data table that needs to be verified. In situations when the verification only needs to check whether a reference record exists or merely need to check the Boolean value within the referenced record, i.e. the values within this record do not need to be compared against a value contained within the referenced record; this attribute may store the foreign key value.

In future developments, database maybe used instead of file path, and since a query may affect more than one data table, the table name attribute may be dropped, and creating a separate table in which the affected table names are associated with each query.

8.4.12 QueryVerificationReference table

QueryVerificationReference table list the data tables required for reference by the data verification process. This table contains seven attributes, *QueryName*, *ServerName*, *FilePath*, *TableName*, *ForeignKeyAttribute*, *AttributeToVerify*, and *Condition*.

QueryName: This attribute stores the name of the query for which reference is associated with. The query name in this attribute corresponds to that stored in the Queries table.

ServerName: This attribute stores the name of the server on which the database containing the data required for reference is stored.

FilePath: This attribute stores the path to the database file that contains the data required for reference is stored.

TableName: This attribute stores the name of the data table containing the data required for reference.

ForeignKeyAttribute: This attribute stores the name of the attribute that contains the value also appearing in the record affected, so that the affected record and the relevant reference record can be associated.

AttributeToVerify: This attribute stores the name of the attribute that needs to be tested against a specified attribute of the record affected by the query.

Condition: This attribute stores the condition that must be satisfied in order to maintain integrity and security of the affected record. Within the prototype system, one of three possible conditions (Exists, Be True, Be Equal) can be used. In future implementations, if additional conditions need to be determined such as whether a value within a record is greater than a value within the referenced record, the procedure for determining the condition can be implemented as a *Fact Processor* and Boolean (True or False) value of the fact can be returned.

8.4.13 Adding New Queries

Database queries that require monitoring (i.e. affected records require counter verification against another record) can be added through the event generator.

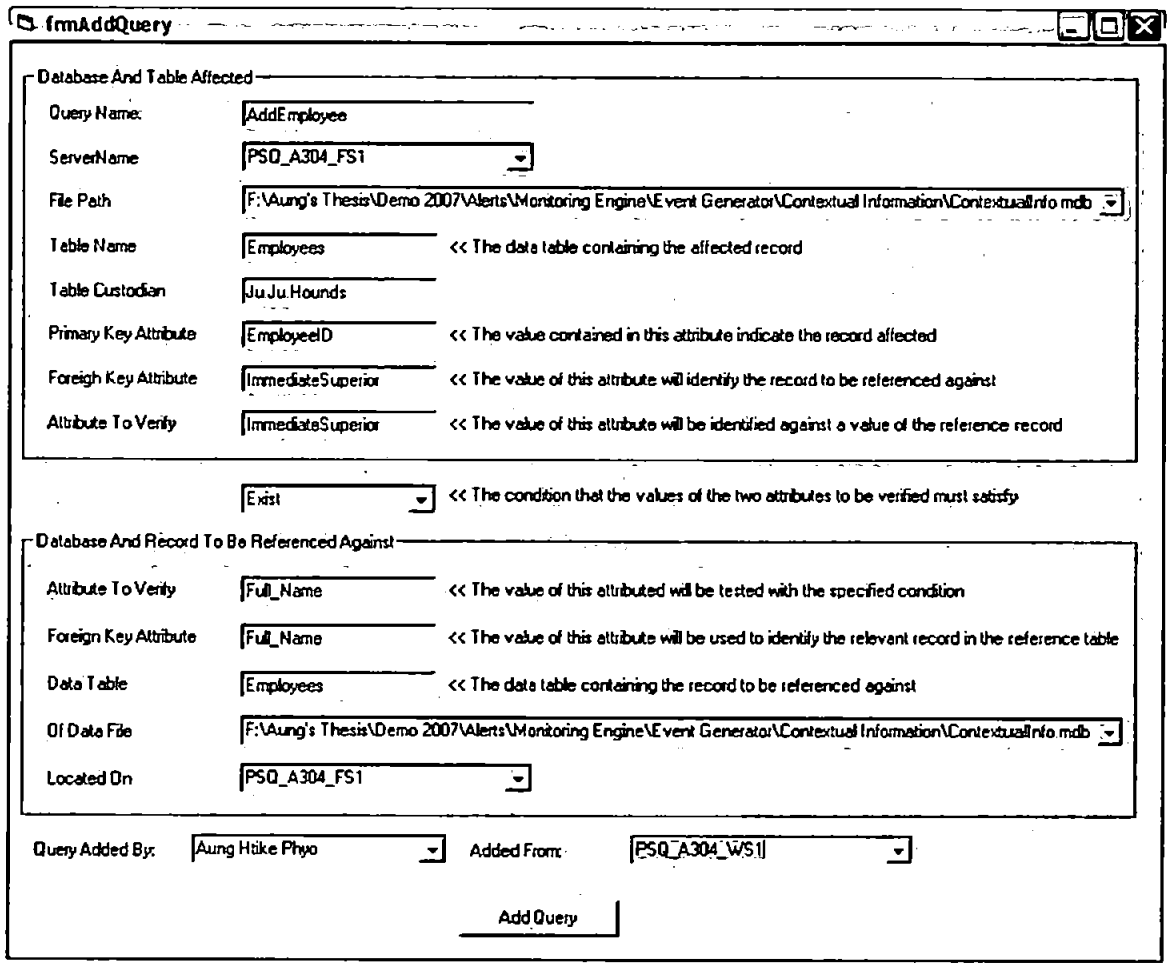


Figure 8.11 Interface for Adding Queries to be Monitored

8.4.14 Registries table

Registries table contains the name and details of the registries (such as employees, students, users, etc.), to which if a new entity is added; the event should be notified to the person responsible for the integrity of the registry. This table contains eight attributes *RegistryName, ServerName, FilePath, TableName, PrimaryKeyAttribute, RecordNameAttribute, AttributeName, RegistryDescription.*

RegistryName: This attribute stores the given name of the registry.

ServerName: This attribute stores the name of the server on which the affected database is stored.

FilePath: This attribute stores the file path to the database affected.

TableName: This attribute stores the name of the data table the registry is stored in.

PrimaryKeyAttribute: This attribute stores the name of the attribute in which the unique identifier of each record is stored.

RecordNameAttribute: This attribute stores the name of the attribute in which the associated with the affected record is stored.

ForeignKeyAttribute: This attribute stores the name of the attribute containing a value that will be used to identify the person responsible for the integrity of the record or the affected registry.

RegistryDescription: This attribute stores a short description of the list.

8.4.15 RegistryCustodians table

RegistryCustodians table contains the reference records that will identify the custodian responsible for ensuring the integrity of the records added to the *Registries* table. *RegistryCustodians* table contain six attributes, *RegistryName*, *ServerName*, FilePath, *TableName*, *ForeignKey*, and *CustodianAttribute*.

RegistryName: This attribute stores the name of the registry. This value corresponds to the value stored in the *RegistryName* of the *Registries* data table.

ServerName: This attribute stores the name of the server on which the database containing the custodian of the registry.

FilePath: This attribute stores the path to the database file containing the custodian of the registry.

TableName: This attribute stores the name of the data table in which the custodian of the registry is stored.

ForeignKeyAttribute: This attribute stores the name of the attribute containing the same value as the foreign key in the affected record.

CustodianAttribute: This attribute stores the name of the custodian responsible for ensuring integrity of the registry.

8.4.16 Adding New Registries

New registries that require monitoring (i.e. the addition of new entries to the registry need to be verified by an appropriate authority) may be added through the event generator.

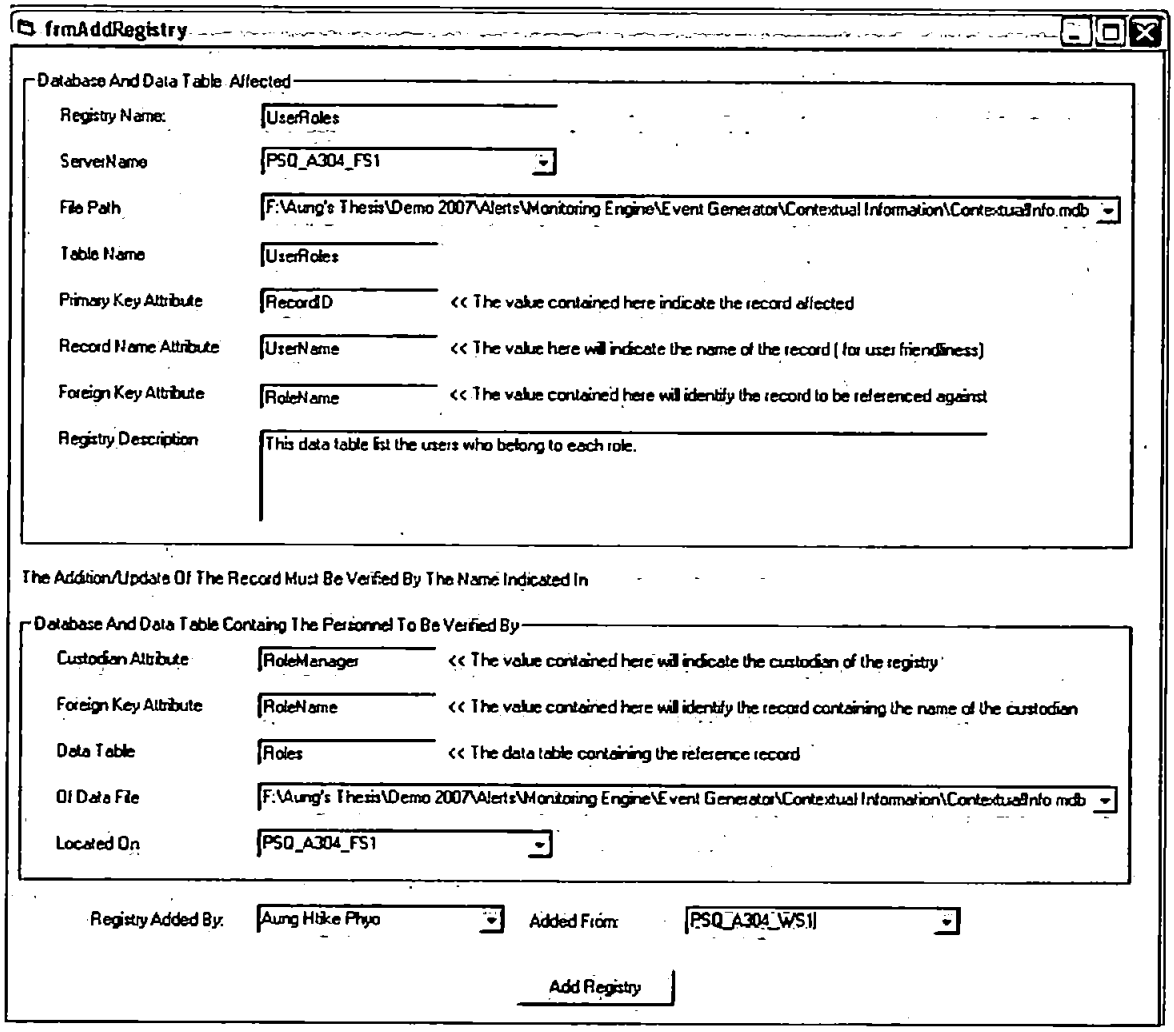


Figure 8.12 Interface for Adding New Registries to be Monitored

8.5 Event Analysis Engine

Event Analysis Engine works in the background, and does not have visible interface. It examines the *Events*, refers to relevant information from the *Knowledge Database*, and makes decisions according to the logic described in the previous chapter. In the prototype system, each event type is associated with a decision-making logic each. Upon detecting a suspicious activity, the event is logged into the alerts database. The *Alert Generator* checks the *Alerts Database* and refers to the *Knowledge Database* in order to alert the appropriate authority through suitable alert interface.

At the moment, reasoning procedure in the prototype system does not branch out once the activity has been identified, and thus sufficient with sequential programming. For example, NIDS needs to use expert system because reasoning can branch out depending upon the protocol used. For example, an IP packet may contain (ICMP, TCP, UDP) and many other protocols. In addition, each protocol also contains a number of fields, the values of which may vary, and the variations may require a different analysis approach.

Event Analysis Engine is able to successfully detect misfeasance activities listed in the introduction section of this chapter. Test data and results are included in the Appendix A. The activities analysed by the *Event Analysis Engine* are only a subset of activities listed in Chapter 5. Activities that can only be detected through statistical analysis are not included, because characterisation of normal behaviour through statistical analysis was not carried out as part of the research.

8.6 Alert Generator

Alert generator checks the entries in the alerts database, and provides detailed alert information to the appropriate authorities such as system administrator, file custodian, table custodian, or registry custodian.

The main alerts interface list all alerts within the alerts database, indicating the *Event ID*, the *perpetrator name*, and the *misuse type*. The administrator can view the details of each alert by clicking on the alert entry in the list. In addition this interface also provides statistics of the alerts, such as total number of alerts in the database, the number of alerts for selected misuse type, the number of alerts for selected user, the number of alerts for selected type by selected user, and percentages.

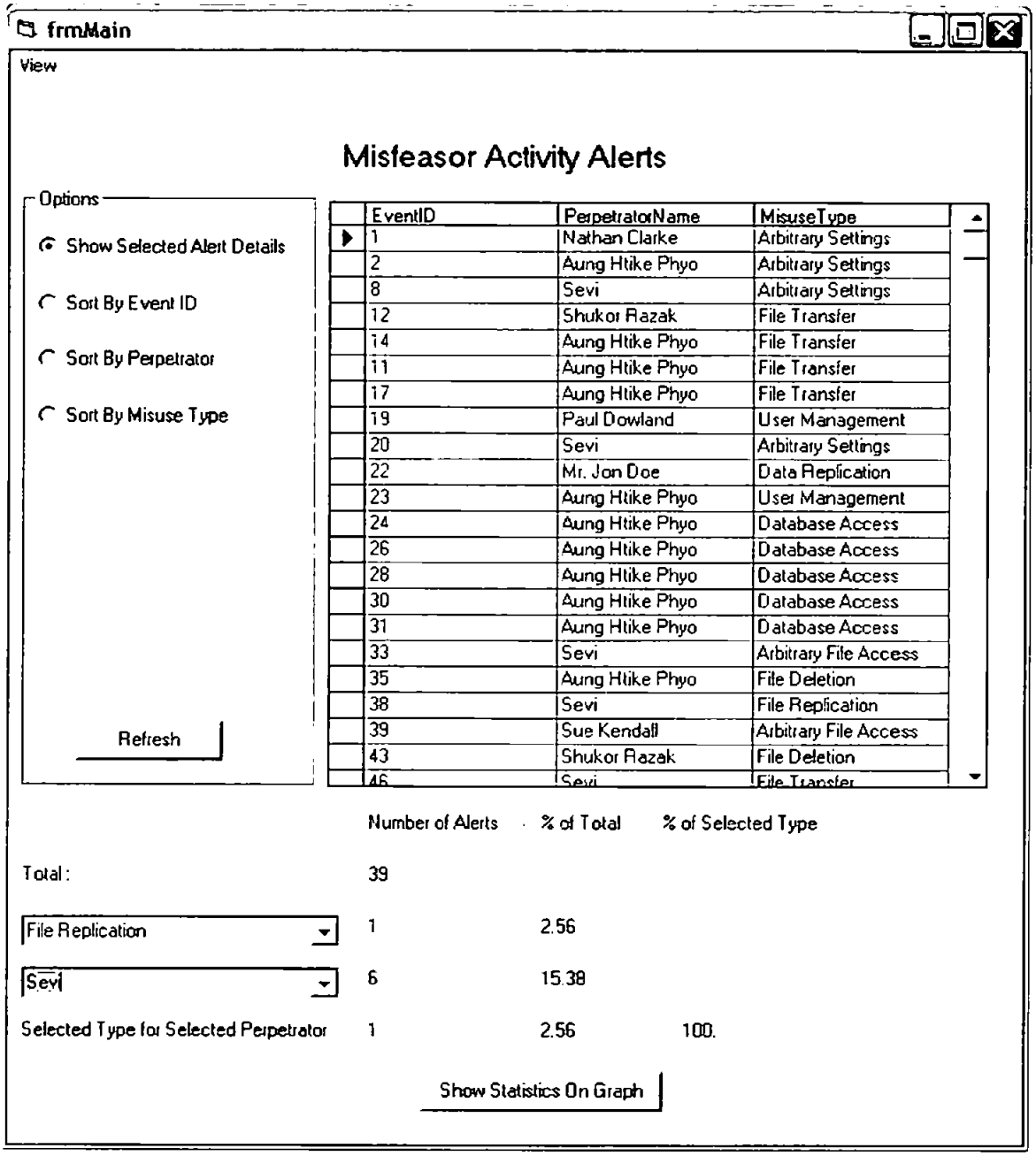


Figure 8.13 Misfeasor Activity Alerts Main Interface

The administrator can also view new queries/registries that have been added to be monitored by clicking on View, then Query/Registry additions.

8.6.1 Arbitrary File Access Alert

If the selected alert to view is an alert for arbitrary file access, the details of the event will be shown through the *Arbitrary File Access Alert interface*.

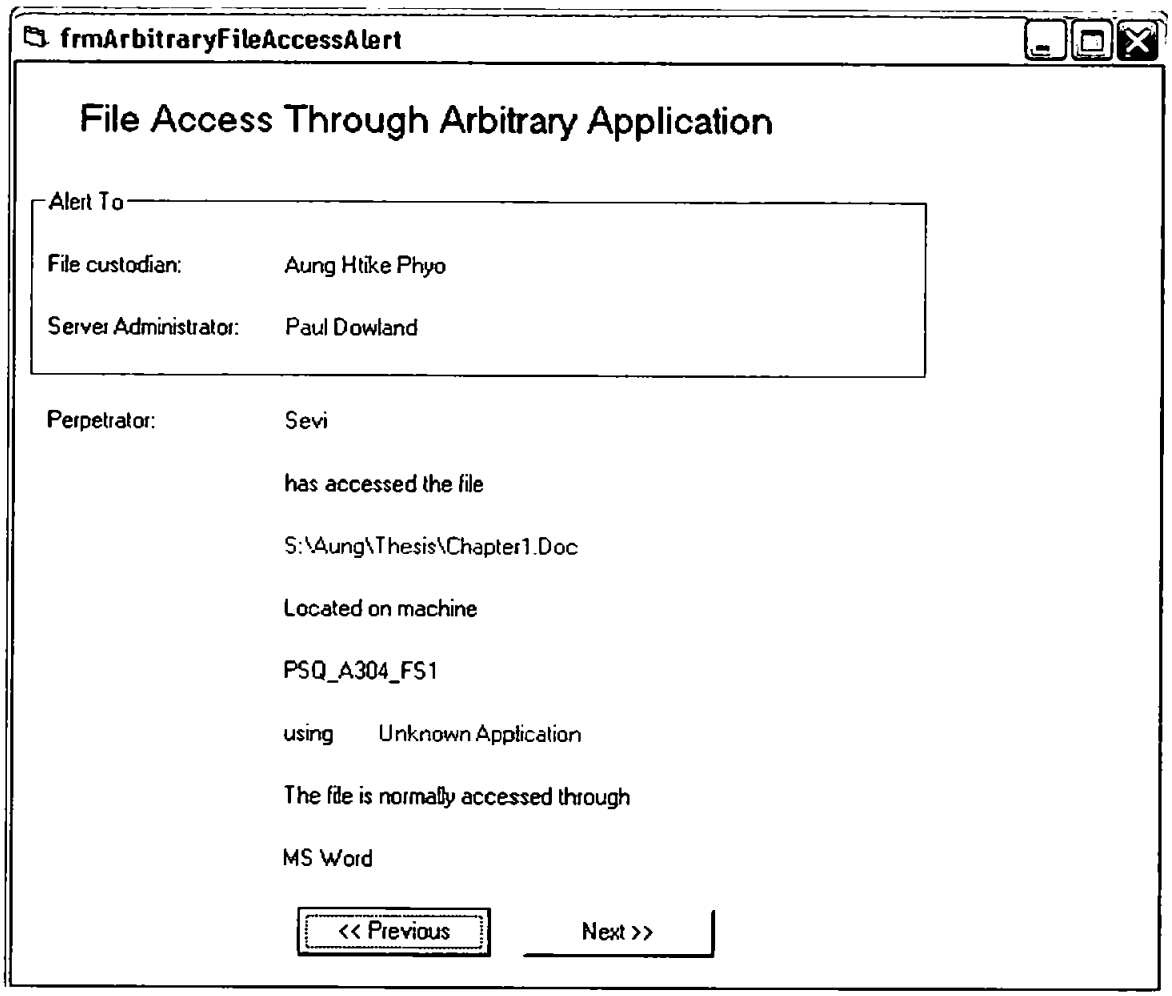


Figure 8.14 Arbitrary File Access Alert Interface

In order to accommodate monitoring user activity within an application environment, the application the user utilised for accessing the file must be able to provide necessary log

data. If the user utilised an application that does not provide necessary log data, it will not be possible to determine the user activity within the application environment. Therefore, as a first point of monitoring, the system determines whether the application utilised by the user to access the file matches the one defined as the norm. If the application utilised for accessing the file did not match the one defined as norm, the custodian of the file, and the administrator of the file server is alerted with the details. The details include the name of the perpetrator, the name of the file server, the file affected, the application utilised, and the application defined as norm.

Case Scenario:

A user accesses the document file through the web browser application in order to upload it onto the web, or to send it through web based mail application.

8.6.2 File Deletion Alert

Some of the monitored files may contain highly valuable content such as product designs, blue prints, and source codes. Therefore, deletions of monitored files need to be alerted to the file custodian, and file server administrator. The alerts of this type are presented through the *File Deletion Alert interface*. The details of the alert include, the perpetrator name, file server, affected file, and file description. The user can also view the details of the file by clicking of the "View File Details" button.

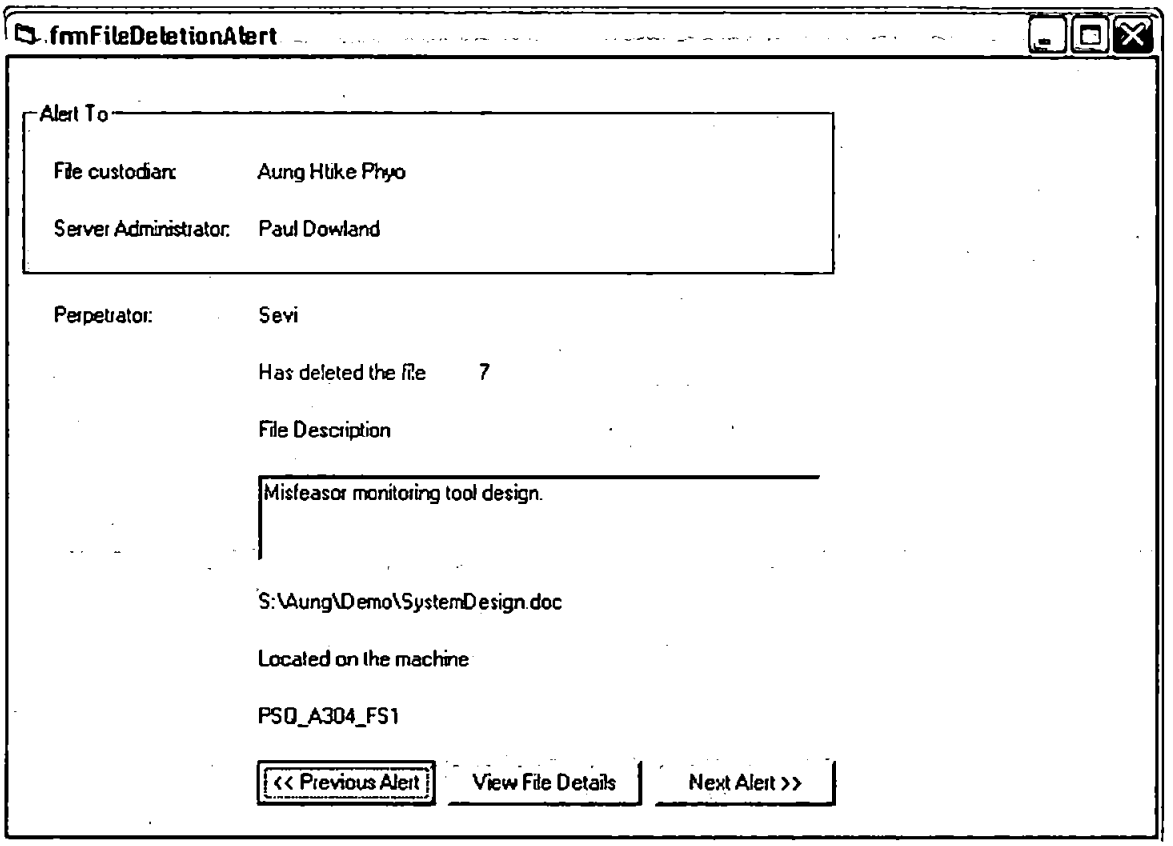


Figure 8.15 File Deletion Alert Interface

In the event of a critical file being deleted; the user may wish to determine whether a replication of the file exists within the network in order to attempt recovery. This information is available within the *File Details interface*. The details provided by the *File Details interface* include:

File ID: used for identification within the registry of monitored files

File Custodian: responsible for security of the file, and the person to be alerted

File Description: for quick identification of contents

Application for normal access: application acceptable for accessing file

Partial replication of contents: acceptable usage policy

Replication of entire file: acceptable usage policy

Saving file to a removable media: acceptable usage policy

Server administrator: administrator of the file server

Server: the machine on which the file is/was located

File path: the exact file path on the file server

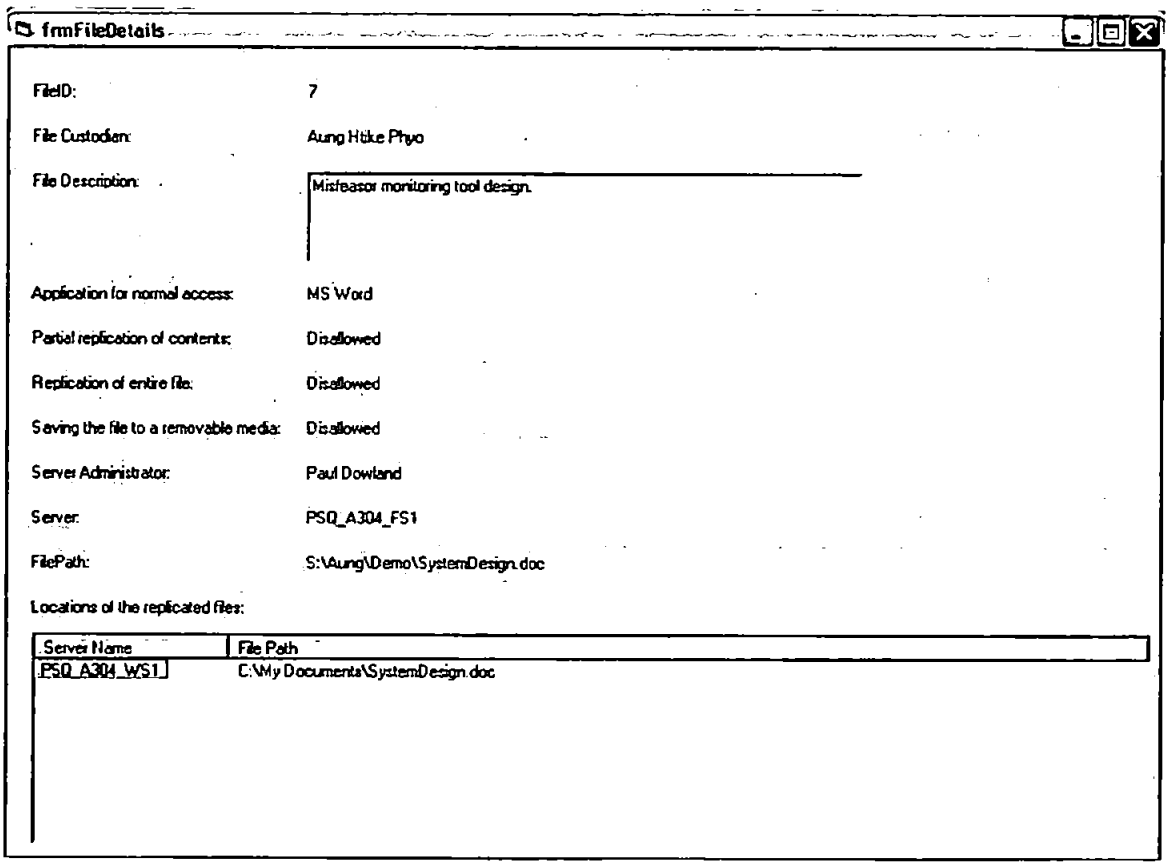


Figure 8.16 File Details Interface

In addition the interface also lists the replicated copies of the file within the network. In order to assist recovery/shredding process, it provides the name of the machine the replica is located, and the exact file path.

8.6.3 File Replication Alert

Some of the monitored files may contain information regarded, as intellectual property of the organisation, and replication of such files need to be monitored. The decision-making logic used for the demonstration is:

1. If the file security policy states that replication of the file to a removable media is acceptable, no alerts will be generated and replication will not be logged.
2. If the file security policy states, that replication of the file to a removable media is not acceptable, but replication of the file to local hard drives or network drives is acceptable, then the activity will be logged, but not alerted.
3. If the file security policy states, that replication of the file to a removable media is not acceptable, and replication of the file to local hard drives or network drives is not acceptable, then the activity will be alerted to the file custodian, and the administrator of the file server.

The details of the file security policy can be viewed by clicking on the "View File Details" button.

The details provided by the *File Replication Alert* include, file custodian, server administrator, perpetrator name, file server, file path, application utilised, destination machine name, and destination file path.

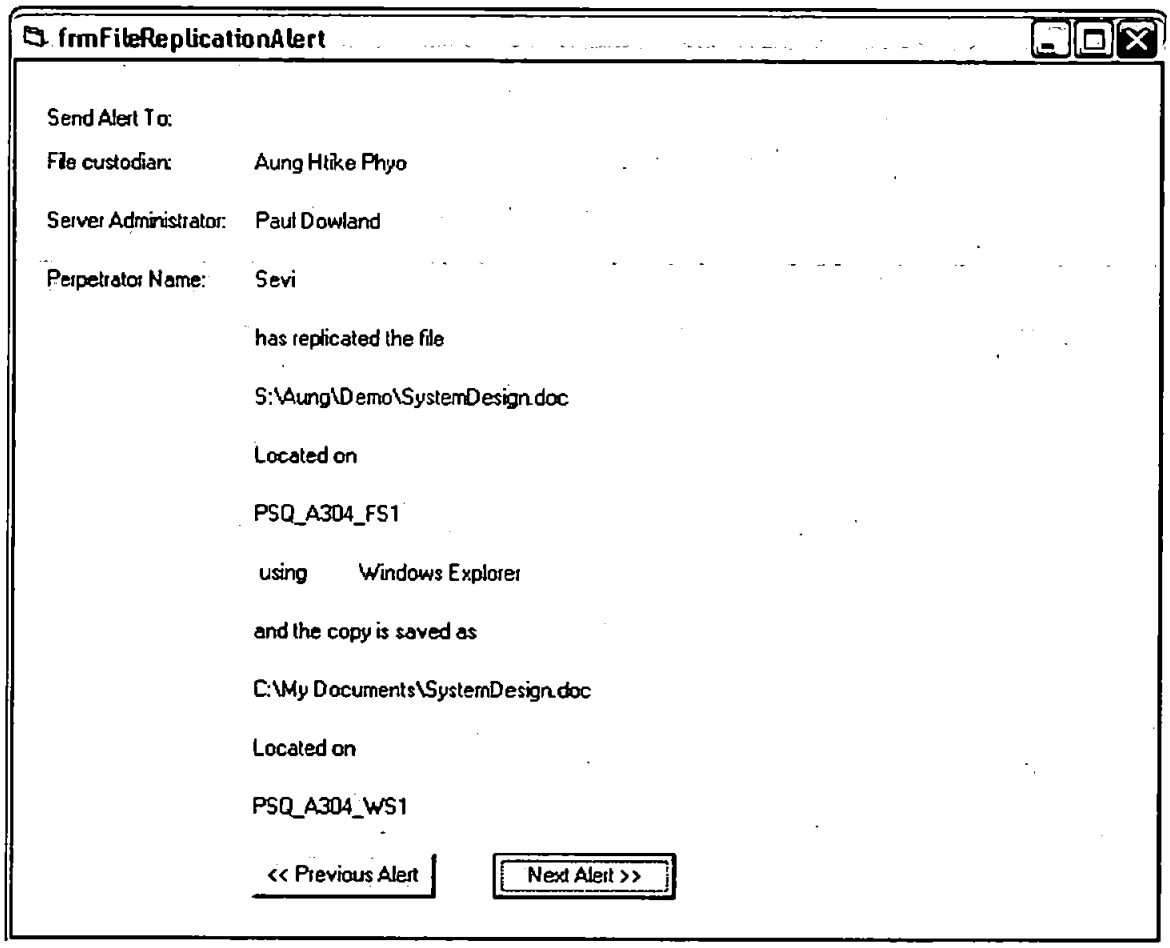


Figure 8.17 File Replication Alert Interface

8.6.4 Partial Content Replication Alert

Some of the monitored files may contain highly sensitive information, such as a summary of market analysis, customer survey, etc. In some scenarios the users may copy (a small

or large percentage of) the content through applications that accommodate direct access to the contents of the file. Depending upon the sensitivity of the content, such activity may result in compromise the confidentiality of the information. Therefore, partial replications of file contents need to be analysed to determine whether the activity is acceptable. The decision-making logic utilised in the prototype system is:

1. If the file's acceptable usage policy state that the replication of the file to a removable media, a local hard disk, or a network drive is acceptable then partial replication of content should be acceptable, and thus analysis ends without alerting anyone.

2. If the file's acceptable usage policy states that the replication of the file to a removable media, a local hard disk, a network drive, or partial replication of content is not acceptable, then the file custodian and the server administrator are alerted of the activity.

The acceptable usage policy of the file can be vied by clicking on the "View File Details" button.

The details provided by the *Partial Content Replication Alert* include, file custodian, server administrator, perpetrator name, file server, file path, application utilised, destination machine name, and destination file path.

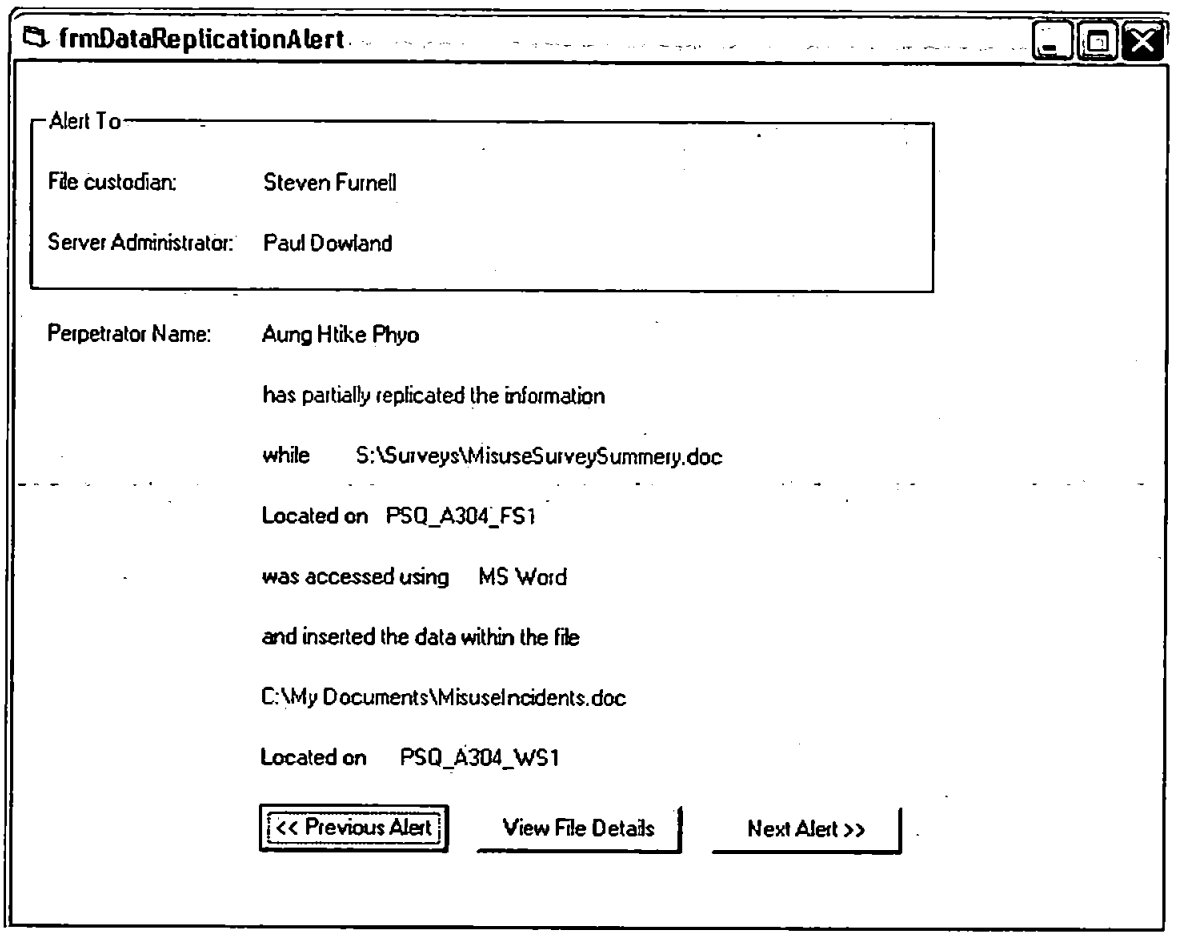


Figure 8.18 Content Replication Alert Interface

8.6.5 File Transfer Alert

In some scenarios, controls may be indirectly bypassed by transferring monitored files through networked applications. Therefore, transfer of files through networked applications need to be monitored. The decision-making logic utilised by the demonstration is:

1. If the file's acceptable usage policy states that saving it to a removable media is acceptable then, further analysis is not necessary, and no alerts need to be generated.

2. If the file's acceptable usage policy states that saving it to a removable media is not acceptable then, the further analysis is made.

3. Is the servers mediating the communication is not managed by the organisation then the activity is alerted. The reasoning is that the file is effectively leaving the organisation's managed systems.

4. If the server mediating the communication is managed by the organisation. Is the recipient an employee of the organisation?

5. If the recipient is not an employee of the organisation, the activity is alerted.

6. If the recipient is an employee, but not authorised to access the file (derived from file allowed roles/users data tables), then the activity is alerted.

7. If the recipient is an employee and allowed access to the file, but the machine utilised for retrieving the mail is not allowed access to the file server the hosting the source copy of the attached file, then the activity is alerted.

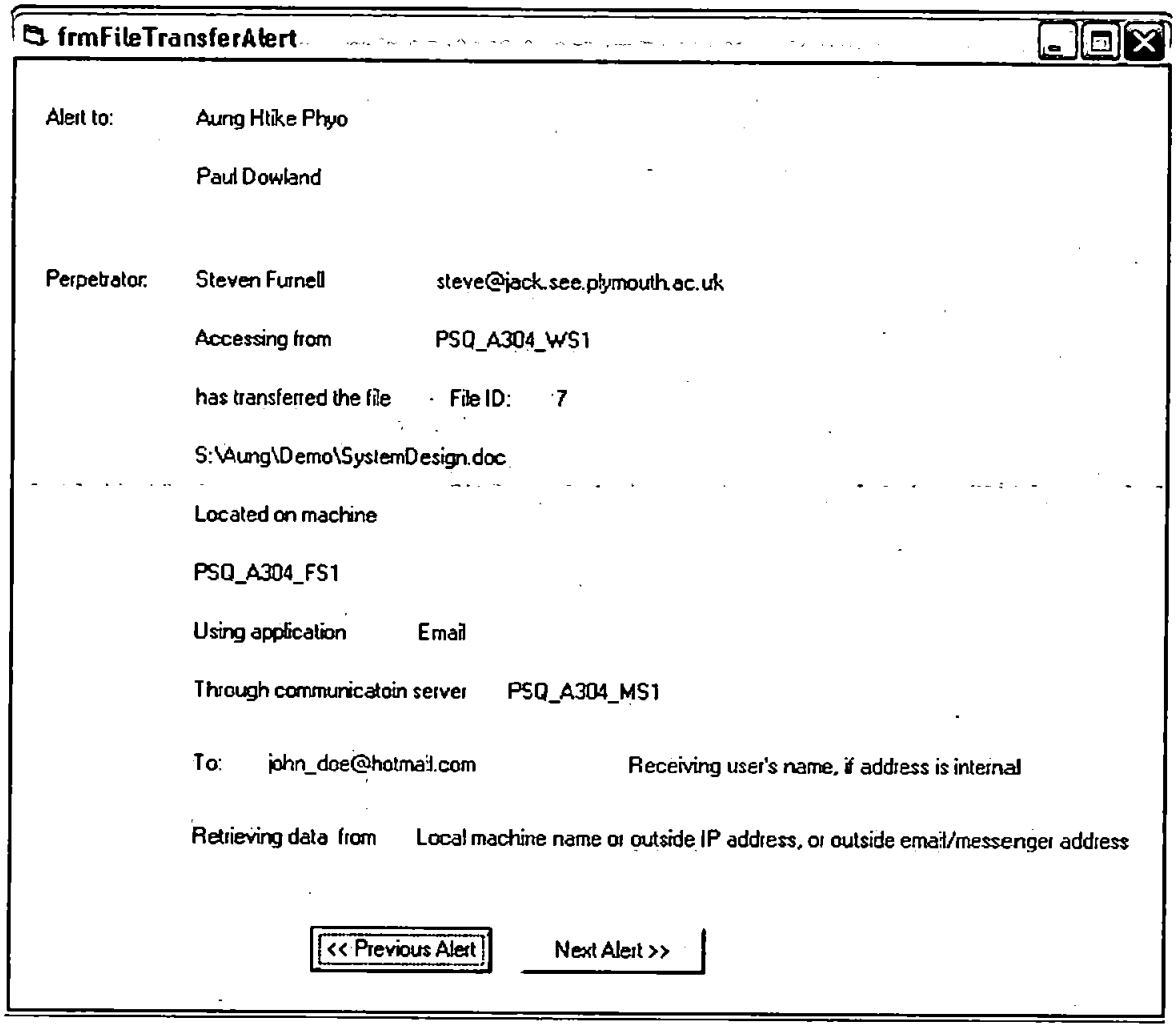


Figure 8.19 File Transfer Alert Interface

Case Scenario 1:

A user attaches the file with an email sent through a mail server that is not managed by the organisation. This would require reconfiguration of the mail client by the user prior to sending the mail, which should also be detected by configuration changes monitoring. The prototype system detects activities of this nature.

Case Scenario 2:

For example, a user who is allowed access to the file attaches the file with an email and sends it to someone who is not an employee of the organisation. The prototype system detects activities of this nature.

The current implementation of the prototype system determines whether the recipient is an employee of the organisation by checking the email address associated with each employee. In future implementations domain name checking may also be employed, i.e. if the receiving mail server is not managed by the organisation then the file has effectively left the organisation.

Case Scenario 3:

A user mails the file to a colleague, who is not authorised to access the file. In this case, although the recipient is an employee of the organisation, the recipient is not authorised to access the file, and thus is a violation of security policy. The prototype system detects activities of this nature.

Case Scenario 4:

A user mails the file to a colleague, who is also authorised to access the file. The recipient retrieves the file from outside the organisation's network. In this case, although the recipient is authorised to access the file, by retrieving the file from a machine outside the

organisation's network, the file's security will not be monitored or controlled by the organisation's systems. The prototype system detects activities of this nature.

Case Scenario 5:

A user mails the file to a colleague, who is also authorised to access the file. The recipient retrieves the file from a machine within the organisation's network. However, the machine utilised for retrieving the file is not allowed to access the file server hosting the source copy of the attached file. In this case, the machine may not be allowed access to the file server because it does not have a monitoring agent installed, or located in a less secure physical location, which may ultimately result in the compromise of the file's security. The prototype system detects activities of this nature.

8.6.6 Database Access Alert

Database abuse such as accessing a greater percentage of records compared to the rest of the users, or users issuing a query that has wide search criteria will be easily noticed by a competent database administrator, and database management systems already include such monitoring features. However, a single query that may constitute misuse within the context of business controls/process may not be easily noticed, and thus require automated counter verification approach for each query issued within critical databases.

8.6.7 Registry Access Alert

Within organisations registries play an important role. Human resource registry provides the list of employees belonging to the organisation, and thus may receive salary. Customer registry provides the list of individuals eligible for the services provided by the organisation. A computer user registry provides the list of users authorised to utilise the organisation's computers. A database user registry provides the list of users authorised to access the database system. A criminal registry provides the list of individuals consider as a danger to the society, and thus to be treated with suspicion and contempt. Therefore, belonging to a registry has its rewards or punishments. Thus addition of new records to each of organisation's registries needs to be verified by the appropriate authority. This requires the monitoring system to identify which registry has been affected, and to determine the authority to be alerted for verification. The prototype system provides this feature, and the alerts are presented through the *Registry Access Alert* Interface.

8.6.7.1 Identifying appropriate authority for verification

Case Scenario 1: As a starting point, it would be most appropriate to verify the addition of new records to the employee database of any organisation, since this database can be used as a reference for monitoring other activities (computer user account creation, creation of payroll account, etc). In any organisation, each and every employee reports to someone in a supervisory position. Therefore, involving the person in supervisory position for verification would be appropriate, and create a segregation to detect the abuse of the ability to add new records to the employee database. The figure illustrates,

when user "Aung Htike Phyo" adds a new record "Daniel Rosenberg" to the Employees registry with the user "Bruce Lee" as assigned supervisor.

Case Scenario 2: For example, if access rights and privileges are associated with user roles/groups, the addition of a user to a role/group will grant that user with the access rights of the role. A role manager will certainly have the knowledge of which users should belong to the role. This will aid in detecting privilege misuse by system technicians, and create a verification/authorisation scenario to prevent abuse of trust.

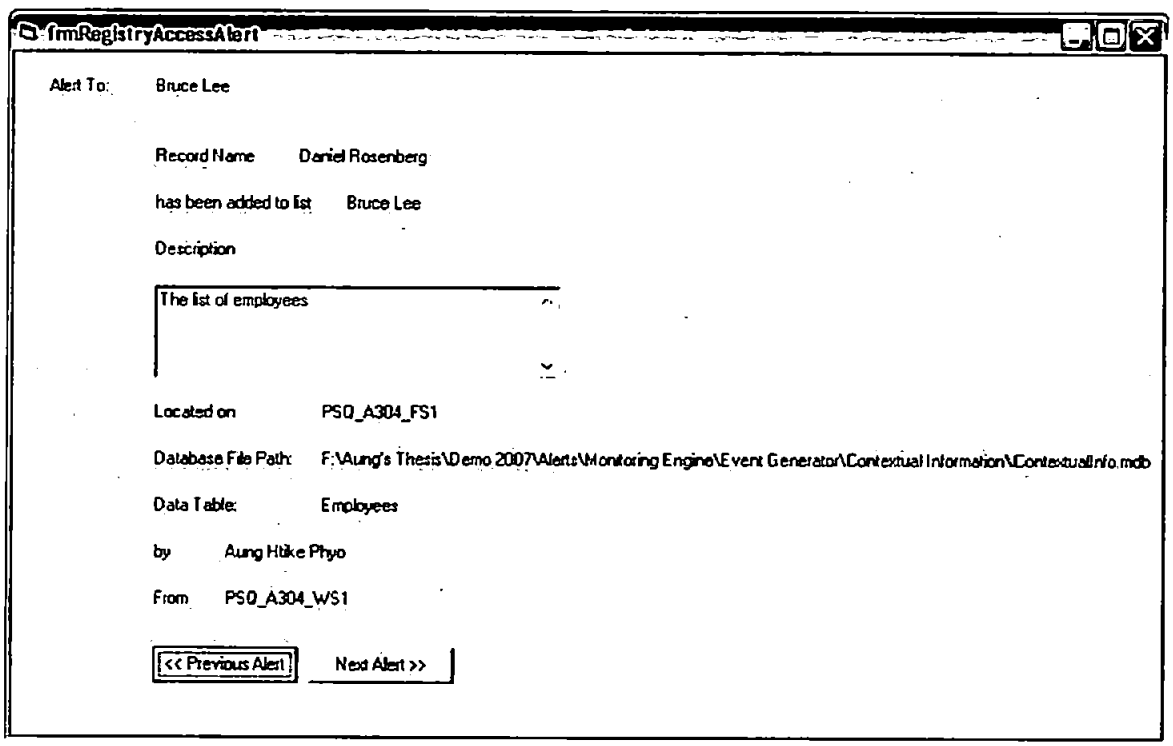


Figure 8.20 Registry Modification Alert Interface

Case Scenario 3: Upon addition of an employee to the payroll database, the manager of the department the employee belongs to can be involved for verification process. The department attribute of the affected record can be referenced against the department manager attribute within the human resource database (or any other database) to determine the appropriate person to be involved for verification.

8.6.7.2 Ensuring existence

For example, in order to detect creation of ghost accounts (payroll-account, user-account); when a user account is created or a user is added to a system, verification can be made to ensure that the account creation or user addition is legitimate. In order do accommodate automated verification a reference is needed. The verification may need to reference one of the organisation's registry (employees, contractors, customers, etc) database, and the monitoring rule may state that the added entity must exit in the referenced registry, and if the rule is not satisfied, the activity can be alerted to the information systems security officer and respond accordingly.

Case Scenario 1: A system administrator creates a ghost user account for a non-existent employee. The system detects this by referring to the employee registry.

This type of rule can certainly be implemented in to custom applications as application level control. However, such capability may not be available in commercial off the shelf applications, and indeed it may not be practical to include such controls because each

organisation's registry location/structure/purpose can be different to that of another, and may not be easily tied to the application. In addition it may be desirable to enable similar verifications for various applications, and to add such functionality to each application may require large overheads.

8.6.7.3 Status check

Case Scenario 1: In order to detect creation of illegal system user accounts i.e. addition of new records to the system user database, the verification status of the employee's record within the employee registry can be referenced.

Case Scenario 2: In order to detect creation of ghost accounts within payroll database, the verification status of the employee's record within the organisation's employee registry can be referenced.

Case Scenario 3: For each successful login session, the status (actively employed) of the individual within the organisation's employee registry may also be checked in addition to identification and authentication mechanism of the accessed system.

8.6.7.4 Ensuring equality

Case Scenario: In order to detect possible fraud, upon creation/update of an employee's payroll record, the agreed salary for the given employee within the human resource database can be referenced for verification.

Case Scenario 1: Difference of total deposits and total withdrawals must equal to account balance.

Although it is possible to include such verification as part of access control, it is not practical to include such many factors within the access control system, as it would complicate application functionality and contribute to compatibility issues in future developments of application. In addition, if application level controls also exist this type of counter verification ensures that the application functions as expected, and if there were anomalies, such as the salami case problem in which the application code itself is affected, the monitoring tool would be able to detect.

IDS systems are a counter verification technology to ensure that software/users behave in a particular manner, despite the existence of access control mechanisms. IDS systems exist because software/users can behave in ways unexpected at the time of the system design.

8.6.8 Arbitrary Settings Alert

In order to ensure proper functionality of the applications (be it security related or service related), it is essential that the application be correctly configured. Therefore, verification is required to ensure that the system/application configurations are exactly as desired. In order to accommodate verification, the desired settings for each application on each machine are stored in the *Settings* table of the *Contextual Information* database for

reference. If the settings made by a user do not match the settings defined, the activity is logged and the administrator of the affected system is alerted. The prototype-monitoring tool provides this type of alerts through the Arbitrary Settings Alert interface.

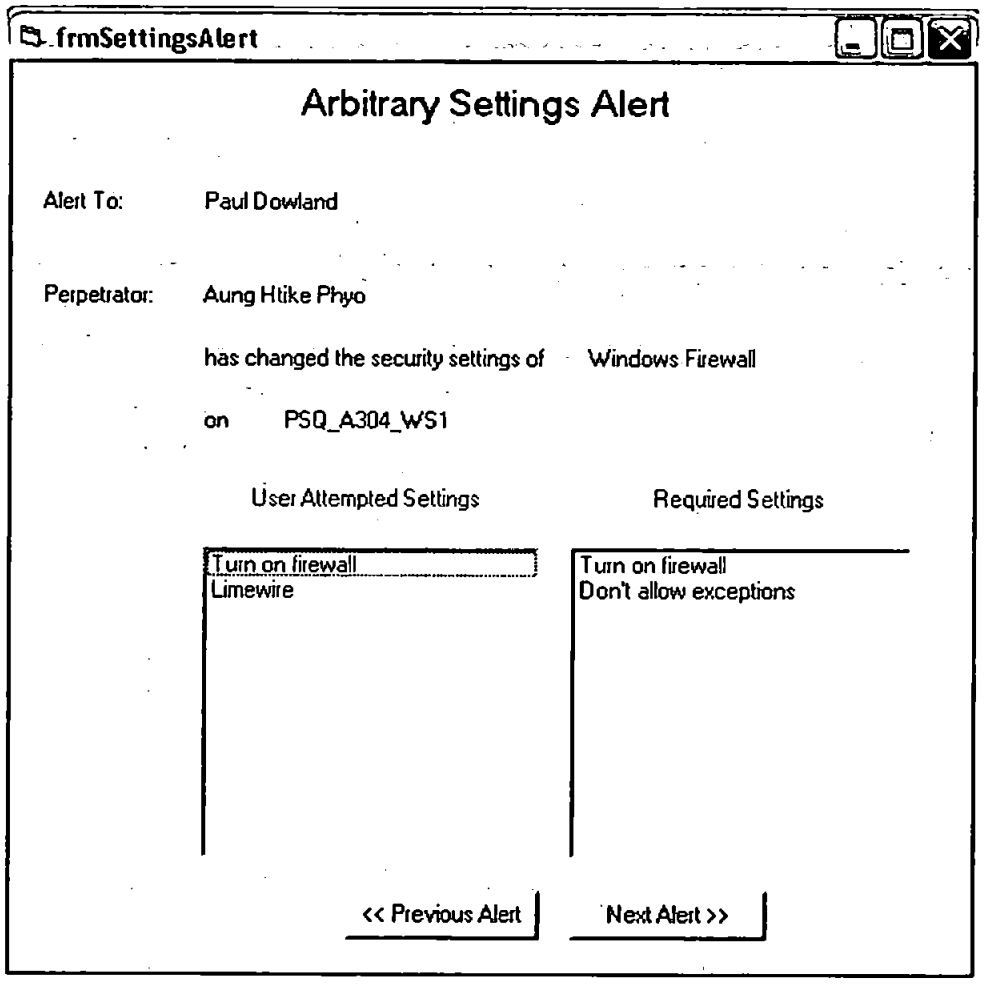


Figure 8.21 Arbitrary Settings Alert Interface

Case Scenario 1: Turning off virus scanner, or changing the settings of anti-virus software can result in organisation's machines being infected with malicious code.

Case Scenario 2: Changing the firewall settings, and allowing unauthorised listening services, or allowing unauthorised applications to access the Internet.

Case Scenario 3: Changing the settings of back up application, so that the data is backed up to a different machine than desired, resulting in compromise of data security.

8.7 Conclusions

A prototype misfeasor monitoring system developed according to the architecture described in Chapter 7 is presented here. The prototype is evaluated against a number of activities that may result in misfeasance within a generic organisation. The prototype has been successfully validated against a number of activities selected for evaluation, including:

- File Access through an arbitrary application
- Deletion of files considered as property of the organisation
- Replication of confidential files
- Replication of confidential content
- Dissemination of confidential files
- Modification of database records
- Addition of users or new records to organisation's registry databases
- Modification of system/application settings

The evaluation validates the theory that existing detection technologies and strategies can indeed be employed for detection of misfeasance. The prototype has utilised inferential analysis, and a knowledgebase containing specification of acceptable norms related to each activity in order to successfully detect potential misfeasance. The prototype also validates the claim made in Chapter 5 that the successful detection of misfeasance depends upon the availability of data relevant for analysis of misfeasance. The prototype demonstrated that relevant audit data from appropriate level of the system (Network, OS, Application, and Data), and contextual information relating to each parameter analysed are needed for successful detection. Through the use of the prototype system to detect a number of scenarios, it has also demonstrated that in some cases (such as dissemination of data, and verification of database access), the data required for analysis may need to be gathered from more than one level of the system.

The prototype system has validated the conceptual architecture presented in Chapter 7, including the analysis procedures developed for contextual analysis of user activities. It validates the fact that the activity needs to be identified in order to determine appropriate analysis procedure, consequently validating the relevance of *Event Identifier* [Figure 7.2] in the conceptual architecture. The prototype has also shown that the audit data associated with each activity needs to be classified, which validates the relevance of Data *Classifiers* [Figure 7.2.] in the conceptual architecture. The prototype has also shown that contextual information from dispersed databases can be extracted for misfeasance detection and alert

distribution; validating the relevance of *Fact Processors*, and *Alert Generator* components [Figure 7.2] of the conceptual architecture. The prototype has demonstrated that alerts can be dynamically distributed (by extracting information from organisation's live databases) to the appropriate authority for verification of the activity, which validates the relevance of the *Alert Generator* component [Figure 7.2] of the conceptual architecture. The prototype system has also demonstrated the fact that some of the data representing contextual conditions may only be available from organisation's production databases.

Chapter 9 Conclusions

9 Conclusions

This chapter concludes the thesis by presenting a summary of the achievements, and the limitations of the research. It also presents the improvements that can be made and the future work that can be based upon the work carried out.

9.1 Achievements of the research

The research has achieved all of the objectives specified in Chapter 1, with introduction of new conceptual architecture and practical work carried out in a number of areas in order to validate the theory and concepts. The specific objectives achieved are:

1. Limitations of access control mechanisms with regards to insider misuse (especially dissemination of data and sabotage through deletion of files) have been identified, and possible improvements have been suggested (Chapter 3). Access controls require that the mechanism be embedded within the environment in which the activity is regulated, and thus pose a limitation.

2. Limitations of traditional IDS design with regards to detecting misfeasance have been identified (i.e. most of the data available for analysis by traditional IDS is not particularly relevant for detection of misfeasor activities). The reasons why misfeasance activities cannot be detected by traditional IDS was established (Chapter 4). Consequently, requirements that need to be met for successful detection of misfeasance have been identified.

3. A taxonomy for identifying the appropriate level within the IT system where relevant data for misfeasor analysis can be collected has been developed (Chapter 5). Relevant data needed for analysis of each type of misfeasance has been identified.

4. The applications, operations, and data that are likely to be misused have been identified. A checklist for identifying applications, operations, and data that is most likely to be misused by insiders has been established (Chapter 6).

5. A conceptual architecture for monitoring misfeasor activities has been designed; facilitating the use of contextual data only available from dispersed databases, and highlighting where existing detection technologies fit within the architecture. Appropriate inference algorithms have been developed for monitoring each type of misfeasor activity (Chapter 7).

6. A prototype misfeasor monitoring system has been developed incorporating novel use of existing detection technologies and strategies, and tested against a number of misfeasance scenarios prove the validity of the concept (Chapter 8), and how misfeasance can be detected. The prototype proves that existing IDS technologies and strategies can be applied to misfeasor monitoring if relevant data for analysis is available and suitable analysis procedures are developed.

A number of papers relating to each part of the research have been presented at refereed conferences and journals, (the published papers are attached in Appendix) and have received encouraging comments from delegates and reviewers. Thus it is believed that the research has made valid and useful contributions to the field of IT security, intrusion detection, and misfeasance detection in particular.

9.2 Limitations of the research

Despite achieving the overall objectives outlined at the start, the limitations associated with the work needs to be explained, so that improvements can be made in the future. The main limitations of the research are:

1. Data collection components are developed only for Active Window Monitor, and Bandwidth Usage Monitor. The data used for validating the prototype system was generated artificially. The monitoring system relies upon the operating system and the application vendors to include features for collection of relevant log data. In that sense, the monitoring system cannot be considered complete. Difficulties and compatibility issues may arise when vendors attempt to implement new Application Programming Interfaces within the operating system.

2. The prototype uses higher-level information, which requires a number of stages to process and correlate lower-level log data in to facts that can actually be used for

inference. Collection and correlation of lower-level logs are not carried out as part of the research.

3. Regarding statistical analysis, the characterisation of norms is not carried out as part of the research i.e. the process for establishing normal profiles of user/system behaviour was omitted. As a consequence, detection based upon statistical analysis and historical profiling was not conducted.

4. Database records made accessible to the monitoring system may become subject of inference attacks. Safeguarding of log data or contextual data accessed by the monitoring system was not considered as part of the research.

9.3 Suggestions and scope for future work

On the basis of the discoveries made from the research, it is possible to identify a number of areas in which future work can be carried out to build upon the work undertaken. A number of ideas have been suggested in parts of the previous chapters. New ideas in addition to those mentioned previously are outlined here.

1. One of the areas that could not be solved by monitoring is the abuse of privileges by system administrators. Therefore, system administrative operations that requires segregation and system components that can be separated needs to be

identified in order to enforce segregation of duties for administrative users, so that the consequences of sabotage by system masters can be minimised.

2. It has been identified that some of the contextual data may only be available from dispersed databases of the organisation. Therefore, middleware components that will accommodate the monitoring system to easily extract the data from relevant databases need to be developed.

3. From the discussions within the thesis, it can be noted that the ability to include relevant parameters enables existing detection technologies to be applied in a new context. Therefore, future work should focus upon features that will allow users to create inference rules that consider new parameters relevant to the context in which the activity is analysed.

4. Inference rules need to be developed to represent contextual conditions governing acceptable usage of legitimate user activities. However, contextual conditions vary from one organisation to another. In addition, such conditions may only be understood by business managers. Therefore, user friendly interfaces/process need to be developed, so that little technical knowledge is required to create inference rules and fact identifiers to define the acceptable conditions governing the access of database access.

9.4 The future of misfeasor monitoring

Organisations across various disciplines are becoming increasingly reliant upon IT systems for the proper functioning of their businesses. Therefore, advance security countermeasures need to be developed to ensure the integrity and availability of the systems while maintaining confidentiality of data. Intrusion detection systems have been widely employed within many IT environments. However, traditional IDS were designed to detect attacks and misuses usually employed by those who do not have legitimate system level access, and they were not designed to detect misuse of legitimate access. Nonetheless, authorised users with legitimate access may misuse granted privileges, and the consequences of privilege misuse can be severe as IT dependency increases. Although, acceptable usage monitoring systems are available on the market, they focus mainly on the monitoring of Internet access by employees. As have been highlighted throughout the research, in addition to Internet access, various activities such as dissemination of data, configuration of systems, creation/management of accounts, and database access can result in misfeasance. Thus future IDS need to include features to analyse such activities in order to determine whether the operation is acceptable within the context in which it occurred.

References

1. Aleph One (1996), "Smashing the Stack for Fun and Profit", Phrack online journal, Vol. 7, Issue 49, 8 November 1996

2. Almgren, M. and Lindqvist, U. (2001), "Application-Integrated Data Collection for Security Monitoring", Lecture Notes in Computer Science, Vol. 2212, pp.22

3. Amoroso, E. (1999), "Intrusion Detection: An Introduction to Internet Surveillance, Correlation, Traps, Trace Back, and Response", Second Printing, Intrusion.Net Books, New Jersey, June 1999

4. Anderson, J.P. (1980), "Computer Security Threat Monitoring and Surveillance", Technical Report, James P Anderson Co., Fort Washington, April 1980.

5. Arena, K. (2001), "Hanssen Pleads Guilty to Spying for Moscow", CNN.com, http://archives.cnn.com/2001/LAW/07/06/hanssen/index.html , 8 July, 2001

6. Asiaweek (1995), "Billion-Dollar Man", Asiaweek.com, 29 December, 1995

7. Audit Commission (1991), "Survey of Computer Fraud & Abuse: Supplement", Audit Commission, January 1991

8. Audit Commission (1994), "Opportunity Makes a Thief: An Analysis of Computer Abuse", HMSO Publications Centre, ISBN 0-11-886137-9, 1994

9. Audit Commission (2001), "Yourbusiness@risk: An update on IT abuse 2001", Audit Commission Publications, ISBN 1 86240 289 2, September 2001.

10. Audit Commission (2005), "ICT Fraud and Abuse 2004 – An Update to yourbusiness@risk", Audit Commission Publications, U.K. June 2005

11. Axelsson, S. (2000), "Intrusion Detection Systems: A Survey and Taxonomy", Chalmers University, March 2000, http://citeseer.ist.psu.edu/axelsson00intrusion.html

12. Balasubramaniyan, J.S. Garcia-Fernandez, J.O. Isacoff, D. Spafford, E. and Zamboni, D. (1998), "An Architecture for Intrusion Detection using Autonomous Agents", ACSAC, pp.13-24, 1998

13. Barbara, D. Couto, J. Jajodia, S. Popyack, L. and Wu, N. (2001), "ADAM: Detecting Intrusions by Data Mining", In the Proceedings of the 2001 IEEE Workshop on Information Assurance and Security, United States Military Academy, West Point, NY, 5-6 June 2001.

14. BBC (2007a), "Ferrari spy still at the team", bbc.co.uk, 8th July 2007

15. BBC (2007b), "Facebook cost businesses dear", bbc.co.uk, 11th September 2007

16. Bejtlich, R. (2005), "Extrusion Detection: Security Monitoring of Internal Intursions", Addison Wesley, ISBN-10:0321349962, November 2005

17. Bell, D.E. and LaPadula, L.J. (1975), "Secure computer systems: Unified exposition and Multics interpretation", technical report ESD-TR-75-306, The Mitre Corporation, Bedford, MA, March 1975

18. Biermann, E. Cloete, E. and Venter, L.M. (2001), "A Comparison of Intrusion Detection Systems", Computers & Security, Vol.20, No.8, pp.676-683

19. Briney, A. Prince, F. (2002), "ISM Survey 2002", Information Security Magazine. http://infosecuritymag.techtarget.com/2002/sep/2002survey.pdf, September, 2002.

20. Brackney, R.C. and Anderson, R. H. (2004), "Understanding the Insider Threat", In the Proceedings of a March 2004 Workshop", RAND National Security Research Division, ISBN 0-8330-3680-7 , 2004

21. Carr, S. (2005), "Desk skiving popular with UK workers", sillicon.com, 13 April, 2005

22. Cappelli, D. Moore, A. Shimeall, T. and Trzeciak, R. (2006), "Common Sense Guide to Prevention and Detection of Insider Threats", 2nd Edition, Carnegie Mellon University, http://www.cert.org/insider_threat/ , July 2006

23. Cheswick, W.R. and Bellovin, S.M. (1994), "Firewalls and Internet Security: Repelling the Wily Hacker", Addison-Wesley Publishing Company, 1994.

24. Chung, C.Y. Gerts, 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

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

26. Denning, D.E. (1987), "An Intrusion-Detection Model", IEEE Transactions on Software Engineering, Vol. SE-13, No. 2, February 1987, pp. 222-232.

27. Dhillon, G. and Moores, S. (2001), "Computer Crimes: Theorizing about the enemy within", Computers & Security, Vol. 20, No.8, pp. 715-723, 2001

28. Dowell, W. (1997), "I didn't set out to rob a bank: Interview with Toshihide Iguchi", Time Magazine, Vol.149, No. 6, 10 February, 1997

29. DTI. (2002). "Information Security Breaches Survey 2002". Department of Trade & Industry, April 2002. URN 02/318.

30. Duck, M. Bishop, P, and Read, R. (1996), "Data Communications for Engineers", Addision Wesley, ISBN 0-201-42788-5

31. Encarta (2000), "Microsoft Encarta World English Dictionary", Bloomsbury Publishing PLC, ISBN-13: 978-0747543718, 2000

32. Einwechter, N. (2002), "Preventing and Detecting Insider Attacks Using IDS", http://www.securityfocus.com/infocus/1558, March 20, 2002.

33. Escamilla, T. (1998), "Intrusion Detection: Network Security Beyond the Firewall", John Wiley & Sons, Inc. ISBN 0-471-29000-9, 1998.

34. Ferraiolo, D. and Kuhn, R. (1992), "Role-Based Access Control", In the Proceedings of the 15th National Computer Security Conference, 1992.

35. Ferraiolo, D.F. Gilbert, D.M. and Lynch, N. (1993), "An examination of federal and commercial access control policy needs." In NIST-NCSC National Computer Security Conference, pages 107-116, Baltimore, MD, September 20-23, 1993

36. Ferraiolo, D.F. Cugini, J.A. Kuhn, R.D. (1995), "Role-Based Access Control (RBAC): Features and Motivations", Computer Security Applications Conference

37. Forrst, S. Hofmeyr, S.A. Somayaji, A. and Longstaff, T.A. (1996), "A Sense of Self for Unix Processes", In Proceedings of the 1996 IEEE Symposium on Securit and Privacy, 120-128. IEEE Computer Security Society Press, Los Alamitos, CA

38. Furnell, S.M. and Dowland, P.S. (2000), "A Conceptual Architecture for Real-Time Intrusion Monitoring", Information Management & Computer Security, Vol.8, No.2. pp.65-74, 2000

39. Garg, A. Pramanik, S. Vidyaraman, S. and Upadhayaya, S. (2004), "Dynamic Document Reclassification for Preventing Insider Abuse", In Proceedings of the Fifth Annual IEEE Information Assurance Workshop (IAW 04), United States Military Academy, West Point, pp.218-225, 2004

40. Gaudin, S. (2000), "Case Study of Insider Sabotage: The Time Lloyd/Omega Case", Computer Security Journal, Volume XVI, Number 3, 2000

41. Gavrila, S.I. (1998), "Formal Specification for Role Based Access Control User/Role and Role/Role Relationship Management", Third ACM workshop on Role Based Access Control, October 1998

42. Gill, L. (2002), "IT Nightmare: The Enemy Within", NewsFactor Network, http://www.newsfactor.com/perl/story/18778.html , 29 July, 2002

43. Gordon, A. L. Loeb, M.P. Lucyshyn, W. and Richardson, R. (2004), "CSI/FBI Computer Crime And Security Survey", Computer Security Institute Publications, 2004

44. Gordon, A. L. Loeb, M.P. Lucyshyn, W. and Richardson, R. (2005), "CSI/FBI Computer Crime And Security Survey", Computer Security Institute Publications, 2005

45. Gordon, A. L. Loeb, M.P. Lucyshyn, W. and Richardson, R. (2006), "CSI/FBI Computer Crime And Security Survey", Computer Security Institute Publications, 2006

46. Haines, L. (2007), "The UK Office: Hotbed of Net Smut Addiction" , The Register, http://www.theregister.co.uk/2007/09/04/uk_office_downloading/ , 4th September 2007

47. Hofmeyr, S.A. Forrest, S. and Somayaji, A. (1998), "Intrusion Detection Using Sequences of System Calls", Journal of Computer Security, Vol.6 , No. 3, pp.151-180, 1998

48. Ilgun, K. (1993), "USTAT: A Real-Time Intrusion Detection System for Unix", Proceedings of the 1993 (IEEE) Symposium on Research in Security and Privacy, Oakland, CA, pp. 16-28, 1993

49. Ilgun, K. Kemmer, R. and Porras, P. (1995), "State Transition Analysis: A Rule Based Intrusion Detectino Approach", IEEE Transactions on Software Engineering, Vol. 21, No. 3, pp. 181-199, 1995

50. ISO/IEC 17799, (2005) "Information technology - Security techniques - Code of practice for information security management", International Organisation for Standardisation, June 2005

51. Keeney, M. Kowalski, E. Cappelli, D. Moore, A. Shimeall, T. Rogers, S. (2005), "Insider Threat Study: Computer System Sabotage in Critical Infrastructure Sectors", U.S. Secret Service and CERT Coordination Center/SEI, May 2005

52. Kerschbaum, F. Spafford, E.H. and Zamboni, D. (2000), "Using Embedded Sensors for Detecting Network Attacks", Proceedings of the 1st ACM Workshop on Intrusion Detection Systems, November 2000

53. Kim, G.H. and Spafford, E.H. (1993), "The Design and Implementation of Tripwire: A File System Integrity Checker", ACM Conference of Computer and Communications Security, pp.18-29, 1994

54. Kotadia, M. (2003), "Identity Theft Remains a Growing Problem", ZDNet UK, http://news.zdnet.co.uk/internet/0,1000000097,2137915,00.htm , 21st July 2003

55. Koziol, J. (2003), "Intrusion Detection with Snort", Sams Publishing, ISBN-13: 978-1578702817, May 2003

56. Kuhn, D.R. (1997), "Mutual Exclusion of Roles as a Means of Implementing Separation of Duty in Role-Based Access Control Systems", Second ACM Workshop on Role-Based Access Control, 1997

57. Kumar, S. and Spafford, E. (1994), "A Pattern Matching Model for Misuse Intrusion Detection", Proceedings of the 17th National Computer Security Conference, pp.11-21, 1994

58. Kumar, S. (1995), "Classification and Detection of Computer Intrusions", PhD Thesis, Purdue University, 1995

59. Lampson, B.W. (1971), "Protection". In 5th Princeton Symposium on Information Science and Systems, pp. 437-443, 1971, Reprinted in ACM Operating Systems Review 8(1):18-24, 1974

60. Lea, M. (2004), "The Ministry of Porn", The Sun Newspaper Online, August 26, 2004

61. Lee, S.C. and Heinbuch, D.V. (2001), "Training a Neural-Network Based Intrusion Detector to Recognise Novel Attacks", IEEE Transactions on Systems, Man & Cybernetics, Part A (Systems and Humans), Vol.31, No.4, pp.249

62. Leeson, N. (1997), "Rogue Trader", Time Warner Paperbacks, SBN-13: 978-0751517088, June 1997

63. Lindqvist, U. and Jonsson, E. (1997), "How to systematically Classify Computer Security Intrusions", In the Proceedings of the 1997 IEEE Symposium on Security and Privacy, IEEE Computer Society Press, May 4-7, 1997

64. Lindqvist, U. and Porras, P. (1999). "Detecting computer and network misuse through the production-based expert system toolset (P-BEST)". In the Proceedings of the 1999 IEEE Symposium on Security and Privacy, pp. 146-161, Oakland, California, May 9-12, 1999.

65. Lindqvist, U. and Porras, P. (2001), "eXpert-BSM: A Host Based Intrusion Detection Solution for Sun Solaris", 17th Annual Computer Security Applicatins Conference, New Orleans, December 2001

66. Low, W.L. Lee, J. and Teoh, P. (2002), "DIDAFIT: Detecting Intrusions in Databases through Fingerprinting Transactions", ICEIS 2002, Proceedings of the 4th International Conference on Enterprise Information Systems, Ciudal Real, Spain, April 2-6, 2002.

67. Lunt, T.F. (1990), "IDES: An Intelligent System for Detecting Intruders", In the Proceedings of the Symposium: Computer Security, Threat and Countermeasures", Rome, Italy, 1990

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

69. Magklaras, G.B, Furnell, S.M, (2002), "Insider Threat Prediction Tool: Evaluating the probability of IT misuse", Computers & Security, Vol. 21, No 1, pp62-73.

70. Marin, J. Ragsdale, D. and Surdu, J. (2001), "A Hybrid Approach to the Profile Creation and Intrusion Detection", In the Proceedings of the DARPA Information Survivability Conference and Exposition, Anaheim, CA, 12-14 June 2001

71. Mitchell, H.L. (2002), "Electronic auditing and fraud detection techniques", The Wisconsin Institute of Certified Accountants, www.wicpa.org, May 2002.

72. Moon, C. Park, D. Park, S. Baik, D. (2004), "Symmetric RBAC model that takes the separation of duty and role hierarchies into consideration", Computers & Security Vol.23, p.p 126-136, 2004

73. NetReplay, (2007), Chronicle Solutions, http://www.chroniclesolutions.com

74. Neumann, P.G. (1999), "Challenges of Insider Misuse", Workshop on Preventing, Detecting, and Responding to Malicious Insider Misuse, August, 1999.

75. Neumann, P.G. and Parker, D.B (1989), "A summary of computer misuse techniques", In the Proceedings of the 12th National Computer Security Conference, Baltimore, USA, 10-13 October, 1989, pp. 396-407.

76. Northcutt, S. and Novak, J. (2002), "Network Intrusion Detection, Third Edition", ISBN 0-7357-1265-4, September 2002

77. Orchestria (2007), http://www.orchestria.com

78. Papadaki, M. (2004), "Classifying and Responding to Network Intrusions", PhD Thesis, University of Plymouth, U.K. 2004

79. Park, J.S, Giordano, J. (2006), "Access Control Requirements for Preventing Insider Threats", ISI 2006, LNCS 3975, pp.529-534, 2006

80. Phyo, A.H. and Furnell, S.M. (2004), "A Detection Oriented Classification of Insider IT Misuse", Proceedings of the 3rd Security Conference, Las Vegas, USA, 14-15 April, 2004

81. Phyo, A.H. and Furnell, S.M. (2004b), "A Conceptual Framework for Monitoring Insider Misuse", Proceedings of Euromedia 2004, Hasselt, Belgium, 21-23 April, pp90-95, 2004

82. Porras, P. and Valdes, A. (1998), "Live Traffic Analysis of TCP/IP Gateways", In the Proceedings of the 1998 ISOC Symposium on Network and Distributed Systems Security, March 1998

83. Power, R. (1995), "Current and Future Danger: A CSI Primer on Computer Crime and Information Warfare", San Francisco, CA: Computer Security Institute.

84. Power, R. (2001), "CSI/FBI Computer Crime and Security Survey", Computer Security Issues & Trends, vol. VII, no. 1. Computer Security Institute. Spring 2001.

85. Power, R. (2002), "2002 CSI/FBI Computer Crime and Security Survey", Computer Security Issues & Trends, vol. VIII, no. 1. Computer Security Institute. Spring 2002.

86. Ptacek, T.H. and Newsham, T.N. (1999), "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection", Secure Networks Inc. 1999

87. Purser, S. (2002). "Why access control is difficult". Computers & Security ISSN 0167-4048, Vol. 21 No. 4, 2002

88. Ramaswamy, C. and Sandhu, R. (1998), "Role-Based Access Control Features in Commercial Database Management Systems", 21st National Information Systems Security Conference, October 1998

89. Randazzo, M.R. Keeney, M. Kowalski, E. Cappelli, D. Moore, A. (2004), "Insider Threat Study: Illicit Cyber Activity in the Banking and Finance Sector", U.S. Secret Service and CERT Coordination Center/SEI, August 2004

90. RFC 768, (1980), "User Datagram Protocol", Information Sciences Institute, http://www.faqs.org/rfcs/rfc768.html , August 1980

91. RFC 791, (1981), "Internet Protocol", DARPA Internet Program Protocol Specification, http://www.faqs.org/rfcs/rfc791.html , September 1981

92. RFC 792, (1981), "Internet Control Message Protocol", DARPA Internet Program Protocol Specification, http://www.faqs.org/rfcs/rfc792.html, September 1981

93. RFC 793, (1981), "Transmission Control Protocol", DARPA Internet Program Protocol Specification, http://www.faqs.org/rfcs/rfc793.html, September 1981

94. Richardson, R. (2003), "CSI/FBI Computer Crime And Security Survey", Computer Security Institute Publications, 2003

95. Ryan, J. Lin, M. and Miikkulainen, R. (1998), "Intrusion Detection with Neural Networks", in the Proceedings of Advance in Neural Information Processing Systems, Vol.10, The MIT Press, 1998

96. Sandhu, R. (1996), "Access Control: The Neglected Frontier", Proceedings of the first Australasian Conference on Information Security and Privacy, Wollongong, Australia, June 23-26, 1996

97. Sandhu, R.S. and Samarati, P. (1994) "Access Control: Principles and Practice", IEEE Communications Magazine Vol.32 No.9 pp.40-48, 1994

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

99. Shaw, E. Ruby, K.G, Post, J.M. (1998a), "Insider Threats to Critical Information Systems, Technical Report #2; Characteristics of the Vulnerable Critical Information Technology Insider (CITI)", Political Psychology Associates, Ltd., June 1998.

100. Shaw, E. Ruby, K.G, Post, J.M. (1998b), "The Insider Threat to Information Systems: The Psychology of the Dangerous Insider", Security Awareness Bulletin No.2-98

101. Silerschatz, A. Galvin, P. and Gagne, G. (2000), "Applied Operating System Concepts", First Edition, John Wiley and Sons Inc. ISBN 0-471-36580-4 , 2000

102. Singh, H. Furnell, S.M. Lines, B.L. and Dowland, P.S. (2001), "Investigating and Evaluating Behaviour Profiling and Intrusion Detection Using Data Mining", Proceedings of International Workshop on Mathematical Methods, Models and Architectures for Computer Networks Security, St. Petersburg, Russia, 21-23 May 2001

103. Taylor, J. (2000), "Constable Admits Using Police Database for Personal Reasons", ABC News Australia, http://www.abc.net.au/am/stories/s105272.htm , 2000

104. Teng, H.S. Chen, K. Lu, S.C. "Security Audit Trail Analysis Using Inductively Generated Predictive Rules". In the Proceedings of the 11th National Conference on Artificial Intelligence Applications, pp. 24-29, IEEE, March 1990

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

106. Uppuluri, P. and Sekar, R. (2001), "Experiences with Specification-Based Intrusion Detection", RAID 2001, LNCS 2212, pp.172-189, 2001

107. Verton, D. (2001), "Spy case demos insider threat: FBI suspect's system use went undetected", Computerworld.com, 26 February, 2001

108. Ward, P. and Smith, C.L. (2002), "The Development of Access Control Policies for Information Technology Systems", Computers & Security ISSN 0167-4048, Vol.21 No.4, 2002

109. Websense (2007), PortAuthority Technologies, http://www.websense.com/

Appendix A – Evaluation of Prototype

This section presents the test scenarios and event data used for evaluation of the prototype, and the outcomes of each test for detecting potential misfeasance. The activities listed below have been tested in varying scenarios, some of which violate contextual rules and thus result in misfeasance.

- File Access through an arbitrary application
- Deletion of files considered as property of the organisation
- Replication of confidential files
- Replication of confidential content
- Dissemination of confidential files
- Modification of database records
- Addition of users or new records to organisation's registry databases
- Modification of system/application settings

File Access through Arbitrary Application

This section evaluates the prototype system's ability to detect access of files through applications that differ from the defined norm for accessing the file concerned.

The logic of the analysis procedure for file access test scenario is described in Chapter 7, section 7.7.1.

Test Scenario:

This scenario presents a user accessing a file through an application that does not match the norm defined in the security policy associated with the file.

File ID	4
Machine Name	PSQ_A304_FS1
File Path	S:\Technology\Expenses.mdb
File Custodian	Steven Furnell
File Description	Expense claims for technology department
Application for Access	MS Access

Entry of File Details in File Inventory within Knowledgebase

Event ID	6
User Name	Sue Kendall
Machine Name	PSQ_A304_WS3
Application Name	Open Office
Server Name	PSQ_A304_FS1
File Path	S:\Technology\Expenses.mdb

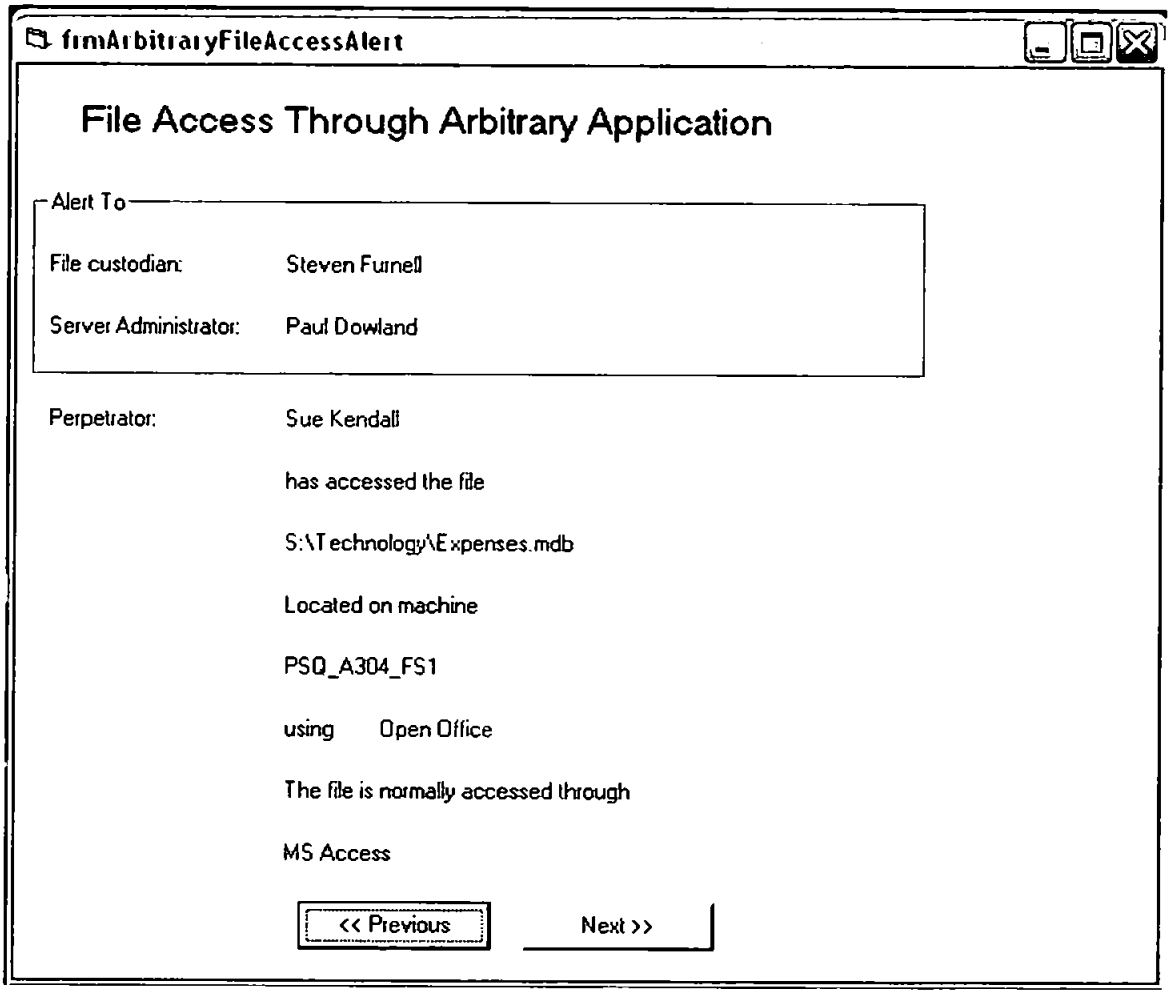
Entry of File Access Log within the Events Database

Event ID	Command ID	Event Type
6	6	File Access

Entry of Event log in the Events data table of Events database

Outcome:

The analysis engine logs the event to the *Alert* database, because the application utilised for accessing the file does not match the defined norm.



Outcome of Arbitrary File Access Test Scenario

Deletion of Files Considered as Property of the Organisation

This section evaluates the prototype system's ability to detect deletion of files considered as intellectual property of the organisation.

Test Scenario:

This scenario presents a user deleting a file considered as intellectual property of the organisation. The logic for the analysis procedure is described in Chapter 7, section 7.7.2

File ID	2
Machine Name	PSQ_A304_FS1
File Path	S:\Aung\Thesis\Thesis.doc
File Custodian	Steven Furnell
File Description	Entire Thesis
Application for Access	MS Word

Entry of File Details in File Inventory within Knowledgebase

Event ID	24
User Name	Sevi
Machine Name	PSQ_A304_WS3
Application Name	Windows Explorer
Server Name	PSQ_A304_FS1
File Path	S:\Aung\Thesis\Thesis.doc

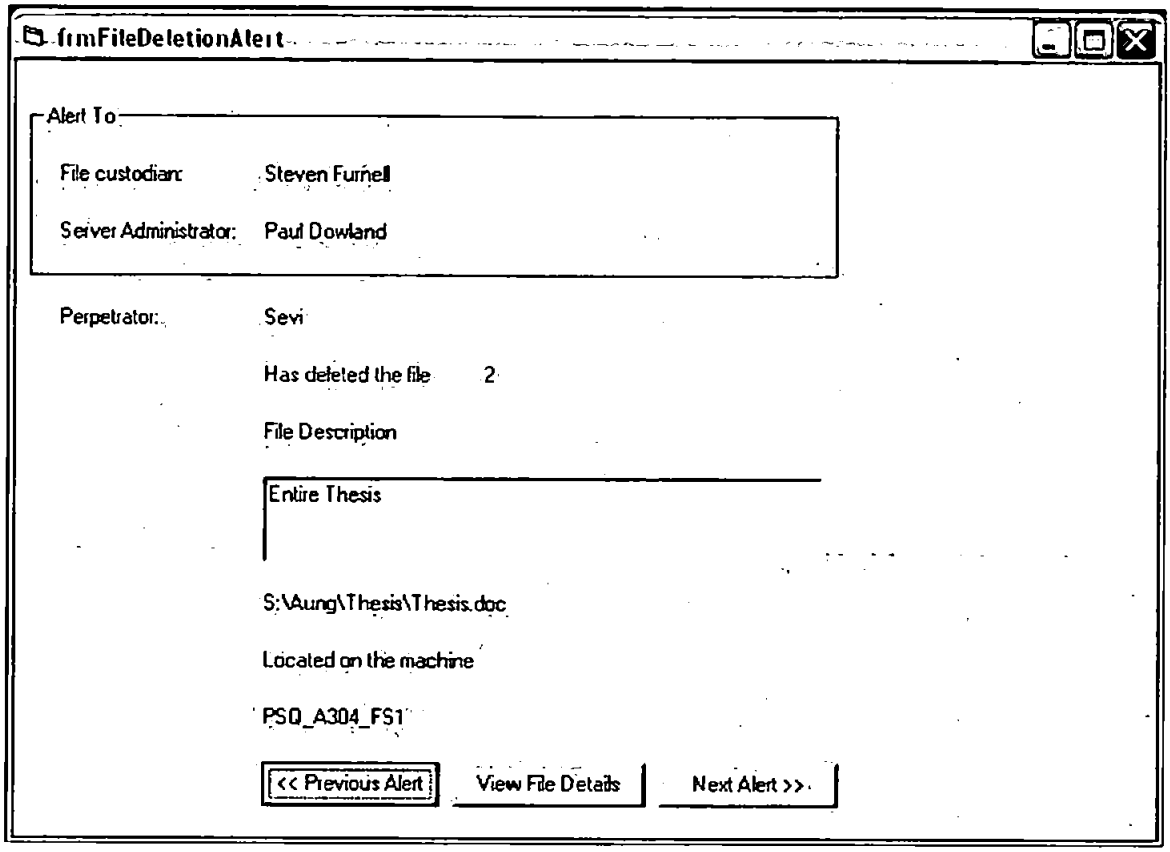
Entry of File Deletion Log within the Events Database

Event ID	Command ID	Event Type
24	9	File Deletion

Entry of Event log in the Events data table of Events database

Outcome:

The analysis engine logs the event to the *Alert* database, because the file is listed in the *File Inventory*, and thus considered as intellectual property of the organisation.



Outcome of File Deletion Test Scenario

Replication of Confidential Files

Test Scenario:

This scenario presents a user replicating a file that is allowed to be saved to a removable media. The logic for the analysis procedure is described in Chapter 7, section 7.7.3.

File ID	3
Machine Name	PSQ_A304_FS1
File Path	S:\Nath\SharedNewDesign.jpg
File Custodian	Nathan Clarke
File Description	Picture
Application For Access	ACD See
Partial Replication	False
Whole Replication	False
Save to Removable	True

Entry of File Details in File Inventory within Knowledgebase

Event ID	17
Command ID	10
Machine Name	PSQ_A304_WS2
User Name	Aung Htike Phyo
Application Name	ACD See
Server Name	PSQ_A304_FS1
File Path	S:\Nath\SharedNewDesign.jpg

Entry of File Replication (Copy) Log within Event Database

Event ID	18
Command ID	11
Machine Name	PSQ_A304_WS2
User Name	Aung Htike Phyo
Application Name	ACD See
Server Name	PSQ_A304_WS2
File Path	C:\My Documents\NewDesign.jpg

Entry of File Replication (Paste) Log within Event Database

Event ID	Command ID	Event Type
17	10	File Replication
18	11	File Replication

Entry of Event log in the Events data table of Events database

Outcome:

The analysis engine does not log the event to the *Alert* database, because the replication of the file to a removable media is defined as acceptable.

Test Scenario:

This scenario presents a user replicating a file that is not allowed to be saved to a removable media, but allowed to be replicated. The logic for the analysis procedure is described in chapter 7, section 7.7.3.

File ID	5
Machine Name	PSQ_A304_FS1
File Path	S:\Aung\Thesis\Chapter1.Doc
File Custodian	Aung Htike Phyo
File Description	Thesis Chapter 1
Application For Access	MS Word
Partial Replication	False
Whole Replication	True
Save to Removable	False

Entry of File Details in File Inventory within Knowledgebase

Event ID	20
Command ID	10
Machine Name	PSQ_A304_WS3
User Name	Sevi
Application Name	Windows Explorer
Server Name	PSQ_A304_FS1
File Path	S:\Aung\Thesis\Chapter1.Doc

Entry of File Replication (Copy) Log within Event Database

Event ID	21
Command ID	11
Machine Name	PSQ_A304_WS3
User Name	Sevi
Application Name	Windows Explorer
Server Name	PSQ_A304_WS3
File Path	C:\My Documents\Chapter1.doc

Entry of File Replication (Paste) Log within Event Database

Event ID	Command ID	Event Type
20	10	File Replication
21	11	File Replication

Entry of Event log in the Events data table of Events database

Outcome:

The analysis engine logs the event to the *Alert* database with the alert status set to false, thus keeping track of the replicas, but no alerts generated, because replication of the file is defined as acceptable.

Test Scenario:This scenario presents a user replicating a file that is not allowed to be saved to a removable media, and not allowed to be replicated. The logic for the analysis procedure in described in chapter 7, section 7.7.3.

File ID	2
Machine Name	PSQ_A304_FS1
File Path	S:\Aung\Thesis\Thesis.doc
File Custodian	Steven Furnell
File Description	Entire Thesis
Application For Access	MS Word
Partial Replication	True
Whole Replication	False
Save to Removable	False

Entry of File Details in File Inventory within Knowledgebase

Event ID	22
Command ID	10
Machine Name	PSQ_A304_WS2
User Name	Nathan Clarke
Application Name	Windows Explorer
Server Name	PSQ_A304_FS1
File Path	S:\Aung\Thesis\Thesis.doc

Entry of File Replication (Copy) Log within Event Database

Event ID	23
Command ID	11
Machine Name	PSQ_A304_WS2
User Name	Nathan Clarke
Application Name	Windows Explorer
Server Name	PSQ_A304_WS2
File Path	C:\My Documents\Thesis.doc

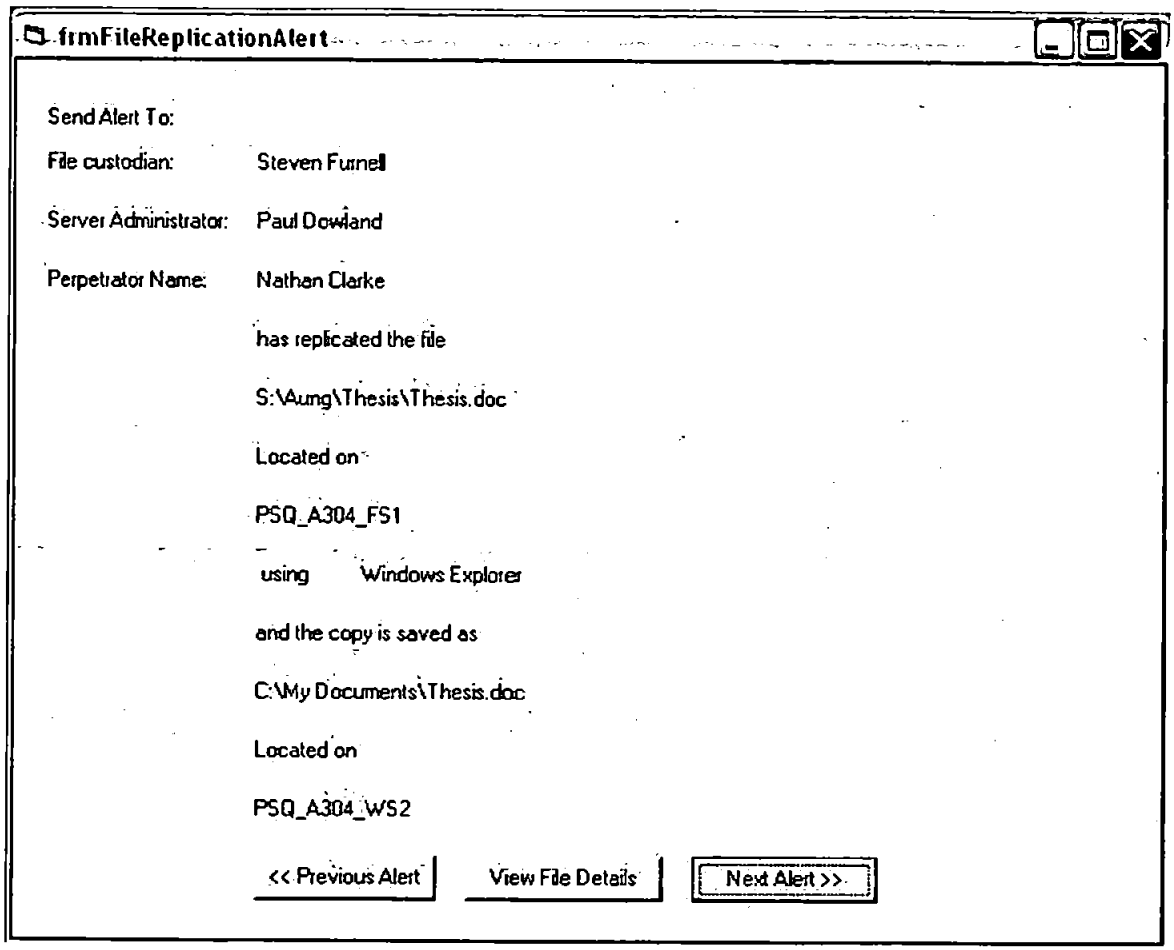
Entry of File Replication (Paste) Log within Event Database

Event ID	Command ID	Event Type
22	10	File Replication
23	11	File Replication

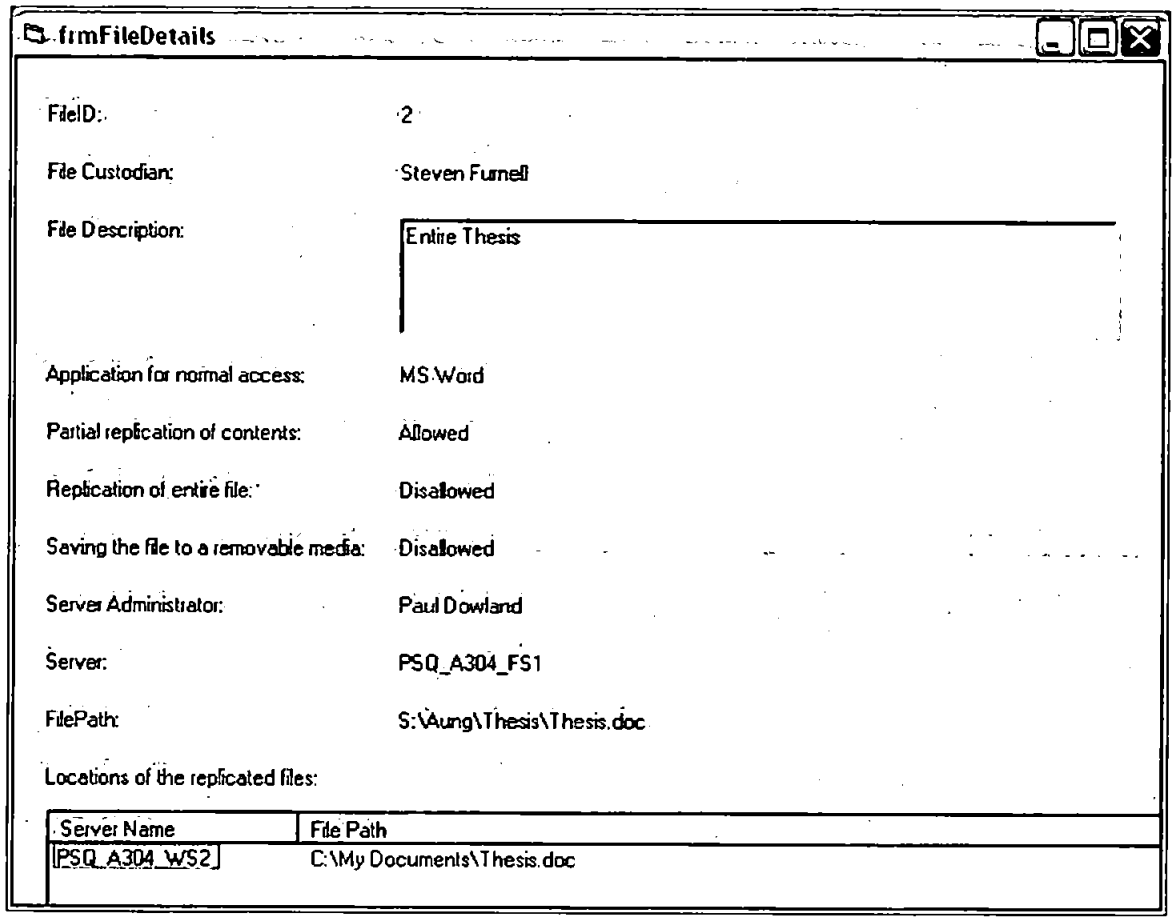
Entry of Event log in the Events data table of Events database

Outcome:

The analysis engine logs the event to the *Alert* database with the alert status set to true, thus keeping track of the replicas and generating alerts, because replication of the file is not acceptable.



Outcome of File Replication Test Scenario



File Details from Outcome of Test Scenario

Replication of Confidential Content

Test Scenario:

This scenario presents a user replicating the contest of a file that is not allowed to be replicated, and replication of contents is not acceptable. The logic for the analysis procedure is described in chapter 7, section 7.7.4.

File ID	6
Machine Name	PSQ_A304_FS1
File Path	S:\Aung\Demo\SystemDesign.doc
File Custodian	Aung Htike Phyo
File Description	Misfeasor monitoring tool design.
Application For Access	Open Office
Partial Replication	False
Whole Replication	False
Save to Removable	False

Entry of File Details in File Inventory within Knowledgebase

Event ID	26
Command ID	1
Machine Name	PSQ_A304_WS1
User Name	Nathan Clarke
Application Name	Open Office
Server Name	PSQ_A304_FS1
File Path	S:\Aung\Demo\SystemDesign.doc

Entry of Data Replication (Copy) Log within Event Database

Event ID	27
Command ID	2
Machine Name	PSQ_A304_WS1
User Name	Nathan Clarke
Application Name	Open Office
Server Name	PSQ_A304_WS1
File Path	C:\My Documents\DetectionSystems.doc

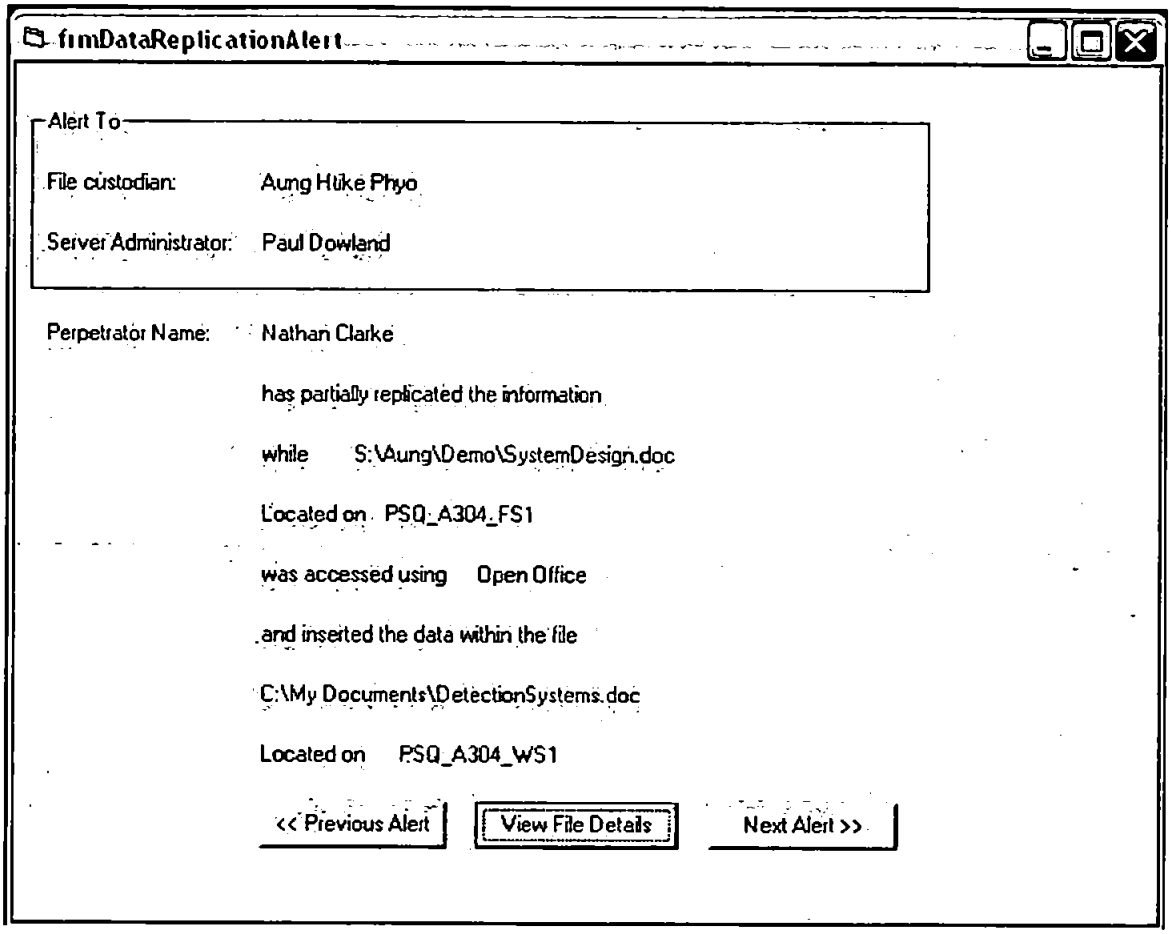
Entry of Data Replication (Paste) Log within Event Database

Event ID	Command ID	Event Type
26	1	Partial Data Replication
27	2	Partial Data Replication

Entry of Event log in the Events data table of Events database

Outcome:

The analysis engine logs the event to the *Alert* database, because replication of the file's contents is not acceptable.



Outcome of Content Replication Test Scenario

Dissemination of Confidential Files

For the analysis of file dissemination, the analysis procedure utilises information from several data tables within the knowledge base, which includes:

- Employee Registry
- List of Internal Machines
- Users allowed to receive each inventoried file

- File Inventory
- List of machine allowed to host files from each server

The logic of the analysis procedure for all file transfer test scenarios is describe in chapter 7, section 7.7.5.

Machine Name	IP Address	System Administrator
PSQ_A304_FS1	192.168.0.1	Paul Dowland
PSQ_A304_WS1	192.168.0.3	Tarik
PSQ_A304_PS1	192.168.0.2	Nathan
PSQ_A304_WS2	192.168.0.4	Shukor Razak
PSQ_A304_WS3	192.168.0.5	Aung Htike Phyo
PSQ_A304_MS1	192.168.0.6	Andy
PSQ_A304_WS4	192.168.0.7	Sevi

Table: List of Internal Machines

Full Name	Report To	E-mail Address
Steven Furnell		steve@jack.see.plymouth.ac.uk
Paul Dowland	Steven Furnell	paul@jack.see.plymouth.ac.uk
Aung Htike Phyo	Steven Furnell	aung@jack.see.plymouth.ac.uk
Nathan Clarke	Steven Furnell	nathan@jack.see.plymouth.ac.uk
Sevi	Nathan Clarke	sevi@jack.see.plymouth.ac.uk

Full Name	Report To	E-mail Address
Shukor Razak	Paul Dowland	shukor@jack.see.plymouth.ac.uk
Sue Kendall	Steven Furnell	sue.kendall@plymouth.ac.uk
Jules	Sue Kendall	jules@plymouth.ac.uk

Table: Employee Registry within the Knowledgebase

Test Scenario:

This scenario presents a user transferring a file that is not allowed to be saved to removable media. Specific conditions violating the policy were tested, which includes:

- Communication server is not an internal machine
- Recipient is not an insider
- Recipient is an insider but does not have authority to receive the file involved
- Recipient is an insider and has authority to receive the file involved, but the machine utilised by the recipient is not authorised to host files from the originating file server.

Test: Communication Server is Not an Internal Machine

File ID	5
Machine Name	PSQ_A304_FS1
File Path	S:\Aung\Thesis\Chapter1.Doc
File Custodian	Aung Htike Phyo

File Description	Thesis Chapter 1
Application For Access	MS Word
Partial Replication	False
Whole Replication	True
Save to Removable	False

Entry of File Details in File Inventory within Knowledgebase

Event ID	14
Sender Name	Sevi
Sender Machine Name	PSQ_A304_WS2
Application Name	Email
Communication Server	200.168.0.6
File Server	PSQ_A304_FS1
File Path	S:\Aung\Thesis\Chapter1.Doc
Sender Address	sevi@jack.see.plymouth.ac.uk
Receiver Address	thano@hotmail.com

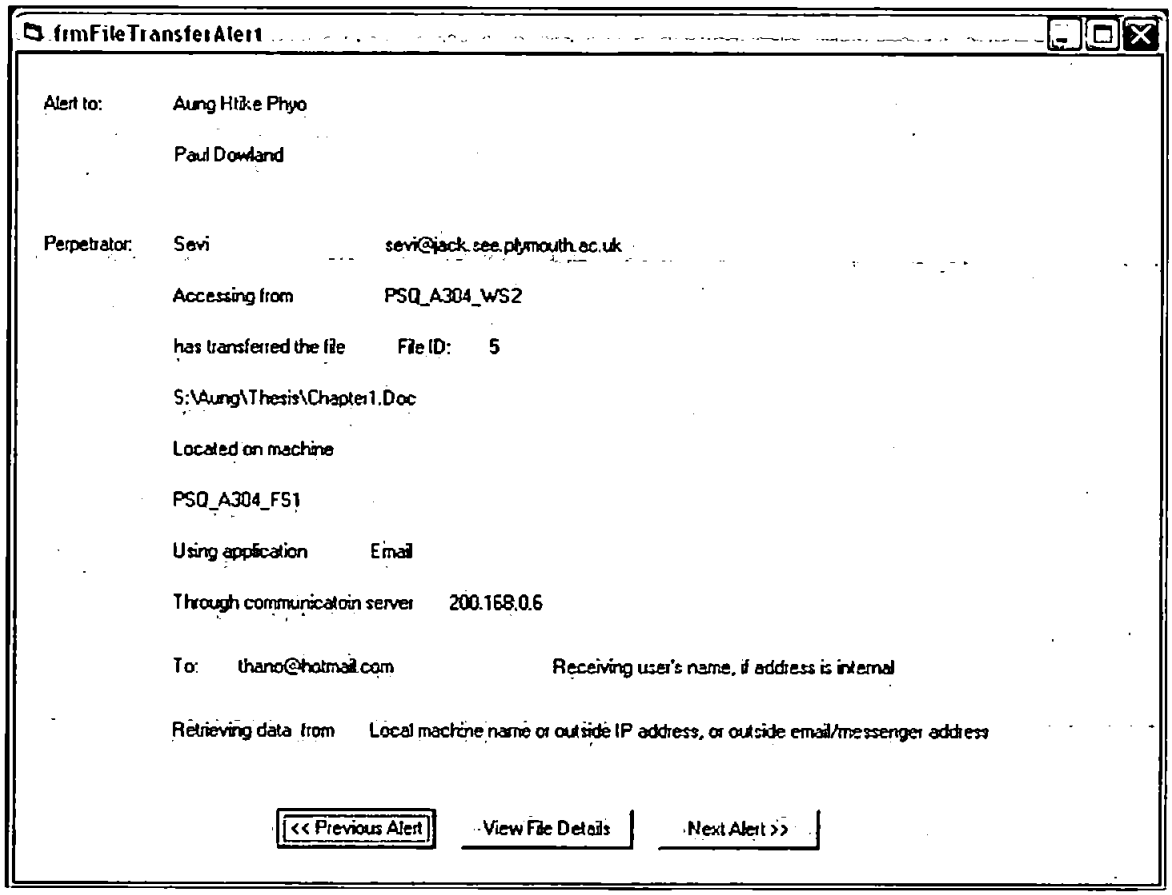
Entry of Data Transfer Log

Event ID	Command ID	Event Type
14	4	File Transfer

Entry of Event log in the Events data table of Events database

Outcome:

The analysis engine logs the event to the *Alert* database, because the communication server mediating the transfer is not an internal machine.



Outcome of File Transfer Test Scenario 1

Test: Recipient is Not an Insider

File ID	2
Machine Name	PSQ_A304_FS1

File Path	S:\Aung\Thesis\Thesis.doc
File Custodian	Steven Furnell
File Description	Entire Thesis
Application For Access	MS Word
Partial Replication	True
Whole Replication	False
Save to Removable	False

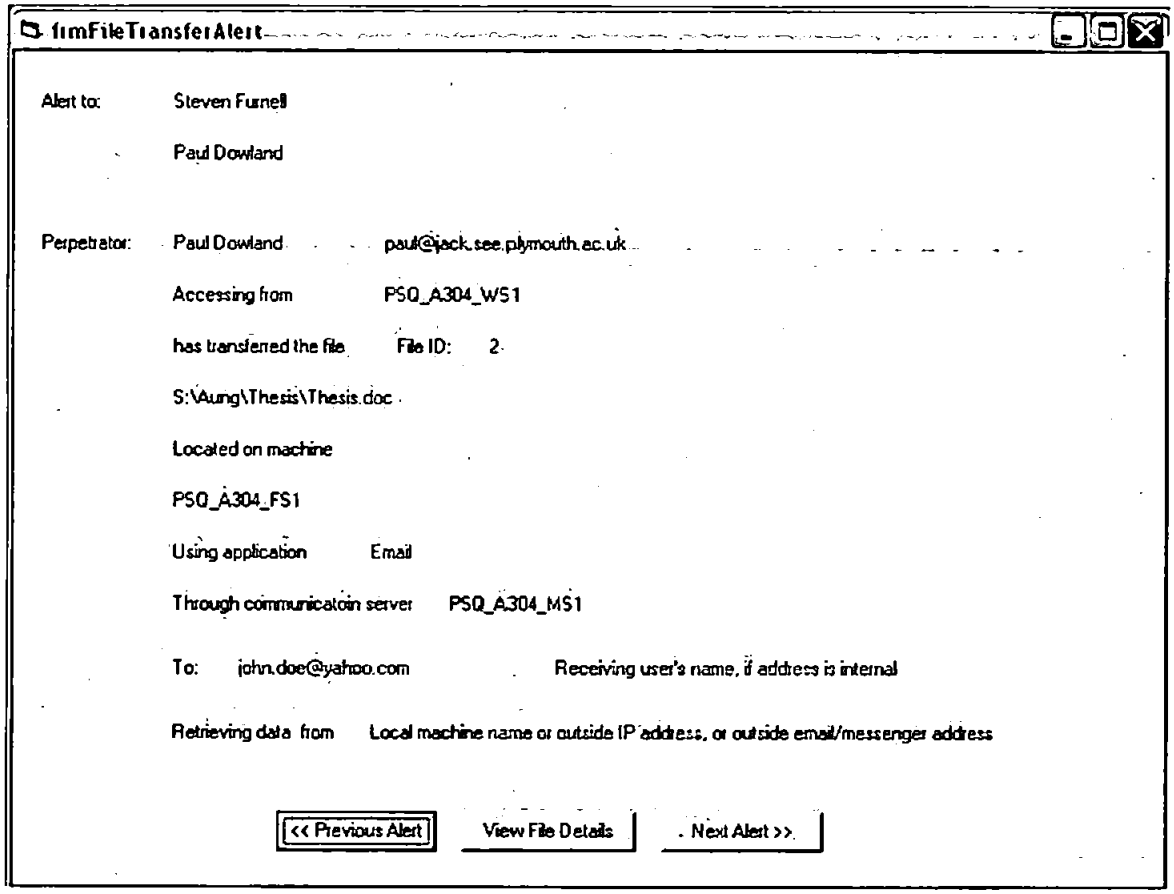
Entry of File Details in File Inventory within Knowledgebase

Event ID	28
Sender Name	Paul Dowland
Sender Machine Name	PSQ_A304_WS1
Application Name	Email
Communication Server	192.168.0.6
File Server	PSQ_A304_FS1
File Path	S:\Aung\Thesis\Thesis.doc
Sender Address	paul@jack.see.plymouth.ac.uk
Receiver Address	john.doe@yahoo.com

Entry of Data Transfer Log

Event ID	Command ID	Event Type
28	4	File Transfer

Entry of Event log in the Events data table of Events database



Outcome of File Transfer Test Scenario 2

Test: Recipient is an Insider, but does not have Authority to Access

File ID	4
Machine Name	PSQ_A304_FS1

File Path	S:\Technology\Expenses.mdb
File Custodian	Steven Furnell
File Description	Expense claims for technology department
Application For Access	MS Access
Partial Replication	True
Whole Replication	False
Save to Removable	False

Entry of File Details in File Inventory within Knowledgebase

Event ID	12
Sender Name	Sue Kendall
Sender Machine Name	PSQ_A304_WS1
Application Name	Email
Communication Server	192.168.0.6
File Server	PSQ_A304_FS1
File Path	S:\Technology\Expenses.mdb
Sender Address	sue.kendall@plymouth.ac.uk
Receiver Address	jules@plymouth.ac.uk

Entry of Data Transfer Log

Event ID	Command ID	Event Type
12	4	File Transfer
13	5	Data Retrieval

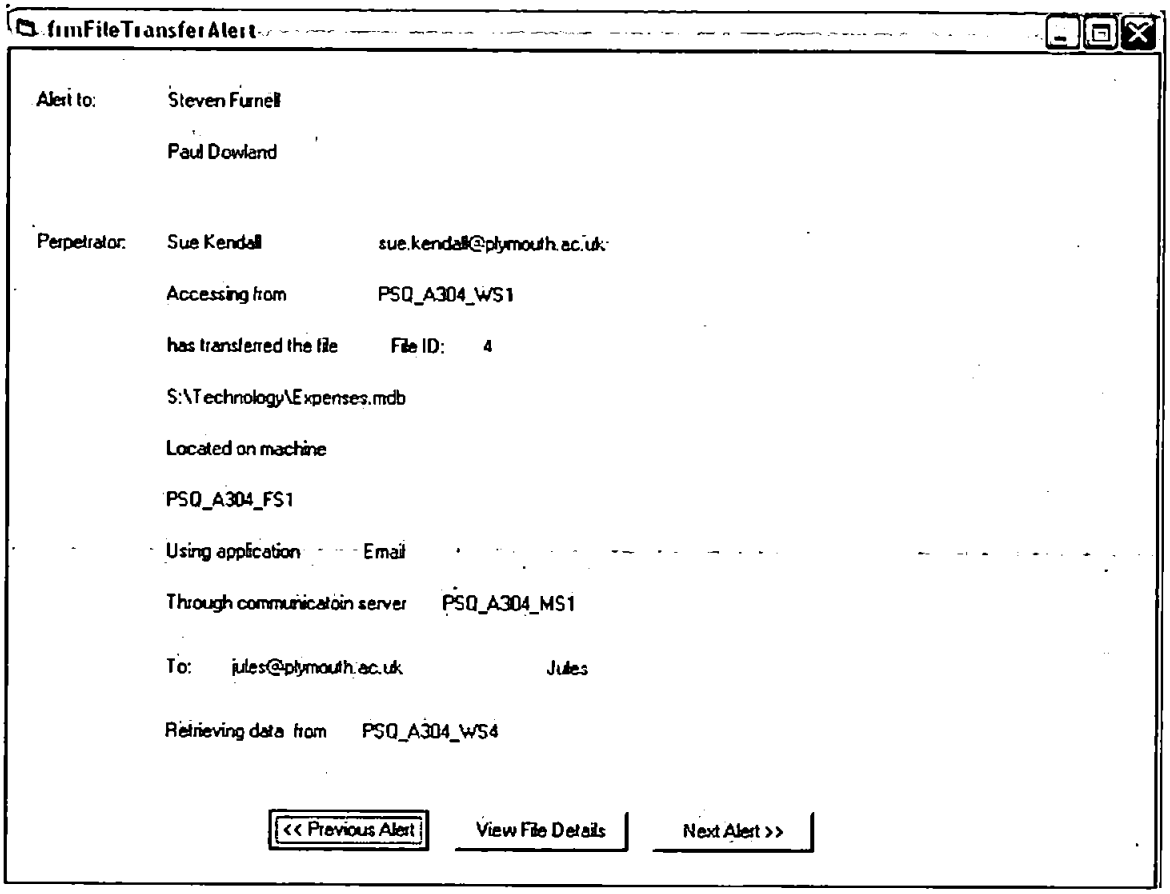
Entry of Event log in the Events data table of Events database

Event ID	Reference Event ID	Application Name	Receiver Machine
13	12	Email	192.168.0.7

Entry of Data Retrieval log table of Events database

Outcome:

The analysis engine logs the event to Alert database, because although the recipient is an insider, the recipient is not authorised to access the file received.



Outcome of File Transfer Test Scenario 3

Test: Recipient is an Insider, and has Authority to Access, but the Machine Utilised for Accessing the File is not authorised.

File ID	2
Machine Name	PSQ_A304_FS1
File Path	S:\Aung\Thesis\Thesis.doc
File Custodian	Steven Furnell

File Description	Entire Thesis
Application For Access	MS Word
Partial Replication	True
Whole Replication	False
Save to Removable	False

Entry of File Details in File Inventory within Knowledgebase

Event ID	15
Sender Name	Paul Dowland
Sender Machine Name	PSQ_A304_WS1
Application Name	Email
Communication Server	192.168.0.6
File Server	PSQ_A304_FS1
File Path	S:\Aung\Thesis\Thesis.doc
Sender Address	paul@jack.see.plymouth.ac.uk
Receiver Address	shukor@jack.see.plymouth.ac.uk

Entry of Data Transfer Log

Event ID	Command ID	Event Type
15	4	File Transfer
16	5	Data Retrieval

Entry of Event log in the Events data table of Events database

Event ID	Reference Event ID	Application Name	Receiver Machine
16	15	Email	192.168.0.7

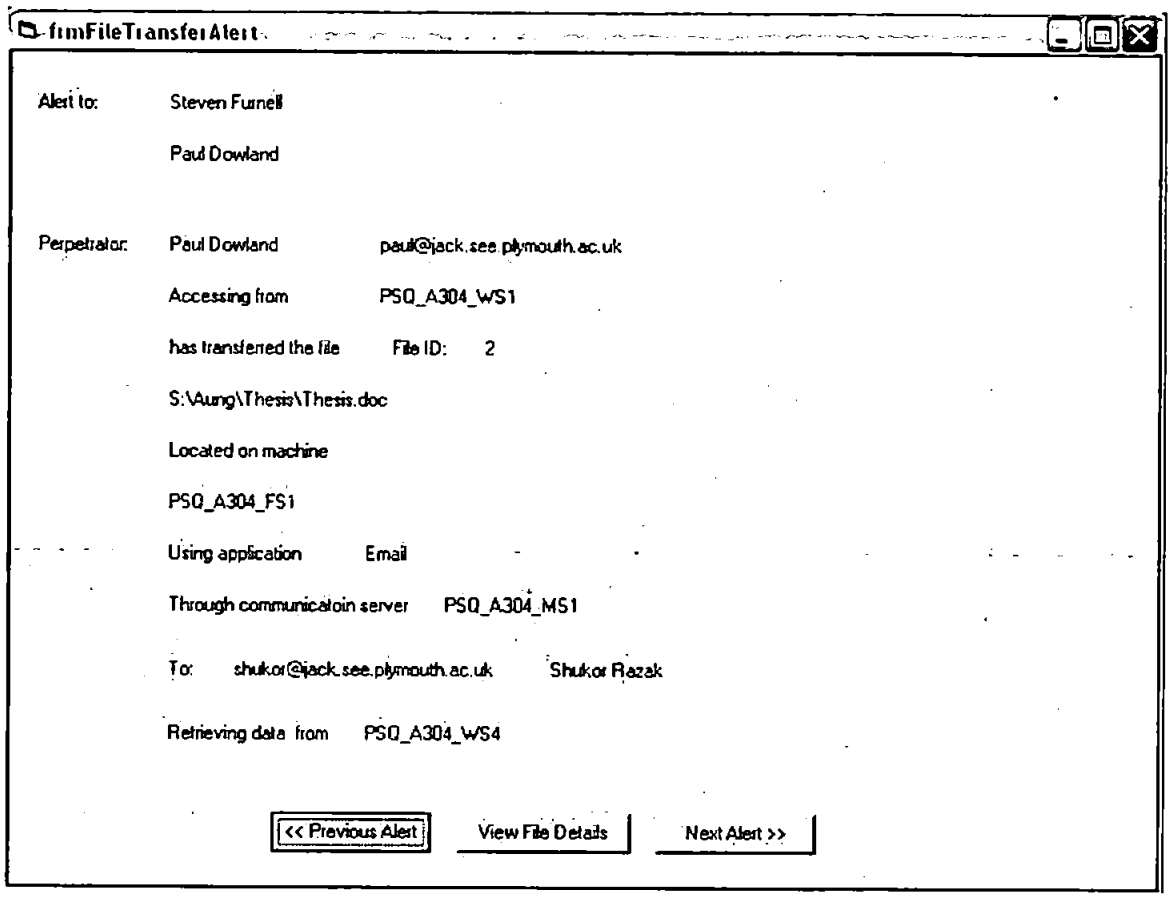
Entry of Data Retrieval log table of Events database

File ID	Full Name
2	Shukor Razak

Table: List of Users Authorised to Receive each File

Server Name	Machine Name
PSQ_A304_FS1	PSQ_A304_WS1
PSQ_A304_FS1	PSQ_A304_WS2
PSQ_A304_FS1	PSQ_A304_WS3

Table: List of Machine Authorised to Host Files from Each Server



Outcome of File Transfer Test Scenario 4

Modification of Database Records

The logic of the analysis procedure for all database access scenarios is described in chapter 7, section 7.8.

Test Scenario:

This scenario presents a user adding a record to a data table, and the condition states that a corresponding reference record must exist. For this particular example, the perpetrator assigns the person who does not exist in employee registry as a file custodian.

File ID	7
Machine Name	PSQ_A304_FS1
File Path	S:\MScStudents\StudentList.doc
File Custodian	Jackie C
File Description	Student list
Application For Access	Open Office
Partial Replication	False
Whole Replication	False
Save to Removable	False

Entry made to the File Inventory table of the knowledgebase

Event ID	11
User Name	Aung Htike Phyo
Machine Name	PSQ_A304_WS2
Application Name	Database Application
Server Name	PSQ_A304_FS1
File Path	E:\Aung's Thesis\Demo 2007\Alerts\Monitoring Engine\Event Generator\Contextual Information\ContextualInfo.mdb
Table Name	FileLocation
Record ID	7

Entry of Database Access Log

Query Name	AssignFileCustodian
Server Name	PSQ_A304_FS1
File Path	E:\Aung's Thesis\Demo 2007\Alerts\Monitoring Engine\Event Generator\Contextual Information\ContextualInfo.mdb
Table Name	FileLocation
Table Custodian	Paul Dowland
Primary Key Attribute	FileID
Common Attribute	FileCustodian
Attribute To Verify	FileCustodian

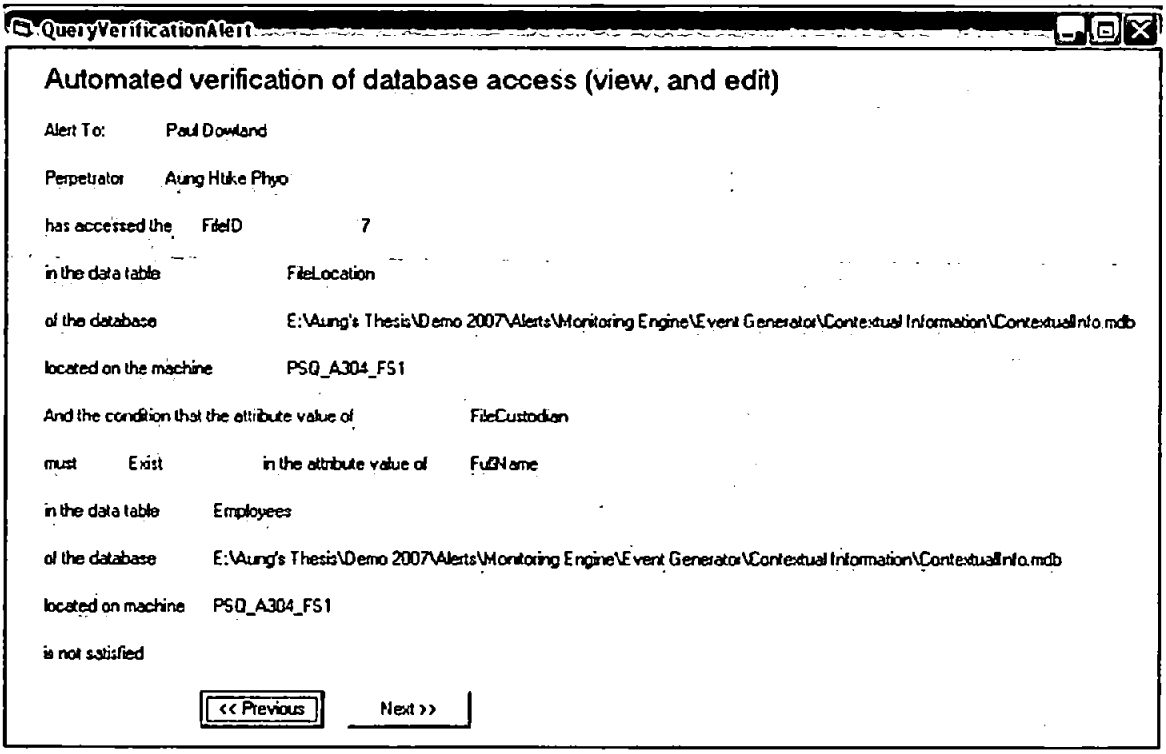
Entry of Data Verification Table

Query Name	AssignFileCustodian
Server Name	PSQ_A304_FS1
File Path	E:\Aung's Thesis\Demo 2007\Alerts\Monitoring Engine\Event Generator\Contextual Information\ContextualInfo.mdb
Table Name	Employees
Common Attribute	FullName
Attribute To Verify	FullName
Condition	Exist

Entry of Data Verification Reference Table

Outcome:

The analysis engine logs the event to the Alert database, because the assigned custodian of the file does not exit in the employee registry.



Outcome of the Database Access Test Scenario 1

Test Scenario:

This scenario presents a user adding a record do a data table, and the condition states that a certain value from the new record must equate to a certain value from the corresponding reference record. For this example, the perpetrator updates the account of a customer, the balance of which must equal to the total deposits made.

Query Name	BalanceCheck
Server Name	PSQ_A304_FS1
File Path	E:\Aung's Thesis\Demo 2007\Test Databases\CustomerRecords.mdb
Table Name	Accounts
Table Custodian	Accounts Manager
Primary Key Attribute	CustomerID
Common Attribute	CustomerID
Attribute To Verify	AccountBalance

Entry of Data Verification Table

Query Name	BalanceCheck
Server Name	PSQ_A304_FS1
File Path	E:\Aung's Thesis\Demo 2007\Test Databases\CustomerRecords.mdb
Table Name	Deposits
Common Attribute	AccountID
Attribute To Verify	TotalDeposits
Condition	Be Equal

Entry of Data Verification Reference Table

Customer ID	Customer Name	Account Balance
1	Account 1	100

Entry of Account data table

Account ID	Total Deposits
1	1000

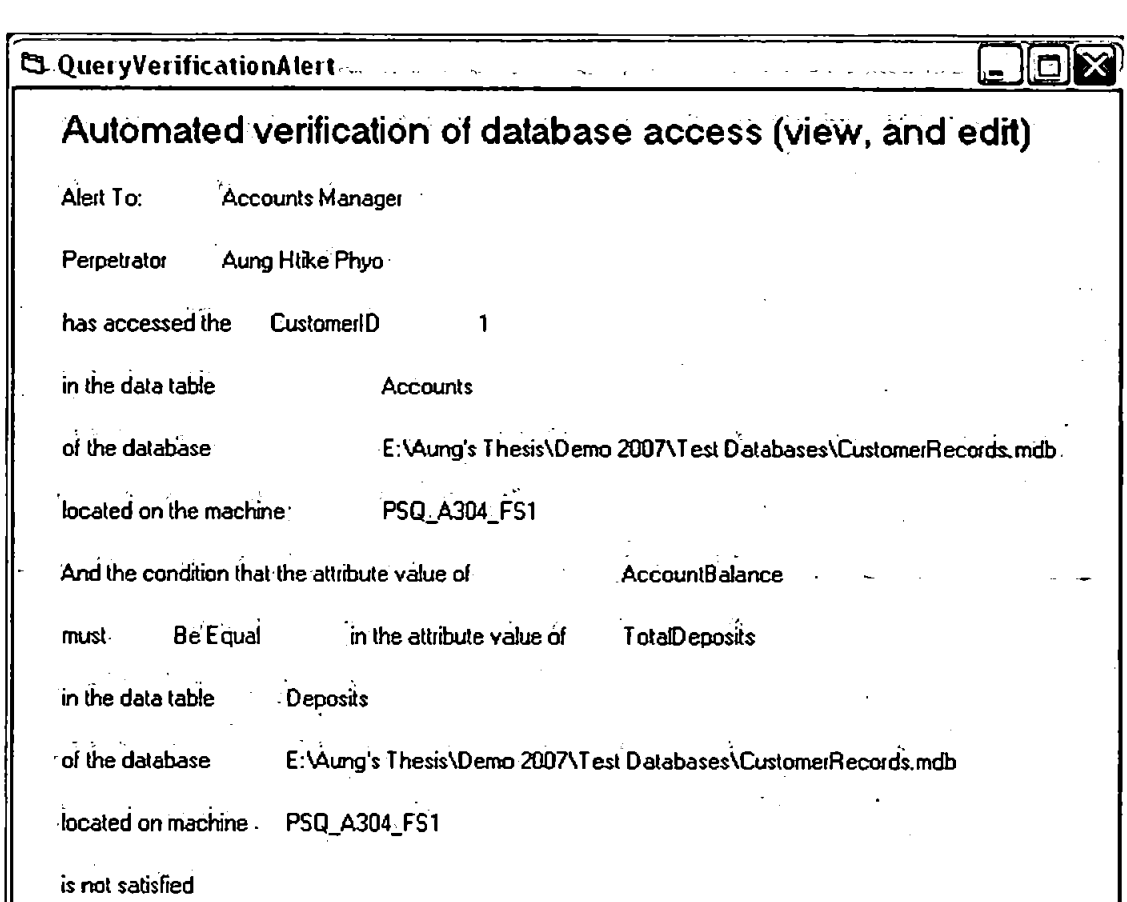
Entry of Deposits data table

Event ID	Command ID	Event Type
29	8	Database Access

Entry of Event log

Outcome:

The analysis engine logs the event to the Alert database, because the updated account balance does not equal to the total deposits.



Outcome of Database Access Test Scenario 2

Test Scenario:

This scenario presents a user adding a record to a data table, and the condition states that a certain value from the corresponding reference record must be true.

Query Name	VerifyRecordAccess
Server Name	PSQ_A304_FS1
File Path	E:\Aung's Thesis\Demo 2007\Test Databases\AccessValidation.mdb
Table Name	VerifiedRecordAccess
Table Custodian	Database Manager
Primary Key Attribute	LogID
Common Attribute	CustomerID
Attribute To Verify	CustomerID

Entry of Data Verification Table

Query Name	VerifyRecordAccess
Server Name	PSQ_A304_FS1
File Path	E:\Aung's Thesis\Demo 2007\Test Databases\AccessValidation.mdb
Table Name	AccessRequests
Common Attribute	CustomerID
Attribute To Verify	Verified
Condition	Be True

Entry of Data Verification Reference Table

Log ID	Employee ID	Customer ID	Access Request ID	Access ID
1	Aung Htike Phyo	1	1	1

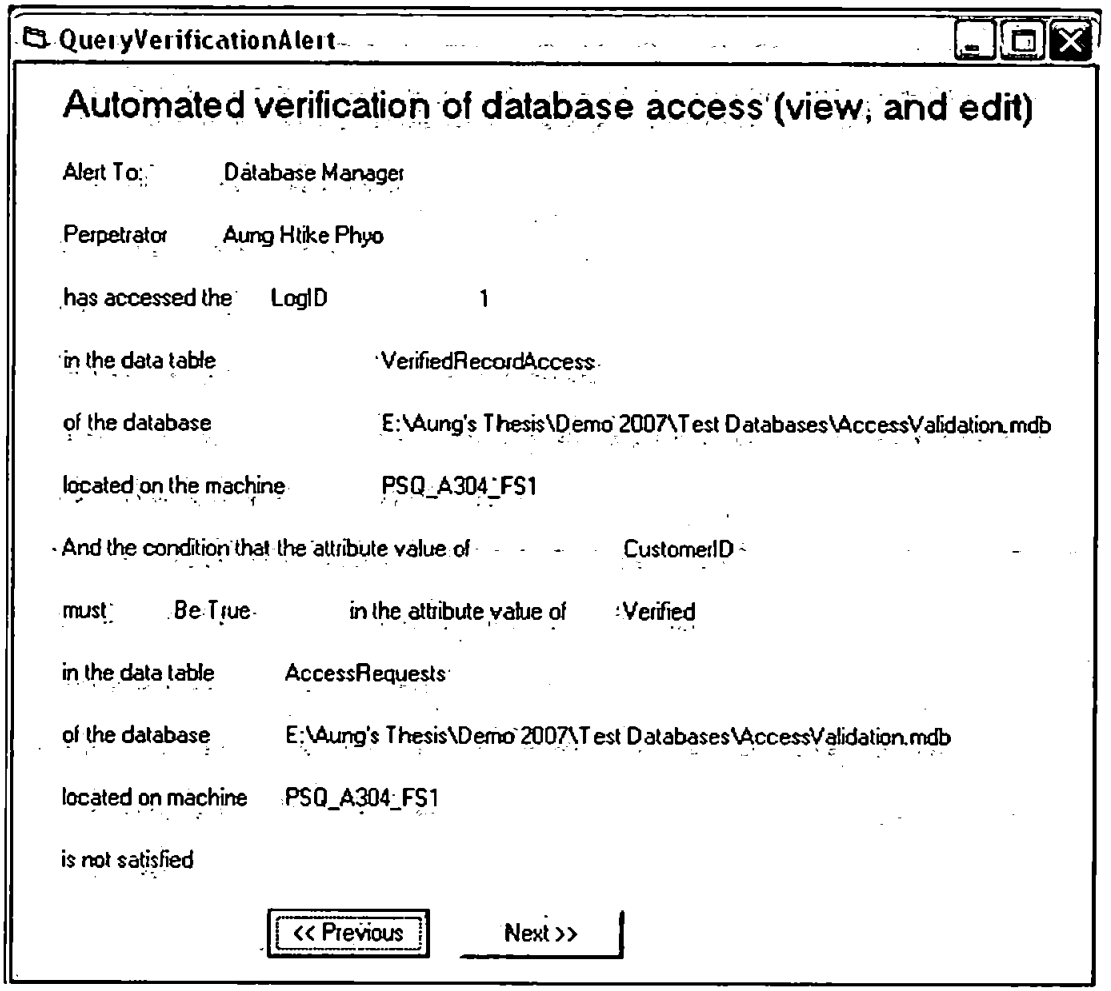
Entry of VerifiedRecordAccess Table

Log ID	Customer ID	Verified
1	1	False

Entry of AccessRequests Table

Outcome:

The analysis engine logs the event to the Alert database, because the "Verified" attribute of the reference record in AccessRequests table is false.



Outcome of Database Access Test Scenario 3

Addition of Users or New Records to Organisation's Registry Databases

The logic of the analysis procedure for the test scenarios is described in chapter 7, section 7.81.

Test Scenario:

This scenario presents a user adding a new record to the employee registry.

Event ID	9
Machine Name	PSQ_A304_WS2
User Name	Steven Furnell
Server Name	PSQ_A304_FS1
File Path	E:\Aung's Thesis\Demo 2007\Alerts\Monitoring Engine\Event Generator\Contextual Information\ContextualInfo.mdb
Table Name	Employees
Record ID	8

Log Entry of the User Management Table

Employee ID	8
Full Name	Jules
Department	Technology
Immediate Superior	Sue Kendall
Job Title	Receptionist
Email	Jules@plymouth.ac.uk

Entry of the Employees data table

List Name	Employees
Server Name	PSQ_A304_FS1
File Path	E:\Aung's Thesis\Demo 2007\Alerts\Monitoring Engine\Event Generator\Contextual Information\ContextualInfo.mdb
Table Name	Employees
Primary Key Attribute	EmployeeID
Record Name Attribute	FullName
Attribute Name	ImmediateSuperior
List Description	The list of employees

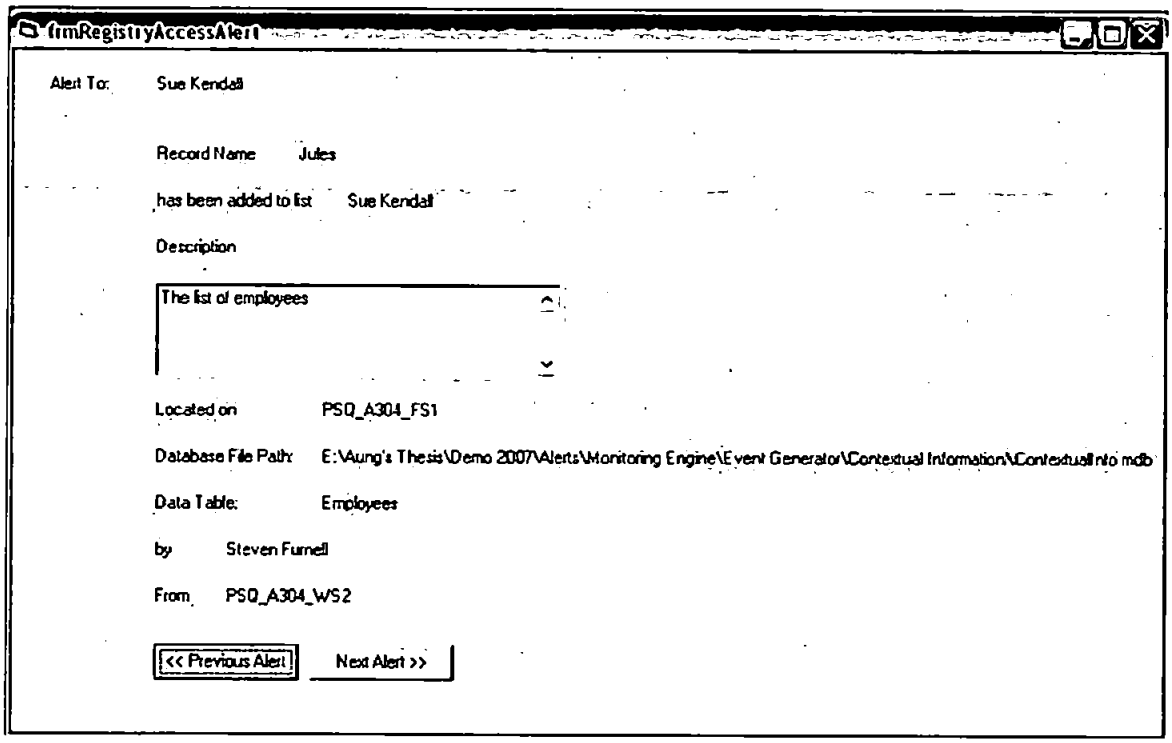
Entry of the Lists data table (indicates monitored registries)

List Name	Employees
Server Name	PSQ_A304_FS1
File Path	E:\Aung's Thesis\Demo 2007\Alerts\Monitoring Engine\Event Generator\Contextual Information\ContextualInfo.mdb
Table Name	Employees
Attribute Name	Full Name
Custodian Attribute	Full Name

Entry of the ListCustodians data table

Outcome:

The analysis engine logs the event to the Alert database, so that the record can be verified by the list custodian of the category to which the new employee has been added. The alert generator utilised the information to identify the person responsible for verification of the record.



Outcome of the User Management Test Scenario 1

Test Scenario:

This scenario presents the addition of a new user to a role.

Event ID	10
Machine Name	PSQ_A304_WS3
User Name	Nathan Clarke
Server Name	PSQ_A304_FS1
File Path	E:\Aung's Thesis\Demo 2007\Alerts\Monitoring Engine\Event Generator\Contextual Information\ContextualInfo.mdb
Table Name	UserRoles
Record ID	4

Log Entry of the User Management Table

Record ID	User Name	Role Name
4	Luke Skywalker	NRG Researchers

Entry of the UserRoles data table

Role ID	1
Role Name	NRG Researchers
Department	NRG
Role Manager	Steven Furnell
Role Description	PhD students and network researchers

Entry of the Roles data table

List Name	UserRoles
Server Name	PSQ_A304_FS1
File Path	E:\Aung's Thesis\Demo 2007\Alerts\Monitoring Engine\Event Generator\Contextual Information\ContextualInfo.mdb
Table Name	UserRoles
Primary Key Attribute	RecordID
Record Name Attribute	UserName
Attribute Name	RoleName
List Description	List of users for each role

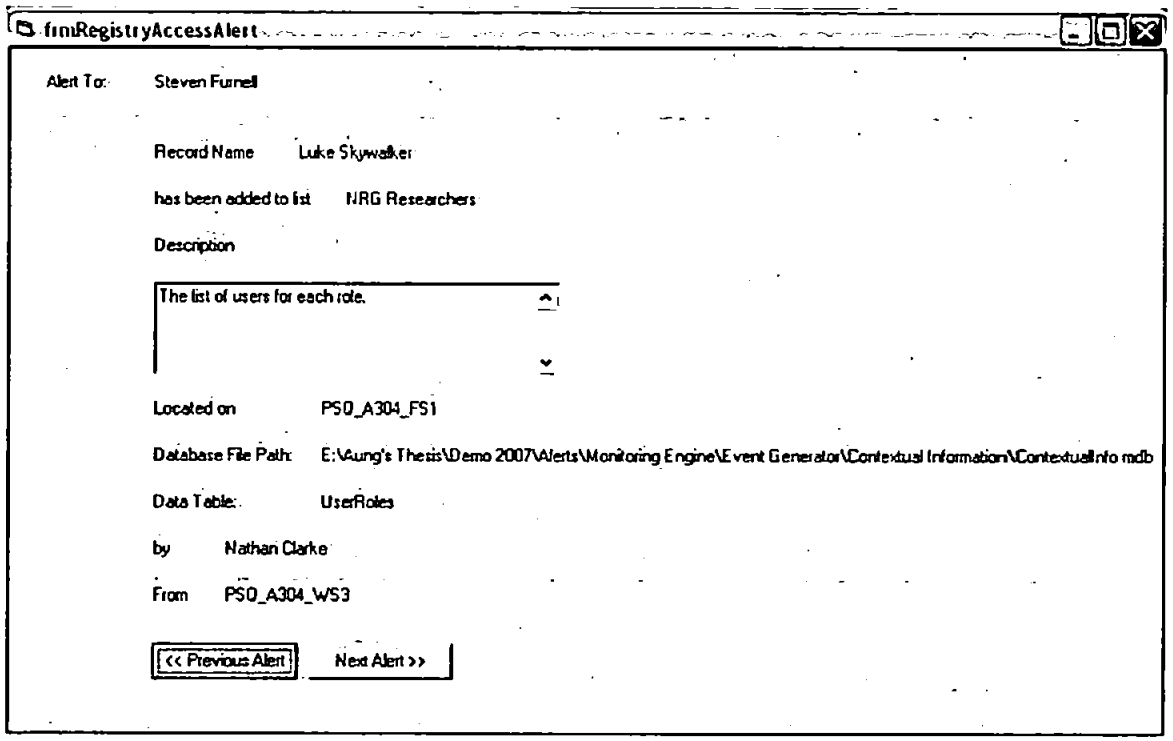
Entry of the Lists data table (indicates monitored registries)

List Name	User Roles
Server Name	PSQ_A304_FS1
File Path	E:\Aung's Thesis\Demo 2007\Alerts\Monitoring Engine\Event Generator\Contextual Information\ContextualInfo.mdb
Table Name	Roles
Attribute Name	Role Name
Custodian Attribute	Role Manager

Entry of the ListCustodians data table

Outcome:

The analysis engine logs the event to the Alert database, so that the record can be verified by the custodian of the category to which the new user has been added. The alert generator utilised the information to identify the person responsible for verification of the record.



Outcome of the User Management Test Scenario 2

Modification of System/Application Settings

The logic of the analysis procedure for test scenarios is described in chapter 7, section 7.6.

Test Scenario:

This scenario presents a user entering the settings with some of the required flags missing.

Event ID	31
Machine Name	PSQ_A304_WS1
User Name	Sevi
Application Name	Windows Firewall

Entry of the Settings data table within Events database

Event ID	Flag Description
31	Turn on firewall

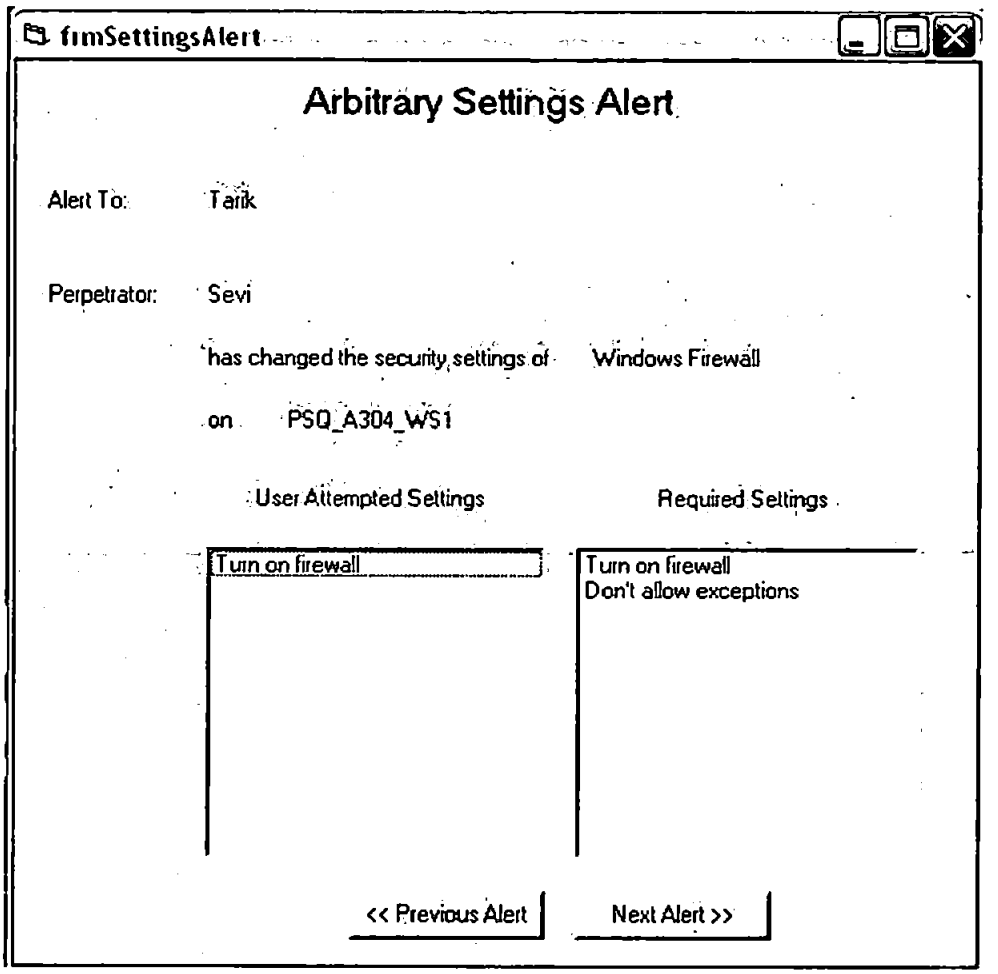
Entry of the Flags data table

Machine Name	Application Name	Flag Description
PSQ_A304_WS1	Windows Firewall	Turn on firewall
PSQ_A304_WS1	Windows Firewall	Don't allow exceptions

Entry of the Settings data table within the knowledgebase

Outcome:

The analysis engine logs the event to the Alert database, because "Don't allow exceptions" flag required by the policy is missing.



Outcome of the Arbitrary Settings Test Scenario 1

Test Scenario:

This scenario presents a user entering the settings with more flags than required by the policy.

Event ID	7
Machine Name	PSQ_A304_FS1
User Name	Shukor Razak
Application Name	Windows Firewall

Entry of the Settings data table within Events database

Event ID	Flag Description
7	Turn on firewall
7	FTP
7	Apache Web Server

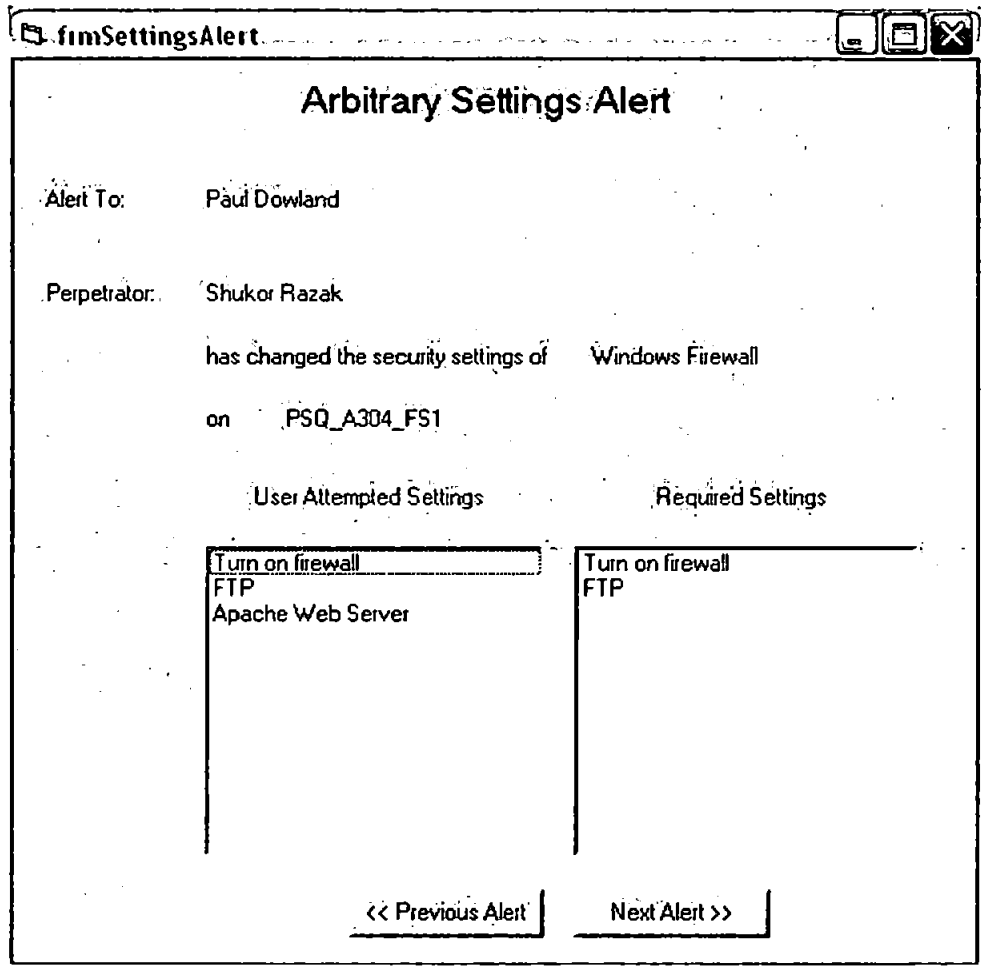
Entry of the Flags data table

Machine Name	Application Name	Flag Description
PSQ_A304_FS1	Windows Firewall	Turn on firewall
PSQ_A304_FS1	Windows Firewall	FTP

Entry of the Settings data table within the knowledgebase

Outcome:

The analysis engine logs the event to the Alert database, because one of the flags "Apache Web Server" is not required by the defined policy.



Outcome of the Arbitrary Settings Test Scenario 2

Various test carried out has validated the functionality of the prototype and its ability to detect various forms of misfeasance identified at the start of Chapter 8.

Appendix B – List of Publications

*Those marked with a * are included for reference. Some of the publications were edited and republished. Therefore, only the most relevant edition is included for reference.*

Furnell, S.M. and Phyo, A.H. (2002), "Watching your own: The problem of insider IT misuse", Proceedings of AiCE 2002 Third Australian Institute of Computer Ethics Conference, Sydney, Australia, 30 September 2002, pp17-24, 2002

Furnell, S.M. and Phyo, A.H. (2003), "Considering the Problem of Insider IT Misuse" Australian Journal of Information Systems, vol. 10, no. 2, pp134-138, 2003 *

Phyo, A.H. and Furnell, S.M. (2003), "Data Gathering for Insider Misuse Monitoring", Proceedings of the 2nd European Conference on Information Warfare and Security, Reading, UK, 30 June - 1 July, pp247-254, 2003 *

Phyo, A.H. and Furnell, S.M. (2004), "A Detection Oriented Classification of Insider IT Misuse", Proceedings of the 3rd Security Conference, Las Vegas, USA, 14-15 April, 2004 *

Phyo, A.H. and Furnell, S.M. (2004), "A Conceptual Framework for Monitoring Insider Misuse", Proceedings of Euromedia 2004, Hasselt, Belgium, 21-23 April, pp90-95, 2004 *

Phyo, A.H. Furnell, S.M. and Ifeachor, E. "A Framework for Monitoring Insider Misuse of IT Applications", Proceedings of the ISSA 2004 Enabling Tomorrow Conference, South Africa, 30 June-2 July, , 2004

Phyo, A.H. and Furnell, S.M. (2004), "A Framework for Role-Based Monitoring of Insider Misuse", Proceedings of IFIP/SEC 2004 - 18th International Conference on Information Security, Toulouse, France, 23-26 August, pp51-65, 2004

Phyo, A.H. Furnell, S.M. and Phippen, A. (2007), "Prerequisites for monitoring insider IT misuse", Proceedings of the Third Collaborative Research Symposium on Security, E-learning, Internet and Networking (SEIN 2007), Plymouth, UK, ISBN: 978-1-8410-2173-7, pp41-52, 2007 *

Considering the problem of insider IT misuse

Steven Furnell and Aung Htike Phyo

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

Email: sfurnell@network-research-group.com

Abstract

In recent years the Internet connection has become a frequent point of attack for most organisations. However, the loss due to insider misuse is far greater than the loss due to external abuse. This paper focuses on the problem of insider misuse, the scale of it, and how it has effected the organisations. The paper also discusses why access controls alone cannot be used to address the problem, and proceeds to consider how techniques currently associated with Intrusion Detection Systems can potentially be applied for insider misuse detection. General guidelines for countermeasures against insider misuse are also provided to protect data and systems.

Keywords: Insider misuse; misuse detection; misuse countermeasures.

Introduction

If one was to play a game of word association and use the terms 'security breach' or 'cybercrime' as the starting point, it is very likely that words like 'hacker' or 'virus' would be amongst the first responses. It is somewhat less likely that terms like 'employees' or 'insiders' would emerge as many peoples' first choices. In reality, however, insiders are very often the cause of the most significant and costly security incidents, and a significant proportion of what is commonly classed as cybercrime can be attributed to them. Indeed, the fact that insiders are already within the organisation often puts them in an ideal position to misuse a system if they are inclined to do so.

Although the great majority of the people are familiar with the generic meaning of the word 'misuse', when we try to map it to an IT context, there is a need to clarify certain issues. Insider IT misuse can be a very subjective term, and one of the most challenging tasks is to draw a clear line that separates an IT misuser from a person who is using a system in an acceptable way and for an approved purpose. 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. However, such a policy, and hence the definition of misuse, can differ from one organisation to the other. For example, where some would give priority to detecting data-theft and unsanctioned modification of data, others might want to detect denial of services and Internet access abuse. Thus no single definition of misuse is appropriate for all organisations.

The aims of this paper are to present evidence of the insider misuse problem, and suggest possible means by which it could be addressed. The discussion begins by examining the scale of the problem, based upon evidence from computer abuse surveys from recent years. This is followed by a more specific consideration of what can actually be considered to constitute IT

misuse in an organisational context, which then leads into a discussion of methods that could potentially be employed to combat the problem.

The scale of the insider misuse problem

If one takes a look back to computer crime literature and surveys dating up to the mid-90s, the evidence presented would certainly suggest that the main threat was to be found from one's own staff (with as much of 80% of computer crime believed to be the result of insider activity). In more recent years, however, many sources have indicated a significant rise in externally sourced incidents (principally in terms of Internet-based attacks such as website defacement and denial of service), with the consequence that although insider misuse is still significant, it now accounts for a far lesser proportion of raw incidents. For example, in the UK, results from the Department of Trade & Industry's Information Security Breaches Survey 2002 revealed that only 34% of businesses considered their worst security incident to have been caused by an insider (DTI 2002). This possibly accounts for why 60% of respondents in the same survey were either not very concerned or not at all concerned about threats originating from their own employees. However, when considering the large businesses (with over 250 employees) only, it should be noted that the split between those experiencing their worst incident as a result of internal staff versus external parties was almost equal.

Another source that has monitored the changing trend regarding internal and external attack is the annual CSI/FBI Computer Crime and Security Survey. Looking back to 1995, a key observation from the CSI was that "the greatest threat comes from inside your own organisation" (Power 1995). In more recent years, however, the survey results have painted a rather different picture, and by 2002 it was reported that, for the fifth year running, more respondents had cited their Internet connection as a frequent point of attack (74%), than had cited internal systems (33%) (Power 2002.). This may well be the case, but presenting the information in this manner tends to create something of a false impression, because the raw number of incidents is not necessarily the factor that we should be most concerned about. Of more interest to most CEOs, for example, will be the effect that the incidents had on their bottom line.

Many of the categories used in the CSI/FBI results encompass incidents that could potentially have been both internally and externally sourced (e.g. theft of proprietary information, sabotage of data networks, and virus). However, three of the categories very clearly indicate the source, and it is interesting to see the level of the annual losses that were associated in each case. The relevant information is presented in Table 1 (Power 2002).

	System penetration by outsider	Inside abuse of Net access	Unauthorized insider access
1998	$1,637,000	$3,720,000	$50,565,000
1999	$2,885,000	$7,576,000	$3,567,000
2000	$7,104,000	$27,984,740	$22,554,500
2001	$19,066,600	$35,001,650	$6,064,000
2002	$13,055,000	$50,099,000	$4,503,000
Total	**$43,747,600**	**$124,381,390**	**$87,253,500**

Table 1 : Annual losses for selected incidents from CSI/FBI surveys

It is quite evident from the results that, although they relate to a five-year period over which the proportion of externally sourced incidents had exceeded internal ones, the quantifiable losses in the latter case dwarf those attributable to outside hackers. It is therefore clear that, in real terms, the level of the insider threat is still much greater than that exhibited by external hackers.

The CSI figures relating to insider abuse of network access clearly suggest that, as well as bringing considerable advantages in terms of web and email communication, Internet access has also ushered in a whole range of new problems. This can be further evidenced by a survey of 544 human resources managers, conducted in 2002 and targeting large UK companies (with 'large' in this case being defined as those employing an average of 2,500 people). The results revealed that almost a quarter of them (23%) had felt obliged to dismiss employees in relation to Internet misconduct (with the vast majority of these cases – 69% - being linked to the downloading of pornographic materials) (Leyden, 2002). Many other cases resulted in less severe courses of action, such as verbal warnings or a discreet word in the ear of the person concerned, and in total the results indicated that 72% of respondents had encountered Internet misuse in some form.

The nature of insider IT misuse

One of the CSI/FBI categories from Table 1 was that of 'unauthorised insider access'. However, one of the complicating aspects with insiders, and the aspect that differentiates this from the other insider category listed in the table, 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. This means that he/she does not need to bypass the authentication mechanisms of the IT infrastructure (no stealing or illegal reproduction of passwords and other forms of authentication tokens). Thus, in an IT context, insider misuse is the act of abusing granted privileges to cause harm. In this context, it can also be observed that users that know more about a system are more likely to abuse their privileges than users who are less knowledgeable.

Although this is not difficult to grasp, vagueness is introduced by the term misuse and what it means to different people or organisations. What is considered illegitimate use in one particular organisation can be perfectly acceptable for another. For example, browsing the web for personal use is outlawed entirely in some companies, whereas others are somewhat more relaxed about it and impose varying limits upon what is acceptable (e.g. some may

permit up to 20 minutes per day, whereas others may allow twice this). In addition, there are myriad other activities that would likely be regarded as misuse in any organization, for example:

- Personal entertainment (e.g. playing games, writing personal email etc.)
- Downloading MP3s, pirated software, pornographic images, or other unsuitable material
- Fraud and theft (e.g. modifying payroll database to increase one's wages)
- Sending out inappropriate material using company computers
- Installing and using pirated software.
- Reading or modifying another user's files.

Although the computer security research community has created a plethora of taxonomies that describe computer intrusions in general (see Furnell et al. 2001 for an overview), little effort has been placed on the construction of a taxonomy that specialises in insider incidents. The earliest attempt to classify internal misuse of computer systems is presented by Anderson (1980) and discusses borders of distinction between *masqueraders*, *clandestine users*, and *misfeasors*. Masqueraders are insiders that exploit weaknesses of the authentication system, thus gaining the identity of other legitimate users. A clandestine user is related to authorised users and their capabilities to bypass audit, control and access resource mechanisms in a particular computer system. Finally, misfeasors are insiders who do not need to masquerade, but abuse the power of their privileges to misuse the system. However, as the small selection of examples above shows, the single category of 'misfeasor' can encompass a whole range of different incidents. As a result, other works have focused more specifically upon the issue of insider misuse, and indicative examples are given below:

- *Tuglular* (2000). This is the first comprehensive taxonomy of misfeasor incidents. The taxonomy classifies computer misuse incident in three dimensions: incidents, response and consequences. The entire taxonomy is orientated towards data collection for insider incident response.

- *Magklaras and Furnell* (2002). This taxonomy is human centric. Mgklaras and Furnell perceived that all actions that constitute IT misuse lead back to human factors. The fundamental aspect for their taxonomy is classifying people in three basic dimensions: system role, reason of misuse and system consequences. This scheme is the most appropriate for threat prediction, but not suitable for detection.

Intentional misfeasor cases are performed for a variety of reasons. The best way to sub-divide them is to consider the motives in a way that could detect the ultimate goal of the abuser. It might be inferred, for example, that a legitimate user is trying to access sensitive data (data theft), take revenge against a particular person or an entire organisation (personal differences), cover indications of unprofessional behaviour, or deliberately ignore a particular regulation of the information security policy.

Unfortunately, despite evidence of the insider threat, there is no substantial effort devoted to addressing the problem of internal IT misuse. In fact, the great majority of misuse countermeasures address forms of abuse originating from external factors (i.e. the perceived threat from hackers). A significant reason for this is the difficulty in actually monitoring and detecting the problem in order to enable a response to be mounted. In the cases above, for

example, it is clear that the misuse would have been very difficult to control or prevent, as the perpetrators concerned were not violating any system-side access rules.

Combating insider misuse

The problem with insider abuse is that, once a user is authenticated to use a system, what he does with the system or the objects he has access rights to is neither monitored nor logged most of the time. Considering the list of potential misuses in the previous section, it is possible that appropriate access controls could be used to prevent some of them, but even these will not be sufficient for all contexts (consider, for instance, the case in which the misfeasor has legitimately been granted administrator level privileges). This epitomizes the difficulty in implementing access controls that resembles organisational hierarchy onto the IT systems. It must also be remembered that one user/process/account having all the privileges can lead to serious misuse by exploiting the situation. Neumann's suggestion of multilevel systems and compartmentalization (Neumann 1999) should be given a serious consideration before we proceed with the insider misuse detection.

Today's commercial operating systems are based on the old systems developed years ago. At the time when the core components of these systems were developed, the users were expected to behave themselves. The problem of insider misuse was not an issue. However the research in the IT security over the years has proved that people do misbehave and that insider misuse is a serious problem. Since these systems were not developed with insider misuse in mind, the preventive mechanism and the logging present in today's commercial systems are not optimized for misuse detection. Existing access controls are not good enough to prevent insider misuse, making it more difficult to enforce insider misuse policies. For example, a user with administrator level privileges may not have the moral right to access confidential data on the system, but access controls present in today's systems cannot prevent such actions. As such, it is considered that some form of supervision system is required to monitor for misuse activity.

Such technologies are already available to some extent in the form of Intrusion Detection Systems (IDS) (Amoroso 1999), but as with many other mainstream security technologies, these are geared towards detecting attacks on the system rather than misuse of it by legitimate users. Nonetheless, some of the principles are transferable. For example, current IDS employ two main strategies to identify attacks namely misuse-based detection and anomaly-based detection, and it is possible to see how each of these could be applied to the insider problem.

- *Misuse-based detection*
  In a traditional IDS, 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 they can be anticipated to attempt based upon the privileges and resources available to them. For example, at a conceptual level, one such misuse signature might relate to a user who is identified as attempting to modify a record about him/herself in a database (e.g. the payroll example indicated earlier). The principle here would be that, although their database privileges may allow them to do so, users should probably not be modifying details relating to themselves without someone else's authority. Another

example could be to watch for any sequence of events where a user accesses confidential information and then attaches it in an email destined for a recipient outside the organization. Neither of these rules would necessarily cause the user in question to be locked out of the system (because in some contexts the actions could still be quite legitimate), but they could be used to flag the activity for closer scrutiny.

- *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 a 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. For example, past behaviour might suggest that a particular user typically downloads an average of 5MB of material from the web per week, and the nature of the attachments they assign to emails are normally documents. Therefore, if activity supervision detects a surge of download activity to 10MB in a single day, or a large number of email messages suddenly being sent with image attachments, then there would be reasonable grounds to investigate whether unsuitable activities might be in progress.

Although the above descriptions make the concepts sound relatively straightforward, it must be appreciated that neither technique can be considered 100% reliable, even in the context of traditional IDS. The consequence is that they can lead to false positives (where legitimate activity is believed to be intrusive) and false negatives (where genuine intrusive activities are misjudged as acceptable). The concept of applying the techniques for the detection of misfeasor activity / insider misuse makes the task more difficult, because we are dealing with legitimate users who are not violating access controls. From a misuse-based detection perspective, it is more difficult to identify the ways in which an insider might misuse the resources to which they have legitimate access, while from an anomaly detection perspective the level of behaviour profiling would need to be much more detailed and precise. When basing the assessment upon a comparison against their behaviour profile, a legitimate user misbehaving will almost certainly be more difficult to identify than a total impostor who is masquerading under the legitimate user's identity. In addition, in an adaptive system, the process of profile refinement might be exploited by wily misfeasors who gradually train the system to accept misuse behavior as normal. As such, this aspect is still an area of active research, as the technical approaches are not mature.

When considering how to protect systems now, it is worth noting that preventative measures need not be technical. Insider misuse is a management problem as much as it is a technical issue. As such, formal internal controls are as important as technical controls. Security guidelines, such as the recommendations provided by the ISO 17799 standard (BSI 2001), typically suggest a number of personnel-related measures, which if employed correctly could dramatically reduce the likelihood of insider misuse being successful:

- Check references of prospective new employees before hiring them;
- Ensure that employment contracts include a clause relating to the acceptable use of IT resources;
- Ensure that adequate reminders about the 'acceptable use' policy are encountered by staff during their day to day use of systems;

- Ensure adequate supervision of staff by line management;
- Provide a means by which staff can confidentially report misuse of IT systems, without fear of recrimination from colleagues.
- Ensure proper division of duties (i.e. such that collusion between staff members would be necessary before significant opportunities for frauds could be identified).
- Concerning the access of data, make sure that access control policies resemble organisation's management hierarchy or rules.
- Security and access control policies need to be maintained to keep up with the change in organisation's management hierarchy.

In the absence of an automated supervision approach, it would still fall to line managers and the like to enforce and monitor these aspects.

Conclusion

Insider misuse poses a great threat to organizations. Even though the Internet connection is the most frequent point of attack, the loss due to insider misuse is far greater than the loss due to external attacks.

At the present time, the system level countermeasures that can be implemented are limited. Current access control systems, although well-suited to guarding against unauthorized activities, cannot prevent insider misuse effectively if the subject is doing something within their legitimately assigned privileges. More advanced mechanisms, in terms of activity monitoring and supervision systems may offer a potential solution in the future. The authors' ongoing research will design and evaluate approaches for realizing the latter approaches, and results will be detailed in future publications.

References

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', 1980.

BSI. 2001. 'Information technology. Code of practice for information security management'. BS ISO/IEC 17799:2000. British Standards Institution, 15 February 2001. ISBN 0580369587

DTI. 2002. 'Information Security Breaches Survey 2002'. Department of Trade & Industry, April 2002. URN 02/318.

Furnell, S.M., Magklaras, G.B., Papadaki, M. and Dowland, P.S. (2001), 'A generic taxonomy for Intrusion Specification and Response', Proceedings of Euromedia 2001, Valencia, Spain, 18-20 April 2001: 125-131.

Leyden, J. 2002. 'P45s for Porn Surfers', The Register, 9 July 2002. http://www.theregister.co.uk/content/6/26098.html

Magklaras, G.B, Furnell, S.M, (2002). 'Insider Threat Prediction Tool: Evaluating the probability of IT misuse', Computers & Security, Vol. 21, No 1, pp62-73.

Neumann, P.G. 1999. 'The challenges of Insider Misuse', SRI Computer Science Laboratory, Paper prepared for the Workshop on Preventing, Detecting, and Responding to Malicious Insider Misuse, 16-18 August 1999, at RAND, Santa Monica, CA.

Power, R. 1995. 'Current and Future Danger: A CSI Primer on Computer Crime and Information Warfare', San Francisco, CA: Computer Security Institute.

Power, R. 2001. '2001 CSI/FBI Computer Crime and Security Survey', Computer Security Issues & Trends, vol. VII, no. 1. Computer Security Institute. Spring 2001.

Power, R. 2002. '2002 CSI/FBI Computer Crime and Security Survey', Computer Security Issues & Trends, vol. VIII, no. 1. Computer Security Institute. Spring 2002.

Tuglular, T. 2000. 'A preliminary Structural Approach to Insider Computer Misuse Incidents', EICAR 2000 Best Paper Proceedings: pp105-125.

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 role-based profile the perpetrator belongs to. Another advantage of role-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 relational databases (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.

A Detection-Oriented Classification of Insider IT Misuse

A.H.Phyo and S.M.Furnell

Network Research Group, School of Computing, Communications and Electronics,
University of Plymouth, Drake Circus, Plymouth, United Kingdom

email: nrg@plymouth.ac.uk

Abstract

Although the problem of insider misuse of IT systems is frequently recognised in the results of computer security surveys, it is less widely accounted for in organisational security practices and available countermeasures. Indeed, the opportunities for insider misuse, by perpetrators with legitimately assigned privileges, are often overlooked until an incident occurs. A possible reason for this is that the problem receives relatively little attention in the commonly recognised classifications of IT-related attackers and intrusions, with most focusing upon attacks and methods involving some form of system penetration and/or unauthorised access. This paper examines the potential forms of insider misuse in more detail, classifying them according to the level within in a target system at which the incidents could be detected. It is considered that such an approach could provide a relevant foundation in terms of subsequent approaches to automate insider misuse detection methods.

Introduction

Frequent headlines reporting hacker break-ins to computer networks and fast spreading computer viruses have steadily increased public awareness of the threats posed to information security. However, external hackers and malicious software are far from being the only threats to the security of an organisation. Survey results consistently show that insiders are very often the cause of the most significant and costly security incidents, and a significant proportion of what is commonly classed as cybercrime can be attributed to them. 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. In 2002 Information Security magazine survey, 23% of respondents rated authorized users as their most important problem, while 11% reported unauthorized users as their most important problem [Briney and Prince 2002]. Similarly, results from the Department of Trade & Industry's Information Security Breaches Survey 2002 revealed that 34% of businesses considered their worst security incident to have been caused by an insider [DTI 2002]. Indeed, the fact that insiders are already within the organisation often puts them in an ideal position to misuse a system if they are inclined to do so. The insider abuse can be more damaging than many outsider attacks, 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 [Einwechter 2002]. 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 how insider misuse incidents may be classified, giving particular attention to the points in the system at which different forms of misuse would be discernable. The discussion begins with a brief overview of existing approaches to classifying incidents and abusers, some of which already pay specific attention to the role of insiders. From this, the paper proceeds to propose a detection-oriented approach to classification, and discusses examples of the different forms of insider-sourced incident that would be detectable at network, operating system, application and data levels within the system.

A review of current intrusion taxonomies

In order to be able to focus on the misuses that may be committed by insiders of an organisation, it is important to understand the type and nature of all kinds of misuses. A number of previous investigations have therefore attempted to classify system attacks and abusers, in order to aid subsequent analysis. Some of these relevant works are summarised in the sections that follow, along with brief commentary in relation to their suitability for classifying incidents relating to insider misuse.

- **Cheswick-Bellovin Classification.** Cheswick and Bellovin have classified attacks into seven categories listed in the Table 1, which is drawn upon their work on firewalls [Cheswick and Bellovin 1994].

1. **Stealing passwords** – methods used to obtain other users' passwords
2. **Social engineering** – talking one's way into gaining information that one should not have
3. **Bugs and Backdoors** – taking advantage of systems that do not meet security specification, or replacing software with compromised versions
4. **Authentication failures** – defeating authentication mechanisms
5. **Protocol failures** – exploiting protocols that are improperly designed or implemented
6. **Information leakage** – utilising systems such as finger or the DNS to obtain information that is necessary for system administration and proper operation of the network, and abusing it
7. **Denial-of-service** – attempts to deny other users from being able to utilise systems and services

Table 1: Cheswick & Bellovin's seven categories of attacks

Although, this approach gives an overview of the attacks and classifies the main categories of attacks, it is too general and does not give an insight to the characteristics of attacks.

- **SRI Neumann-Parker Taxonomy**. The Neumann-Parker taxonomy is based on incidents reported over 20 years [Neumann and Parker 1989]. It classifies

intrusions into nine categories, which describe the nature of the attacks. Table 2 summarises the overall scheme of the taxonomy.

NP1 External Misuse	Non-technical, physically separate intrusions
NP2 Hardware Misuse	Passive or active hardware security problems
NP3 Masquerading	Spoofs and identity changes
NP4 Subsequent Misuse	Setting up intrusions via plants, bugs
NP5 Control Bypass	Going around authorised protection/controls
NP6 Active Resource Misuse	Unauthorised modification of resources
NP7 Passive Resource Misuse	Unauthorised reading of resources
NP8 Misuse Via Inaction	Neglect or failure to protect a resource
NP9 Indirect Aid	Planning tools for misuse

Table 2: SRI Neumann-Parker Taxonomy

- **Lindqvist–Jonsson Taxonomy**. The results gathered from laboratory experiments have indicated the need for further subdivision of Neumann-Parker taxonomy. This scheme is an extension of Neumann-Parker taxonomy. In this taxonomy the security incidents are viewed from the perspective of the system owner, and categories NP5, NP6, and NP7 of Neumann-Parker taxonomy are further classified [Lindqvist and Jonsson 1997].

Extended NP5 Control Bypass	Password attacks, spoofing privileged programs, utilising weak authentication
Extended NP6 Active Resource Misuse	Exploitation of write permissions, resource exhaustion
Extended NP7 Passive Resource Misuse	Manual browsing, automated browsing

Table 3: Lindqvist-Jonsson extension of SRI Neumann-Parker Taxonomy

Although (extended) NP6 and NP7 above do at least recognise the misuse issue, the rest represent the attack methods employed by outsiders, or insiders who utilises the same methods. In addition, the classification of attacks is based on the misuse techniques employed and the consequences of it, and it is not intended for monitoring purposes. However, some other works can also be identified that contain elements more specifically related to insider misuse.

- **Anderson's Taxonomy.** Anderson's early work in this domain classifies system abusers into External Penetrators, Internal Penetrators, and Misfeasors, as shown in Table 4 [Anderson 1980].

Abuser Type	Description
External Penetrators	Outsiders attempting or gaining unauthorised access to the system.

Internal Penetrators	Authorised users of the system who access data, resources or programs to which they are not entitled. Sub-categorised into : • *Masqueraders* Users who operate under the identity of another user. • *Clandestine users* Users who evade access controls and auditing.
Misfeasors	Users who are authorised to use the system and resources accessed, but misuse their privileges.

Table 4 : Categories of system abuser

Although very useful at a broad conceptual level, the classification does not provide any significant assistance in terms of incident detection, with all insider misuse related incidents being grouped under the single 'misfeasor' heading.

- **Tuglular's Taxonomy**. This is the first comprehensive taxonomy of misfeasor incidents [Tuglular 2000]. The taxonomy classifies computer misuse incident in three dimensions: incidents, response and consequences. The Incidents dimension is further classified into target, subject, method, place, and time sub-dimensions. The Response dimension is divided into recognition, trace, indication, and suspect. The Consequences dimension includes disruption, loss, effect, violation, misuse type, misuse act, and result. The sub-dimensions branches into new branches of sub-dimensions and so on until it cannot be further classified. These dimension and sub-dimensions of the scheme are used to characterise each misuse incident. However, the entire taxonomy is orientated towards systematic data collection of insider incidents to provide evidence and incident response.

- **Magklaras-Furnell's Insider Threat Prediction Model**
  This model is human centric, and the authors argue that all actions that constitute IT misuse lead back to human factors. The fundamental aspect for the taxonomy is classifying people in three basic dimensions: system role, reason of misuse and system consequences [Magklaras and Furnell 2002]. However, while this scheme is intended to assist threat prediction, it is not suitable for the detection of insider misuse.

None of the previously mentioned taxonomies are oriented towards detection of insider IT misuse, in terms of considering how we would approach the task of monitoring activities to determine where problems may be apparent. A potential approach to this issue is considered in the remainder of the paper.

A detection-oriented approach to classification

In determining a means to link classification to the method of detection, it is considered appropriate to classify insider misuses based on the level of the system at which they might be detected. The basis for this is that different types of misuses manifest themselves at varying levels of the system (e.g. some may be apparent at the

network level, whereas others are most visible at higher levels, such as the operating system or application levels).

With this form of classification in mind, the concept can be illustrated using a variety of recognised insider misuse activities, and then considering the different levels at which they may be detected. An overall classification is presented in Table 5, and then examples of the incidents concerned are considered in the sub-sections that follow. These consider what could be monitored, and how this could be used to detect, control and restrict misuse-related behaviour.

Misuse	Monitoring Level	Attribute(s) to monitor
Illegal content	Network	Packet Content
Excessive/anomalous usage	Network	Bandwidth Usage
Resource exhaustion	Network	Bandwidth Usage
Playing Network Games	Network	Bandwidth Usage
Illegal software distribution	Network	Bandwidth Usage
Access to isolated subnets and machines	Network	IP Address
Access from unauthorised machines	Network	IP Address
Access to prohibited content	Network	URL
Use of web-based email	Network	URL
Recreational surfing	Network	URL
Instant Messenger	Network	Service Usage
Unauthorised network services	Network	Service Usage
File Sharing	Network, OS	Service/Bandwidth Usage, File Attributes
Web Hosting	Network	Service/Bandwidth Usage
Resource exhaustion	OS	CPU, Memory, Disk Usage
Storage of Image and Multimedia files	OS	File Extensions
Anomalous Command usage	OS	Command Usage
Anomalous Application usage	OS	Application Usage
Information Disclosure	OS	File (read)
Breach of Privacy	OS	File (read)
Data theft	OS	File (read, copy)
Alteration of Data Files	OS	File (write)
Alteration of System Files	OS	File (write)
Hardware Installation	OS	File (create, write) configuration files
Software Installation	OS	File (execute) unauthorised program
Illegal program execution	OS	File (execute) unauthorised program
Sabotage	OS	File (write, delete)
Privileged Program Exploits	OS	System Calls, (File, Memory) Access, I/O Usage
Data Hiding	OS	Input Files to Programs
Encryption	OS	Input Files to Programs
Program Exploitation	Application	User (Input, Interaction)
Alteration of Input	Application	Function Usage
Function Usage	Application	Quereis, API Calls, Windows Messages
Anomalous Database Access	Application	User Queries
Inconsistent Data (Fraud)	Data	Batch Numbers, Date, Time, Strings, Numbers

Duplicate Entries (Fraud)	Data	Batch Number, Uniquely Identifiable Entities, etc.
Maximum Value	Data	Number of Employees, Bonuses, Extratime work, etc.
Minimum Value	Data	Hourly pay rate, Work hours, etc.

Table 5: Insider-Oriented Misuse Classification

Network-level misuses

Given that a great deal of misfeasor activity may relate to the use of network services, several type of misuse would be detectable by monitoring by monitoring activity at the network traffic level. From a practical perspective, this has the advantage that there is no specific necessity to install monitoring / data collection agents on individual end-user systems. Examples of the misuses that could be identified are discussed below.

- *Access of prohibited content*: User access of prohibited content on the web may be monitored through logging and examination of web addresses accessed. Accessed web addresses may be checked against a database of websites containing inappropriate content, such as pornographic material. Another approach would be to create a database of websites that the employees may access to perform their day-to-day tasks, then user accessed websites can be compared against the entries. It is not necessary to block the access to the websites that are not in the database; therefore access is not restricted, but monitored. The latter approach is more desirable if the organisations want to discourage recreational surfing.

- *Downloading inappropriate material*: File extensions of the users' network download can be monitored. For example, a user downloading files with image extensions may be downloading pornographic material. Other file extensions that should be monitored include ".mpeg", ".avi", ".mp3", and ".zip" files. Ideally, download rights should be limited to a few users as any type of downloaded material may introduce viruses into the organisation's networked systems. Downloading of large files can also consume valuable bandwidth and delay legitimate work.

- *Use of web-based email*: Many organisations disapprove the use of web-based email, because of the difficulties in monitoring usage. Employees may be wasting work hours by sending personal emails through the use of web-based email, especially when the users' email accounts in the organisation are being monitored for usage. User accessed web addresses may be checked against a database of known web-based email sites.

- *Online shopping*: Users may be wasting valuable work hours by shopping online. User accessed websites may be checked against a database of online shopping websites.

- *Spamming*: Users sending more than normal amount of emails may be spamming using company computers. On the other hand, the user's email client might be infected with a worm that mails itself to everyone in the user's contact list. Whatever the case, a closer examination is required, when exceeding number of emails are sent from users.

- *Using chat programs*: Employee utilisation of chat programs such as IRC, ICQ, and instant messengers can affect the productivity of the users. Chat programs can also affect the security of the network as they introduce new services and those services may be exploited. Network services utilised by users can be monitored to look out for users using chat programs.

- *Video Conferencing*: Users may be video conferencing with friends or relatives using organisation's computing resources. Network service utilisation and bandwidth usage may be monitored to detect such abuse.

- *Playing network games*: Employees may be playing games on the organisation's local area network. Such activity may consume precious bandwidth. This kind of activity may be monitored through looking out for users with exceedingly high bandwidth consumption.

- *Running servers*: Users may be running personal web-servers from the company network. The motivation of such activity may be for financial gain or for mischievous purposes such as distribution of illegal software.

- *Peer-peer file sharing:* Users utilising file sharing programs may be downloading and sharing inappropriate materials with other internet users. Network service utilisation can be monitored to detect such abuses.

- *Access of isolated sub-networks*: Users accessing sub-networks that are not related to their domain may be suspicious. For example, a software developer establishing a direct connection to the payroll sub-net may have undesirable intentions such as modifying the payroll database to raise one's own wages. Cross network connections may be monitored to detect the access of isolated networks.

Having stated the possible monitoring opportunities for insider misuse at the network level, we should consider the following statement by Schultz [2002], "Insiders do not generally demonstrate the same attack signatures as external attackers". Indeed, 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 exploit the network-level 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.

System-Level misuses

In contrast to detecting network-level incidents, monitoring at the system level necessitates that monitoring activity be conducted upon individual host systems (i.e.

some form of data collection agent would need to be present on the user system). If such monitoring is available, then the following list constitutes some examples of the types of incident that could be identified.

- *Storing inappropriate materials*: Users may be storing inappropriate materials on organisation's computers. For example, users may be storing MP3s, movies, illegal software, and pornographic materials. Users' home directories may be scanned to detect files with certain extensions, such as ".jpeg" to detect the content stored. For example a user having a large number of image/media files may be storing inappropriate materials on the computer. User disk usage may also be monitored for excessive usage. Monitoring excessive disk usage may sometimes lead to the detection of illegal software being stored on company computers.

- *Use of data-hiding programs*: Users may be utilising data-hiding programs, such as steganographic software to hide inappropriate material. Such programs may also be used to disguise proprietary and confidential information before they can be sent out of the organisation. Programs that take file(s) as inputs and produce file(s) outputs should be examined to make sure they are not data-hiding programs, such as encryption and steganographic software.

- *Use of arbitrary programs*: Users may run arbitrary programs to access data. Sometimes when data is accessed through the use of arbitrary programs, application level access controls and auditing may be bypassed. Program executions may be checked against a database of authorised programs. This would require a database of authorised programs along with file check sums to guarantee integrity of the program being executed.

- *Modifying system configuration*: Users may be modifying system configuration files, which may affect the way the system and programs behave; such modifications are undesirable as the system may become insecure as a consequence. Monitoring access to vital system and application configuration files can lead to the detection of such abuse. This would require a database of critical configuration files and their check sums.

- *Adding unauthorised hardware*: Adding additional hardware, such as modems can affect the systems' security. For example, the user's communications through the modem will not be picked up by network intrusion detection systems, and the user may be sending confidential information out of the organisation. Addition of unauthorised hardware can be detected by monitoring system settings and configurations.

- *Output redirection*: Output from applications may be redirected to undesired destinations (files, networks, or machines). The output from certain applications may contain confidential information, which should only be sent to appropriate destinations. For example, backup process sending the backup data to a different machine than usual. In this example, the backup operator may be attempting to get proprietary information out of the company. Output destinations of applications processing important information can be profiled to detect anomalous output destinations.

- *Alteration of audit data*: Users may be altering audit and system accounting file to cover up traces of system abuse. Log files and audit trails should not be modified even by the system administrator, because they contain evidential information regarding system abuses. Modification of log files can be monitored to detect users destroying evidential information.

- *Breach of Privacy*: Users may be accessing other users' files. The perpetrator may be someone with high system privileges or configuration errors may have made the file world readable. This type of incidents can be detected by monitoring users browsing files/directories own by others.

- *Batch Deletion*: Users or processes deleting a large number of files may sometimes represent sabotage of system or data. Therefore, users or processes deleting a batch of files can be monitored to detect possible sabotage of system and data. Managerial controls such as separation of duties should also be applied to deletion of files in work folders. For example, a user can be assigned the job of actually deleting the files, while users can mark files that should be deleted.

- *Installation of unauthorised software*: Every software program installed is a link in the security chain of the organisation. The newly installed program may introduce a new vulnerability through which the system may be exploited. The installed program may be a Trojan or viral infected software. In general software installation rights should be limited to a couple of users and programs should be authorised before installed on organisation's systems. In order to accommodate this, a list of executable directories needs to be established, and only the authorised programs stored in these directories may be executed. A database of authorised programs with associated check sums is also required. With this approach, users executing unauthorised programs or executing programs from arbitrary directories, such as home or temporary directories can be detected.

- *Copying software programs*: Users may copy customised software programs used in organisation's computers. For example, users can copy executable files, shared library files, and registry entries of a proprietary program for malicious purposes. Users accessing executable files in "Read" mode can be monitored to detect copying of executable programs.

- *Excessive Printing*: Users may be abusing organisation's printer facilities, for personal use and private work. Excessive usage of print services may be monitored to detect this type of abuse.

- *Input to programs*: Files containing confidential data may be passed to encryption/steganographic programs as input. Monitoring input to encryption/steganographic programs can detect users attempting to disguise information before sneaking it out of the organisation. This would require a list of encryption/steganographic programs installed on the system. Then the file inputs to such programs can be checked if they are important confidential files.

It is clear from the above that system-level monitoring gives the potential for a far wider range of misuse activities to be identified. However, some types of abuse will be distinguishable from normal activity only with the knowledge of application-level semantics, and consequently may not exhibit malicious behaviour at the system level. Therefore, to be fully comprehensive, some detection strategies will be necessary at the application and database levels.

Application and data-level misuses

Monitoring at this level must again be focused upon individual host systems, but now at a deeper level, collecting data from within individual applications that might attract misfeasor interest. The list below presents some examples of the general forms that misuse at this level might take.

- *Inappropriate inputs*: Users may type in inappropriate inputs into the applications. Inappropriate inputs can cause the application to crash, behave in an unexpected manner, or result in compromised integrity of the data. Entering a different type/format of data to the type/format expected by the application can result in the application misbehaving and disintegration of processed data. Entering a different range of data can result in fraud. User input could be monitored at the interface level where the users interact with the application. In a client server environment, user inputs/request (server messages) may also be monitored at the server side.

- *Anomalous access of databases*: Anomalous access of databases can result in disclosure of confidential information and fraud. Insiders may misuse databases containing medical records, criminal records, customer data, personal records, and statistical information relating to businesses. Query requests by users may be monitored to detect anomalous access of databases.

- *Function usage*: Commercial off-the-shelf applications include many features some of which are not easily disabled, and usage of certain functions may result in disclosure of information or compromised data integrity. Monitoring of access to subroutines, function calls, and API calls can detect user access of features and application functions.

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 monitoring misuse in database and transaction systems, it is conjectured 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. Although, potential occurrence of fraud may be detected by monitoring for violation of separation of duties, the actual occurrence of fraud can only be detected by analysing the application data itself.

Conclusions

Existing intrusion taxonomies mainly describe characteristics of various attacks, and not developed specifically for monitoring insider misuse. Anderson was the first person to classify different types of insiders who misuse the IT systems into, masqueraders, clandestine users, and misfeasors. However, these classifications only characterise the type of users and not the actual misuse or how they may be detected. Tuglular produced the first comprehensive taxonomy of insider misuses. However, Tuglular's taxonomy is primarily aimed for systematic data collection of insider incidents to provide evidence and incident response. The authors have presented a classification of insider IT misuses based upon the level(s) of the system each type of incident may be detected or monitored. Internet abuse may be detected at the Network level, while data theft, sabotage, resource exhaustion, process behaviour, and system modification may be detected at the OS level. Anomalous user interaction with the application, anomalous access of databases, and breach of separation of duties may be monitored at the application level. Although, potential occurrence of fraud may be detected at the application level by monitoring violation of separation of duties, the actual occurrence of fraud may only be detected by analysis of the data.

The authors are using this work to contribute towards the realisation of an active insider misuse monitoring system. An accompanying conceptual architecture has already been specified (Phyo and Furnell, 2004), and work is proceeding towards practical implementation and validation.

References

Anderson, J.P. (1980), 'Computer Security Threat Monitoring and Surveillance', Technical Report, James P Anderson Co., Fort Washington, April 1980.

Briney, A. Prince, F. (2002), 'ISM Survey 2002', Information Security Magazine. http://infosecuritymag.techtarget.com/2002/sep/2002survey.pdf, September, 2002.

Cheswick, W.R. and Bellovin, S.M. (1994), 'Firewalls and Internet Security: Repelling the Wily Hacker', Addison-Wesley Publishing Company, 1994.

DTI. (2002). 'Information Security Breaches Survey 2002'. Department of Trade & Industry, April 2002. URN 02/318.

Einwechter, N. (2002),'Preventing and Detecting Insider Attacks Using IDS', http://www.securityfocus.com/infocus/1558, March 20, 2002.

Lindqvist, U. and Jonsson, E. (1997), 'How to systematically Classify Computer Security Intrusions', In the Proceedings of the 1997 IEEE Symposium on Security and Privacy, IEEE Computer Society Press, May 4-7, 1997.

Magklaras, G.B, Furnell, S.M, (2002). 'Insider Threat Prediction Tool: Evaluating the probability of IT misuse', Computers & Security, Vol. 21, No 1, pp62-73.

Neumann P.G. and Parker, D.B (1989), 'A summary of computer misuse techniques', In the Proceedings of the 12th National Computer Security Conference, Baltimore, USA, 10-13 October, 1989, pp. 396-407.

Phyo, A.H. and Furnell, S.M. (2004), 'A Conceptual Framework for Monitoring Insider Misuse', submitted to The Second International Workshop on Security in Information Systems (WOSIS 2004), Porto-Portugal, April 13.

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: pp105-125.

A CONCEPTUAL FRAMEWORK FOR MONITORING INSIDER MISUSE

Aung Htike Phyo and Steven Furnell

Network Research Group, School of Computing, Communication and Electronic Engineering, University of Plymouth, Plymouth, United Kingdom
e-mail: nrg@plymouth.ac.uk
Web: http://www.plymouth.ac.uk/nrg

KEYWORDS

Intrusion Detection Systems, Insider Misuse, Role-based Monitoring.

ABSTRACT

Traditional Intrusion Detection Systems are ineffective in detecting users who abuse their legitimate privileges at the application level, because they do not have the knowledge of application level semantics, required separation of duties, and normal working scope. This paper outlines a novel framework for solving the problem of insider misuse monitoring. The approach argues that users with similar roles and responsibilities will exhibit similar behaviour within the system, enabling any activity that deviates from the normal profile to be flagged for further examination. The system utilises established role management principles for defining user roles, and the relationships between them, and proposes a misuse monitoring agent that will police application-level activities for signs of unauthorised behaviour.

INTRODUCTION

Many security incidents involve legitimate users who misuse their existing privileges, such that they have the system rights to perform an action, but not the moral right to do so. Current IDSs focus upon detecting problems such as network penetrations, access violations and privilege escalations. These tools are currently geared towards detecting attacks by outsiders, as well as insiders who employ the same methods to mount an attack. However, insiders may not need to exploit the systems because they already have legitimate access to it, and many incidents involve insiders only abusing their existing privileges (Audit Commission 1990), due to lack of separation of duties and application level control. Additionally current IDSs do not have knowledge of the normal working scope of a user for a relevant position and the separation of duties that should be enforced. Therefore, there is a need to provide the detection system with knowledge of organisation hierarchy and role-relationships in order to enable more effective monitoring. Role Based Access Controls (RBAC) (Ferraiolo and Kuhn 1992, Sandhu et al. 1996) utilises knowledge of role-hierarchy and role-relationship to make access decisions. This paper presents a novel framework that uses established role management principles used in RBAC to provide knowledge of organisation hierarchy and business process to the detection system. The next section briefly examines the nature of the insider misuse problem, leading into a discussion of the degree to which the detection strategies employed by traditional IDSs may be applicable. This section also introduces the potential for incorporating role based access controls, and the importance of role-relationship management. These ideas are then combined with the proposal for a novel framework for insider misuse detection.

THE PROBLEM OF INSIDER IT MISUSE

Insider misuse refers to users who have legitimate access to the IT systems and the data stored upon it, but abuse their privileges by using the resources in an inappropriate manner or for an unapproved purpose. Anderson (1980) classifies such users as 'misfeasors'. Computer crime surveys certainly suggest that one's own staff are a significant threat, with the results of recent surveys (Power 2002, Richardson 2003) by the Computer Security Institute (CSI) consistently suggesting that the dollar amount lost due to insider abuse is far greater than that of outsider attacks (e.g. the total losses over the last 6 years that were clearly attributable to outsiders were $46.5m, whereas the costs of insider misuse exceeded $220m).

Opportunities for insider misuse are many and varied (Phyo and Furnell, 2004), it is possible that appropriate use of traditional access controls could be used to prevent some of them. However, these will not be sufficient for all contexts (consider, for instance, the case in which the misfeasor has legitimate access to the payroll database, but modifies records to raise his own salary). One of the problems with insider abuse is that what users do with the system, or objects to which they are granted access rights, is neither monitored nor comprehensively logged most of the time. Different types of misuses can manifest themselves at varying levels of a system. Network access violations will show up at the network level, file access violations and application usage will be evident at the operating system (OS) level, whilst the user behaviour within the application environment will be most evident at the application level. Therefore it is important to collect the data for misfeasor analysis at the appropriate level in

order to increase the relevance of the collected data. The previous payroll example epitomises the case where data collected at the application level would provide more information about the user's intentions, when compared with the data collected at either the network level or the OS level.

Current IDSs are ineffective in detecting misuse of existing privileges. Access here might be just a simple read operation or modifying a database entry. Again, the users may access the resource in an unacceptable manner or for an unapproved purpose. Insider misuse is not only a technical problem, but also a managerial problem, because in some cases it is the improper segregation of duties that presented the opportunity to misuse (Audit Commission 1990). Therefore, in order to effectively monitor misfeasor activity, the monitoring system needs to have the knowledge of application level semantics, organization structure, separation of duties and user responsibilities. Coupled with this knowledge and monitoring at relevant levels of the system, a more effective system for detecting abuse of existing privileges may be designed.

APPLYING IDS TECHNIQUES TO INSIDER MISUSE

Traditional IDS employ two main strategies to identify attacks, namely misuse-based and anomaly-based detection (Amoroso 1999), and it is possible to see how each of these could be applied to the insider problem.

- *Misuse-based detection*: This approach relies upon knowing or predicting the intrusion 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. For example, at a conceptual level, one such misuse signature might relate to a user who is identified as attempting to modify a record about him/her in a database (e.g. the payroll example indicated earlier). The rule here is that no one should modify their own records without someone else's authorisation. The problem with applying misuse-based detection to insider misuse is that the possible misuse scenarios for insiders are wide ranging and could be extremely organisation-specific.

- *Anomaly-based detection*: This approach relies upon watching out for things that do not look normal when compared to typical user activities 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 behaviour that has been established over time. One advantage insider misuse detection system has over outsider attacks is that it is possible to characterise normal activities of insiders according to their job position, as users with the same responsibilities should exhibit similar activities within the system and application environment to complete their daily tasks. The similarities may be profiled to represent normal behaviour for users with the same responsibilities, and different profiles for different job positions. If the user's behaviour deviates from the normal profile that represents his position, the activity should be flagged as suspicious. For example, a user who accesses a critical information system far more frequently than the other users within the same role may be browsing the database for personal gain.

Another problem associated in insider misuse detection is that current IDSs lack the necessary knowledge of business processes, organisation hierarchy, separation of duties, and the role of the users within the organisation structure. This knowledge needs to be expressed in the form that is understandable to the IDS, if effective misfeasor monitoring is to take place. Role management principles specified by Gavrila (Gavrila, and Barkley 1998) are utilised in Role-Based Access Control (RBAC) to support user role assignment, role relationships, constraints and assignable privileges. A role can be thought as a collection of operations required to complete the daily tasks of a user. In RBAC operations are associated with roles and the users are assigned to appropriate roles. This approach simplifies the task of assigning permissions to the user, as the roles for appropriate job functions are created with the least privileges required to complete the relevant tasks and the users are assigned to the role that reflects their responsibilities. Users can be assigned from one role to another, or assigned multiple roles, and permissions can be assigned at role-level to affect all users associated with the role. The type of operations and objects that can be controlled by RBAC is dependant upon the environment and the level at which it has been implemented. For example, at the OS level, RBAC may be able to control read, write, and execute; within database management systems controlled operations may include insert, delete, append, and update; within transaction management systems, operations would take the form that exhibit all properties of a transaction. The term transaction here

means a combination of operation and the data item affected by the operation. Therefore, a transaction can be thought of as an operation performed on a set of associated data items. The ability to control specific transactions, rather than restricting simple read and write operations are very important in database environments. For example, a clerk may be able to initiate a transaction and the supervisor may be able to correct the completed transactions, for which both users need read and write access to the same fields in the transaction file. However, the actual procedures for the operations and the values entered may be different Meanwhile, the clerk may not be allowed to correct the completed transactions and the supervisor may not be allowed to initiate the transactions. The problem is that determining whether the data has been modified in the authorised manner, for it can be as complex as the actual transaction that modified the data. Therefore, transactions need to be certified and classified before associating them with the roles. To characterise the required transactions for a role, duties and responsibilities of the users need to be specified first.

In RBAC separation of duties can be applied by specifying mutually exclusive roles. In the RBAC framework administrators can regulate who can perform what actions, when, from where, in what order and sometimes under what circumstances. Access controls only allow or deny access to certain resources, however there is a need to monitor and analyse the user actions after the access has been gained and the operations had been carried out. In theory the idea of roles and role-management principles can be applied to misfeasor monitoring. Instead of allowing or denying operations to be performed, common user operations can be associated with roles, and the users can be assigned to appropriate roles. If the user's operations deviate from the common profile, a thorough investigation can be carried out to clarify if the user has misused the system in an inappropriate manner or for unapproved purpose.

MISFEASOR MONITORING SYSTEM: ARCHITECTURAL CONSIDERATION

It has been mentioned previously that anomaly detection is more suitable for insider misuse detection, because employees' normal behaviour can be profiled. For example, previous work in the DIDAFIT system (Low et al. 2002) has profiled database transactions by generating fingerprints for authorised SQL queries, along with variables that the users should not change, ensuring that the queries are executed in the expected order and only on the restricted range of records. It is assumed that the users with the same responsibilities within the organisation will exhibit similar activities within the system, and their working-scopes may be established. The idea of establishing working-scopes for users with same responsibilities has been tested in relational database environments by Chung et al (Chung et al. 1999). However, many of the insider misuse cases in Audit Commission (1990) surveys are a result of lack of separation of duties and application level controls. In order to be able to detect violation of separation of duties, the detection system needs to be provided with the knowledge of organisation hierarchy and relationships between roles. RBAC utilises role-relationship management principles to define role-hierarchy and separation of duties. The authors' proposed system aims to combine the ability of RBAC to provide knowledge of role-relationships with intrusion detection techniques to effectively detect users who abuse their existing privileges. Figure 1 presents the framework of the conceptual insider misuse detection system. Functional modules are explained in subsequent paragraphs.

Management Functions

All management functions, such as defining roles, characterisation of operations, association of operations to roles and user assignment to roles, are carried out from the Management Console. The working scope of a user is defined by the operations associated with the role(s) the user assumes. Once the separation of duties between roles has been defined, it is expressed in the Role-Relations Matrix, such as inheritance, static separation of duties, and dynamic separation of duties. Static separation of duties occurs at the role level by specifying mutually exclusive roles. When the two roles are in static separation of duties, a user may not be assigned both roles. Dynamic separation of duties occurs at the operations level and the conditions can be that operations within dynamically separated roles are:

- Mutually excluded
- Disallowed to execute concurrently
- Disallowed to perform both operations on the same set of data

When the two roles are in dynamic separation of duties, the user may not execute the operations that are mutually exclusive or on the same set of data. The relationships expressed in the Role-Relations Matrix are checked against the rules specified by (Gavrila, and Barkley 1998) for consistency.

Host

This is where the actual profiling of user(s) and the detection process takes place. Characteristics of each operation are stored in the *Operations DB* along with an appropriate name for each operation. The characteristics are dependent upon which level of the system they are being profiled at. Characteristics of the operations may be in the form of file access, sequence of system calls, SQL queries, API calls, User interactions, and Network access.

Recording the characteristics of each operation is controlled from the *Management Console*. The profiling should be done at all three levels of the system namely: network, system, and application level. At the network level, roles should be profiled based on the essential access to subnets in order for the users of the role to complete their daily tasks. At the system level, roles should be profiled on the use of applications required to complete the tasks. It should also be established which machines the users of the role can/cannot perform the task from. Again, at the system level, roles should be profiled based on what files need to be accessed in order to complete the task, along with the access mode and the application/process from which the files are accessed. Once the user has gained access to the file, and if the file is accessed from an application in which the file can be modified or manipulated (e.g. Databases), the application level monitoring should commence. At the database level, user queries and the associated values should be monitored. The problem is that determining whether the data has been modified in the authorised manner, for it can be as complex as the actual transaction that modified the data. Therefore, transactions need to be certified and classified before associating them with the roles. The *Detection Engine* then checks the roles available to the active user, and next checks the *RoleOperations* table for the names of the operations available to the user. After which the characteristics of the available operations from the *Operations DB* are compared to the current user actions. If current user actions do not match the characteristics of operations available to the user, the administrator is alerted. This alert may indicate the user performing a totally new operation, or performing a valid operation in the *Operation DB* but is violating separation of duties because the operation is not listed under any roles the user may assume.

The envisaged detection flow is as follows:

1. Detection Engine gets the name of the user from the Client. Looks for the roles the user's name is associated with, in the Role-User table.
2. After acquiring the list of roles for the user, the Detection Engine looks for the names of the operations associated with each role in the Operations DB. (Note: only names of the operations are associated with the Roles.)
3. After acquiring the names of operations available to the user, the Detection Engine reads the characteristics of available operations from the Operations DB and they are compared against current user actions.
4. If the current user action matches with the characteristics of operations available to the user, then the user is not in breach of static separation of duties.
5. If OpA belongs to RoleA, OpB belongs to RoleB, and RoleA and RoleB are in dynamic separation of duties. Condition of the separation is checked to clarify whether the operations are:

   - mutually excluded
   - disallowed to execute concurrently
   - disallowed to perform both operations on the same set of data

If the user violated the specified condition, the system security officer is alerted. In addition, the misuse rules employed in expert systems within traditional IDSs can also be included. These rules may then be associated with an operation, such as modifying the payroll database to increase one's own wages. In this case, the process is as follows: If modification is performed on the payroll database, check that the employee ID of the user is not the same as that of the record being modified.

Client

This is where the actual data is collected and transferred to the Host for analysis. The *Clients* can be network server systems or end-user workstations. The nature of the data collected may vary depending on the type of the *Client*. For example mail logs can be collected from the mail server, user queries from the database server, and application logs from user workstations. The data to be collected is specified by the system administrator from the *Management Console*. The collected data can then be refined to a standard format by the *Communicator* module before sending the data to the *Host*, so that data from heterogeneous *Client* systems is in a standard format. The *Client* may also have a *Responder* module to respond to detected incidents, and the appropriate response for each incident can be specified from the *Management Console*. For example, when a misuse is detected, the *Responder* may be configured to terminate the user session, revoke privileges, deny further access, alert the security officer, or terminate the anomalous process (Papadaki et al. 2003).

Implementation Issues

In order to be able to implement the system successfully, separation of duties would first need to be defined at the organisation level. Before doing this, the responsibilities of the users need to be defined. Then it needs to be checked that the operations a user is allowed to perform would not lead to a successful misuse. All of these are more of a managerial (rather than technical) issue. However, these are not trivial and could require considerable amount of time and labour. Again, at a technical level, monitoring of user behaviour at application level may require modification of the software package if appropriate APIs are not included.

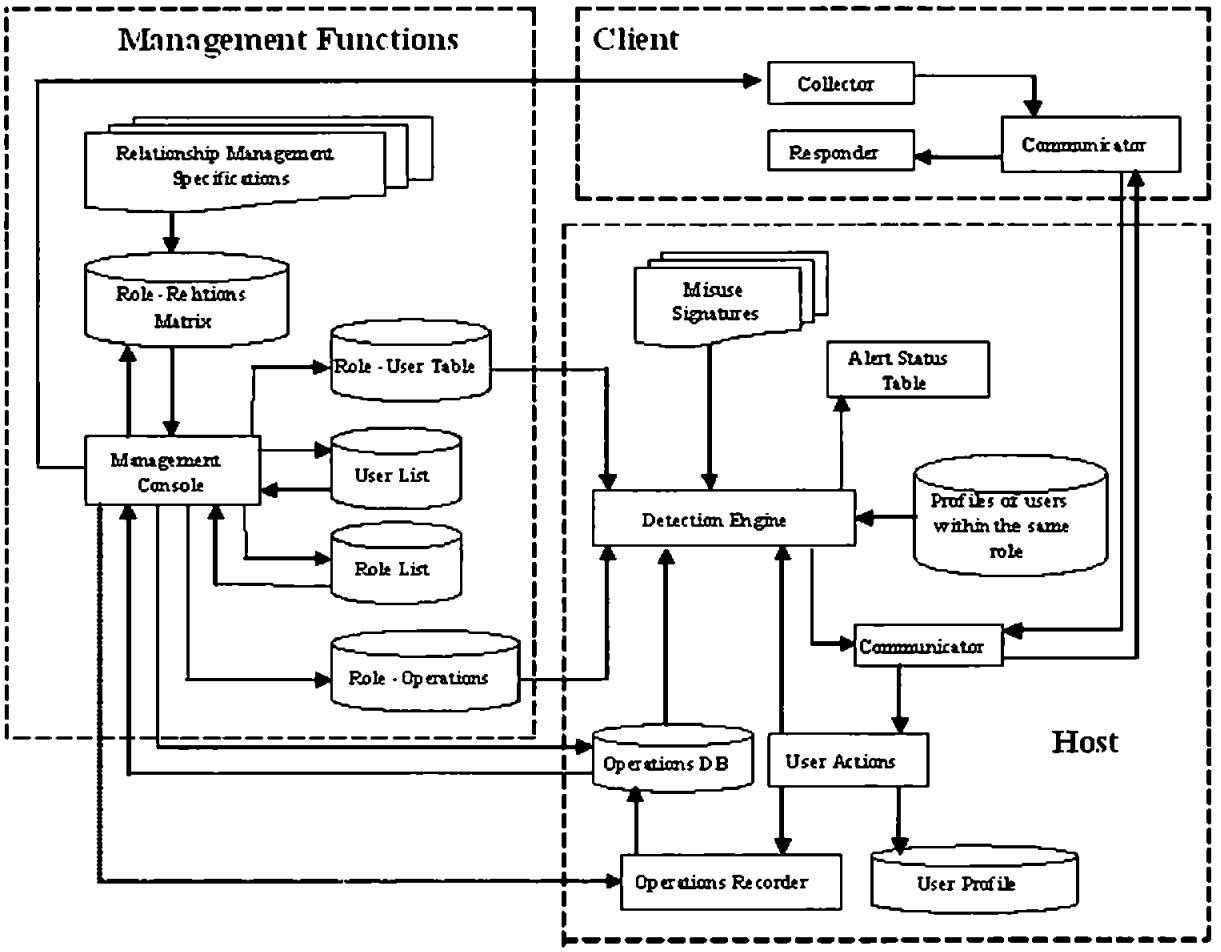


Fig. 1. Conceptual Framework of Misfeasor Monitoring System

CONCLUSIONS

Insiders pose a considerable threat and organisations need to give equal priority in detecting insider abuse as well as outsider attacks. Access controls only allow or deny access; however there is a need to monitor what the user does after gaining access to the system and objects. In order to effectively monitor privilege abuse, IDS require the knowledge of organisation hierarchy, managerial controls, responsibilities and working scopes of each user. The methods employed in RBAC to express knowledge of roles, organisation hierarchy, and separation of duties can be coupled with intrusion detection techniques to detect users who abuse their existing privileges. This paper presented a framework for monitoring users who abuse their existing privileges. The authors' future research will focus on developing the proposed system and testing it against a variety of simulated insider misuses, such as data theft, fraud, net abuse, sabotage, and breach of privacy.

REFERENCES

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

Anderson, J.P. 'Computer Security Threat Monitoring and Surveillance', Technical Report, James P. Anderson Company, Fort Washington, Pennsylvania, April (1980)

Audit Commission, 'Survey of Computer Fraud & Abuse: Supplement.' Audit Commission, (1990)

Chung, C.Y. Gertz, M. Levitt, K. "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. (1999)

Ferraiolo, D. Kuhn, R. 'Role-Based Access Control', In the Proceedings of the 15th National Computer Security Conference, pp. 554-563, Baltimore, MD. October 13-16 (1992)

Gavrila, S.I. Barkley, J.F. 'Formal Specification for Role Based Access Control User/Role and Role/Role Relationship Management', Third ACM workshop on Role Based Access Control, pp. 81-90, Fairfax, Virginia, October 22-23 (1998)

Low, W. L. Lee, J. Teoh, P. (2002) 'DIDAFIT: Detecting Intrusions in Databases Through Fingerprinting Transactions'. In the Proceedings of the 4th Internationsal Conference o Enterprise Information Systems, Ciudal Real, Spain, April 2-6, 2002,

Phyo, A.H., Furnell, S.M. (2004), 'A Detection-Oriented Classification of Insider IT Misuse', to appear in Proceedings of the 3rd Security Conference, Las Vegas, USA,

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

Richardson, R. '2003 CSI/FBI Computer Crime and Security Survey', Computer Security Institute. http://www.gocsi.com, Spring (2003)

Sandhu, R.S. Coyne, E.J Feinstein, H.L and Youman, C.E. 'Role-based access control models'. IEEE Computer, 29(2):38-47, February (1996)

Papadaki, M. Furnell, S.M. Lines, B.M and Reynolds, P.L. 'A Flexible Architecture for Automated Intrusion Response', Proceedings of Seventh IFIP TC-6 TC-11 Conference on Communications and Multimedia Security, pp.65-75, Turin, Italy, October 2-3 (2003)

Prerequisites for Monitoring Insider IT Misuse

Aung Htike Phyo, Steven Furnell, and Andrew Phippen
Network Research Group, University of Plymouth, Plymouth, United Kingdom
E-mail: aung@jack.see.plymouth.ac.uk

Abstract

Although the problem of insider misuse of IT systems is frequently recognised in the results of computer security surveys, it is less widely accounted for in organisational security practices and available countermeasures. The countermeasures available today are oriented towards prevention and detection of outsider attacks on the organisation's IT systems and services. This paper discusses the possibility of applying similar mechanisms and strategies towards monitoring of insider IT misuse. It also discusses the requirements that need to be satisfied before insider misuse monitoring can be put in to practice, and on the basis of the discussion, it is recommended that a misfeasor monitoring system should include features for monitoring file access through arbitrary applications, file replication, partial data replication, file transfer, file deletion, user management, settings/configuration management, database access, and Internet access.

Keywords

Intrusion, Detection, Misuse, Misfeasor

1. Introduction

Frequent headlines reporting hacker break-ins to computer networks and fast spreading computer viruses have steadily increased public awareness of the threats posed to information security. However, external hackers and malicious software are far from being the only threats to the security of an organisation's IT systems and valuable data. CSI/FBI survey results have consistently shown that a significant amount of financial loss can be attributed to insider IT misuse.

Year	System penetration by outsider	Insider abuse of Internet access	Unauthorised insider access
1998	$1,637,000	$3,720,000	$50,565,000
1999	$2,885,000	$7,576,000	$3,567,000
2000	$7,104,000	$27,984,740	$22,554,500
2001	$19,066,600	$35,001,650	$6,064,000
2002	$13,055,000	$50,099,000	$4,503,000
2003	$2,754,400	$11,767,200	$406,300
2004	$901,500	$10,601,055	$4,278,205
2005	$841,400	$6,856,450	$31,322,100
2006	$758,000	$1,849,810	$10,617,000
Total	$49,002,900	$155,455,905	$133,877,105

Table 1: Annual losses for selected incidents from CSI/FBI surveys

The survey results of ICT Fraud and Abuse 2004 (Audit Commission, 2005) has also revealed that the majority of the perpetrators (over 80%) originated from inside the organisation, with operational staff 37%, administrative/clerical staff 31%, and managers 15%.

From the organisation's point of view, insiders can be employees, part-time employees, consultants, contractors and employees of partner firms. From the IT system's perspective, insiders are users with a valid login account and have legitimate rights and privileges to access the resources it manages. Within the scope of this paper, the discussion concerns individuals who have legitimate access to the organisation's IT system and resources, but abuse their access rights. Anderson (Anderson, 1980) termed such users as *misfeasors*. The insider abuse can be more damaging than many outsider attacks, since the perpetrators have a good idea of what is 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 (Einwechter, 2002). A survey commissioned by Microsoft has revealed that amongst the 2,226 UK employees who responded, if there was an opportunity 54% would be willing to gain illegal access to sensitive information stored on their employer's IT systems, while 22% admitted to have already done so (Microsoft, 2006).

This paper evaluates the applicability of existing security mechanisms towards prevention and detection of misfeasor activities. The discussion begins with the motivations involved in misfeasor activities, and associating the motivation with the type and nature of the activities. It then proceeds to analysis of currently available Intrusion Detection Systems, how these tools function and their applicability within the context of misfeasor monitoring. The paper then discusses the requirements that need to be satisfied in order to enable effective monitoring of misfeasor activities in practice.

2. Background

2.1 The definition and the scope of the terms (Insider and Misuse)

It has been stated that within the scope of this paper an insider is an individual with valid login account and have legitimate access to the system and its resources. Then, what is misuse, when the user accesses the system and the resources that he/she has legitimate system level access rights to? Within the scope of this paper, misuse can be defined as any activity that the user has legitimate system level rights to perform, however the activity may not be acceptable within the context of the application, organisation, moral conduct, or ethical conduct. The type of activities may vary, however, motivation behind misfeasor activities can be classified into three distinct categories:

> **Vengeance**: Former/disgruntled employees may be motivated to carry out damaging/disruptive or generally unethical activities upon organisation's IT systems and data. The activities motivated by vengeance may include denial of service attacks on company servers, or sabotage of organisation's IT systems and/or resources, and exposure of confidential information (Gaudin, 2000). For example, deletion of critical business databases, or configuring critical servers in such a way that it becomes vulnerable to attacks, becomes easily accessible to unauthorised users, or becomes inaccessible to authorised users. Another example is intentionally exposing confidential information so that it may damage the reputation of the organisation, or cause embarrassment to an employee/customer. Sometimes the activity may not be directed towards the organisation, but rather a colleague, or an acquaintance that happens to be one of the organisation's customers. However, the organisation may still be held liable for failing to protect the data.

Financial gain: Activities motivated by financial gain may include providing proprietary or confidential information to unauthorised parties, and/or configuring the systems in such a way that unauthorised parties may gain access to proprietary and confidential information, in return for financial benefits. In addition, the misfeasors may also defraud the organisation and/or its customers for financial gain (Dhillon and Moores, 2001).

Recreation & Curiosity: Activities include recreational web-surfing, downloading illegal software, perusing and writing personal emails, chatting through instant-messengers. While performing these activities, users may be unable to carryout productive work. In addition, media downloaded from the Internet may be copyright protected, or contain inappropriate content such as pornography, which may damage the organisation's reputation and the organisation may also be held liable. Misfeasors may also access organisation's business databases for personal reasons, which may result in breach of privacy to an employee or a customer.

In addition, accidental misuse may also occur as a result of negligence or users' lack of IT security awareness (Furnell, 2006).

Deriving from the analysis made previously, the activities that are legitimate in the system and network context, yet may be deemed unacceptable/inappropriate in the organisation/business and application context include:

1. Internet access
2. File access through arbitrary applications
3. File replication (copy, paste, save as)
4. Partial data replication (print screen, copy, paste)
5. File/data transfer through communication applications
6. Settings/configuration changes
7. User management
8. Database access

Now, an evaluation needs to be made in order to determine whether current Intrusion Detection Systems (IDS) can be employed to detect misfeasor activities.

2.2 Intrusion Detection Systems (IDS)

Intrusion detection systems are generally categorised based upon the data analysed in order to recognise an attack.

Network IDS: analyse network packets, network protocols and network statistics in order to detect attempts to exploit network protocols and network applications. A successful attack may result in legitimate users being unable to access organisation's network services, or the attacker may gain access to the machine on which the server application is run.

Host IDS: analyse resource utilisation (CPU/memory/disk usage, number of files opened, number of system calls made), and behaviour of applications (system calls, file access) to detect attempts to exploit system/application vulnerabilities. A

successful attack may result in the attacker gaining access to the machine, or the attacker gaining higher privileges.

Today, hybrid systems that analyse both network, and host data for detecting attacks are available.

IDS can also be categorised based upon the detection strategy employed (Amoroso 1999).

Misuse 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. With this approach the detection system is only as good as the database of attack signatures, and may not be able to detect variations of an attack. The problem is that, misfeasor activities do not demonstrate the same characteristics as external penetration attacks (Schultz 2002).

Anomaly Detection: This approach relies upon watching out for things that do not look normal when compared to typical user activity within the system. 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. With this approach, variations of an attack or novel attacks may be detected. However, characterising normal behaviour is difficult and, deciding the variables to be involved for characterisation still requires insight knowledge of the system and the application environment.

IDS may employ a variety of techniques, including expert systems, neural networks, and statistical analysis for detecting attacks. It is conjectured that existing techniques, and monitoring strategies may also be applied to detecting misfeasor activity. However, majority of currently available IDS are designed to detect network penetrations, and privilege escalation attacks. Misfeasors do not need to perform network penetration attacks, since misfeasors already have legitimate access to the network and systems. By definition, misfeasors do not perform privilege escalation attacks, and do not violate system level controls. However, misfeasor activities may be deemed unacceptable within the application, business, or organisation context. Therefore, any inherent ability to detect misfeasor activity by current IDS would be a coincidence rather than by design. The fact, that misfeasors do not violate system level and application level controls, makes it extremely difficult to identify misfeasor activity due to lack of reference data/information in order to conclude whether violation of (security or acceptable usage) policy has occurred. In addition some misuses may not be evident at network or host level alone, and misuse may only be recognised when analysed in the context of the application, business rules surrounding the operation, and within the context of the organisation. The correlation of network, host, application, contextual information and rules is needed for analysing the possible occurrence of misuse. Therefore, the data required for successful detection of misfeasor activities need to be identified first.

3. Relevant Data for Misfeasor Analysis

Within the IT environment, users access and manipulate the data stored and managed by the computer system through the use of application programs. The entities involved in the data access are, the machines involved (server-client, peer-peer), the data, the users, and the

application utilised. Therefore, information regarding these entities will certainly be relevant for misfeasor analysis.

Machine Details: Files and databases are stored, processed, manipulated, managed, and transferred to and from computer systems. Although a user has legitimate access rights to the data, the machine utilised to access the data may not satisfy security requirements of the data. For example, a user who has access to the data transfers the file to an external machine. Although, the user at the receiving end might be also authorised to access the data, the machine utilised at the receiving end may not be regulated by organisation's security mechanisms. Therefore, security requirements of the system such as which users have access to the machine, which computers can access the file/database server, and other details such as location and physical security of each machine will be relevant for detecting misfeasor activities.

File and Database Security Requirements: Data is stored within files and databases on computer systems. Since the aim is to ensure the security of the data, it is essential that the security requirements of the file/databases are provided to the monitoring system for reference. Access control mechanism determines only whether the user can read the data, and edit or delete the data (Escamilla 1998). In order to detect data theft/leakage, information regarding whether partial/whole replication of data is acceptable, and whether the data can be saved to a removable media need to be defined. It is more difficult to manage the security of multiple copies of confidential data on various machines. Therefore, it is also important to keep track of how many of copies of a critical file exist, and where they are located. Keeping track of critical files will also become useful when recovering data if it is deleted on certain machines, or verifying whether it is the only copy prior to deletion. Since the business managers have better knowledge of the sensitivity, and the users who needs access to the data and the validity of access, the business managers should be given the responsibility of defining the security requirements of the data, instead of the system administrator who may not have equal knowledge of the contents and security requirements of the file/data in the business context. In addition, if an event does not satisfy the security requirements of the data, the business manager should be alerted. Therefore, the information regarding who the file custodian is will also be useful for alerting the right personnel in the event of suspicious activity. It may not be practical to monitor all data files as a computer system may also contain system files and user's personal files, and thus files regarded as intellectual property of the organisation and files that require misfeasor monitoring should be listed and tagged with security policy.

User Details: A misfeasor that has access to the file may transfer the file to someone who is not authorised access. The file may be transferred through email, messengers, or some other programs with communication capability. Therefore, contact addresses of organisation's employees, customers, or contractors should be provided to the monitoring system to determine misfeasor activity. Information regarding, the user's responsibilities and role within the organisation will also be useful when alerting the system administrator or file custodian, so that the file custodian will be able to make better decision regarding the validity of the activity within the business context.

Application function and capabilities: The application utilised determine what the user can do with the system or data accessed. Therefore, data regarding user activity within the application environment will be relevant for detecting misuse. However, it

may not be practical to monitor all applications and application functions. Applications that require monitoring can be divided into two categories based on the data access capabilities.

Applications with access to file or databases: Applications with direct access to file and databases include file managers, word processors, document readers, image editors, media players and database programs. File managers do not have direct access to the contents of the file, but provide capability to replicate, move, and delete the file. User activities regarding file replication, relocation, and deletion needs to be monitored to detect misuse. Document readers and processors have direct access to the entire contents of the file, and also provide capability to edit, and replicate partial or entire contents of the file. It may not be possible to automate the integrity checking of the contents of documents, if various users may be allowed to update the document, because the structure of the data within the documents may vary with each update. Database programs access small part of the file; however a single record may contain critical information regarding the organisation, a business transaction, an employee, or a customer. User access to each record, for both viewing and updating needs to be verified. If possible, access to each record should be validated, and integrity of the record should be verified after each update. To be able to automate this validation and verification process, information for reference needs to be provided to the monitoring system.

Application with no access to file or databases: The applications that do not have direct access to the contents of the file yet may affect the security of the system and data include security applications, configuration managers, user management applications, and applications with communication capability. Security applications can be used to harden or weaken the security of a system or an application, which may result in unauthorised users gaining access or authorised users being unable to access. Therefore, changes to security settings need to be verified against security requirements of the system or application as defined in the policy. Proper functionality of a system or an application depends on the correct configuration, and thus changes to configuration need to be verified against an appropriate reference. Adding users to a system or a role, in effect allows the user to gain access to the system or the files accessible for the assigned role, therefore system administrators and role managers should be asked to authorise the addition of a user. Applications with communication capabilities, such as email and messenger may be used to transfer files, or partial data. In order to detect misuse, the monitoring system needs to determine whether the server mediating the communication is managed by the organisation, whether the recipient is authorised to access the file, and whether the machine utilised by the recipient satisfy security requirements for accessing the file transferred. Therefore, the details of the file, the sender, the server, the recipient, and the machines utilised for communication is required for analysis of possible misfeasor activity.

Before misfeasor monitoring can be put in to practice, the applications need to provide the monitoring system with the information described previously in order to enable misfeasor activity detection.

Contextual rules related to operations: Sometimes, certain conditions may need to be satisfied for an operation to be legitimate within the application environment and business context. Required conditions may vary from one business to another, and one operation to the next. When an operation does not conform to the required conditions, the activity may result in fraud/misuse. There may be pre-requisite conditions to be satisfied. For example, when a user account is created, the contextual may require that the user of the account exists in the human resource database as an employee of the organisation. There may be post-requisite conditions to be satisfied. For example, when a user is added to a role, the policy may state that the role manager must verify the addition of the user to the role, and the time period for verification to be made may also be defined. Within certain applications, other contextual rules may exist. For example, in some businesses if the payment is made within fifteen days of a purchase, then the customer is entitled to a prompt payment discount. Depending upon whether the organisation is the customer or the supplier, there may be opportunities for employees to commit fraud in such cases, and the organisation and the supplier/customer may be defrauded. For the monitoring system to be able to detect, misfeasor activity, the system needs to be provided with the knowledge of contextual rules relating to the operation. For certain operations, the value entered by the user may determine whether/when the verification of the operation takes place. For example, the loss/profit calculation date may determine when the loss/profit calculation for a business takes place and phoney profits may be generated or verification of losses may be delayed.

Questions have been raised as to why the aforementioned contextual rules are not used as access control for operations, rather than monitoring. The reasons for this is that in some cases the application developers could not have foreseen the contextual requirements, and it is not practical to hard-code contextual rules within the application because the rules may not apply to all business transactions, and the rules may change within a short period as the business practice evolves in order to be competitive.

4. A Generic Misfeasor Monitoring Tool

Based upon the requirements noted previously, a generic misfeasor-monitoring tool may be designed. Deriving from the analysis made previously, the user activities that should be monitored are database access, data replication, data transfer through communication programs, user management, and settings/configuration management of system and applications. The information required to determine possible misuse, concerning the described activities will be discussed in detail.

File Access: The application utilised by the user to access the file determine what the user can do with the file accessed. In addition, if an arbitrary application is utilised, the user may by pass application level controls embedded within the normal application. Therefore, the monitoring system should be able to determine whether the application utilised is the normal application for accessing the file concerned. For the monitoring system to be able to determine the access of file through arbitrary application, the system needs to be provided with the information regarding the application normally used for accessing the file, and the application utilised by each user for accessing the file. Thus each file listed for misfeasor monitoring needs to be tagged with the identifier of the application normally used for access, so that the monitoring system

can compare it against the application utilised by the user for accessing the file, in order to determine possible occurrence of misfeasor activity.

File replication: When a user copies and pastes a file, the monitoring system needs to determine whether the source file is listed for misfeasor monitoring. If the source file is listed then, the system needs to determine whether replicating the entire file is acceptable, or saving the file to a removable disk is acceptable. If replicating the entire file, and/or saving the file to a removable disk are acceptable then no further analysis needs to be made and no one needs to be alerted of the activity. However, if replicating the entire file is not acceptable, then the monitoring system needs to alert file custodian of the activity with the details. The details of the event that should be provided are the source file ID, the machine on which the copy is saved, the exact file path of the copy, and the user who performed the activity. Thus each file listed for misfeasor monitoring needs to be tagged with the policy regarding whether replicating to removable disk is acceptable, whether replicating the file is acceptable, and who should be alerted in the event of policy violation.

Partial data replication: When a user performs Print Screen, Cut, or Copy activity when a file is accessed, the monitoring system needs to determine whether the source file from which the data is copied has been listed for misfeasor monitoring. If the source file is listed, then the clipboard data needs to be associated with the source file ID. When the user Paste/Inserts the clipboard data, then the file custodian should be alerted the details of the event. The details of the event include source file ID, the user responsible for the activity, file path of the document into which the copied data is pasted, the machine on which the file is saved. The files listed for misfeasor monitoring needs to be tagged with the policy whether partial replication of the contents is acceptable.

File transfer: When a user transfers a file, the monitoring system first needs to determine whether the file is listed for monitoring, and whether saving the file to a removable disk is acceptable. If the file is listed and saving the file to a removable disk is not acceptable then the monitoring system needs to determine whether the server mediating the transfer is managed by the organisation, i.e. if it is an internal server. If it is not an internal server then, the file custodian and the file server administrator should be alerted of the activity. If the server is internal then, the monitoring system needs to determine whether the recipient is also an insider. If the file is transferred through the email application, the recipient's email should be checked against employee email address list to determine whether the recipient is an insider. If the recipient is not an insider then the file custodian should be alerted. If the recipient is an insider then, the monitoring system needs to determine whether the recipient is authorised to access the file. If the recipient is not authorised to access the source file then, the file custodian should be alerted with the details. If the recipient is authorised to access the source file then, the monitoring system needs to determine whether the machine utilised by the recipient to retrieve the file satisfy security requirements, i.e. whether the machine is authorised to access the File server where the source file is located on. If the machine utilised by the recipient to retrieve the file is not authorised to access the file server of the source file then the system administrator of the file server and file custodian should be alerted of the activity along with the details.
The information needed for monitoring this activity is, whether replication of the file and saving the file to removable disk is acceptable. The monitoring system also needs

to be provided with the list of internal machines for it to determine whether the communication server involved is managed by the organisation. The monitoring system then needs the username/addresses of insiders, so that it can determine whether the recipient's username/address is that of an insider's. The monitoring system then needs to be provided with the role(s) and users allowed to access the file, so that it can determine whether the recipient is authorised to access the file transferred. The monitoring system then needs to determine whether the machine utilised by the recipient for retrieving the file is an internal or external machine. The monitoring system also needs the knowledge of which machines are allowed to access the file server the file originated from, so that it can determine whether the machine utilised by the recipient is authorised to access the file server i.e. acceptable to access the contents of the file server.

File Deletion: When a file is to be deleted, the monitoring system should be able to determine whether it is the only copy that exists within the organisation's IT systems. The list of files that need to be monitored is required, and information regarding how many copies of each file exist and where each file is stored, and who is responsible for the security and availability of the file is needed in order to determine possible sabotage, and to inform the right personnel.

User management: When an account is created or a user is added, the added user will gain access to the system, application, file, or records depending upon the list the user has bee added to. If the user is added to the user-list of a server then the user will gain access to the server, if the user is added to a role then the user will gain access to the resources given access to the role members. Therefore, the List Custodian should be informed of the addition of users to the list. Thus, the monitoring system first needs to identify to which list the user have been added. Once it has been identified then the custodian of the list should be alerted for verification.

Settings/configuration management: Changing the settings of a system/application may also be a stepping-stone towards a misfeasor activity. When a system is first set up, the required settings for both security and functionality should be defined. When a user activity affects settings/configurations, then the monitoring system needs to determine whether current/attempted settings of the system/application satisfy the required settings defined when it was first set up. If the current/attempted settings vary from required settings defined by the policy, then the administrator of the system/application should be alerted with the details. The details should include, the affected machine, the affected application, the user responsible, current settings, and required settings stated by the policy. The monitoring system needs to be provided with the required settings for each application installed on each machine, so that analysis can be made to determine whether the changes made by the user conforms to requirements.

Database access: Each user's database access statistics can be monitored on the basis of the number of records accessed per defined period, the number of records accessed per related event/quantity, and comparing the number of records accessed by each user to that of the average accessed by other users belonging to the same role within the organisation. However, the validity of each record accessed by each user should also be verified. When a record is viewed, the monitoring system should be able to determine whether the user had a valid reason to access the record. If a record is added

or updated, the monitoring system should be able to determine the integrity of the data within the record. The monitoring system needs to be provided with the list of data tables that require monitoring, and the corresponding data table where the data for reference may be found. The monitoring system also needs to know the attributes that share a common value in both data tables, so that the corresponding reference record may be identified. The monitoring system also needs the information regarding the attributes that need to be verified from the two tables, and the condition of the verification, i.e. check for existence, both values must equal, or a value must be True.

Internet access: Employees may abuse the ability to access the Internet through organisation's IT systems by downloading illegal software, online shopping, and accessing inappropriate content. In order to be able to detect abuse, the monitoring system must be provided with acceptable usage policy. The acceptable usage policy may indicate the acceptable number of bytes downloaded per defined period or per user, the URLs deemed acceptable for access, the acceptable amount of time spent utilising the web browser, and the types of media acceptable for download.

5. Conclusions

Without having relevant data for analysis, the monitoring system will not be able to carryout accurate detection of possible misfeasor activity. The data analysed by current IDS related to network and system level events, and these data may be analysed for detecting network penetrations and privilege escalation attacks. However, the misfeasors do not need to perform network penetrations or privilege escalation attacks in order to gain access to the network and systems. Misfeasors already have legitimate access to network and systems in order to carry out their day-to-day tasks. However, while some of the activities may be perfectly acceptable at network and system level, the activity may be unacceptable within the context of the application and acceptable usage policy defined by the organisation. Therefore, in order to be able to detect violation of contextual rules regarding the application, organisation, or a business process, the monitoring system needs to be provided with the contextual information related to application, organisation, business operations, and acceptable usage policy. Currently, a demo misfeasor monitoring tool is being designed based on the specifications derived from the discussion made in this paper, and developed in order to test the relevance of log data mentioned for the analysis of misfeasor scenarios.

6. References

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", Technical Report, James P Anderson Co., Fort Washington, April 1980.

Audit Commission. (2005), "ICT Fraud and Abuse 2004 - An Update to yourbusiness@risk", Audit Commission Publications, UK. June 2005

Dhillon, G., Moores, S. (2001), "Computer Crimes: Theorising about the enemy within", Computers & Security, Vol.20, No.8, pp715-723.

Einwechter, N (2002), "Preventing and Detecting Insider Attacks Using IDS", http://www.securityfocus.com/infocus/1558

Escamilla, T. (1998), "Intrusion Detection: Network Security Beyond the Firewall", John Wiley & Sons, Inc. ISBN 0-471-29000-9, 1998

Furnell, S. (2006), "Malicious or misinformed? Exploring a contributor to the insider threat", Computer Fraud & Security, September 2006

Gaudin, S. (2000), "Case Study of Insider Sabotage: The Tim Lloyd/Omega Case", Computer Security Journal, Volume XVI, No.3

Gordon, L.A. Loeb, M.P. Lucyshyn, W. Richardson, R. (2004), "2004 CSI/FBI Computer Crime and Security Survey", Computer Security Institute, 2004.

Gordon, L.A. Loeb, M.P. Lucyshyn, W. Richardson, R. (2005), "2005 CSI/FBI Computer Crime and Security Survey", Computer Security Institute, 2005.

Gordon, L.A. Loeb, M.P. Lucyshyn, W. Richardson, R. (2006), "2006 CSI/FBI Computer Crime and Security Survey", Computer Security Institute, 2006.

Microsoft. (2006), "Survey Finds: Employer may be leaving the door open to internal espionage", Press Release, Microsoft UK, 30 May 2006. http://www.microsoft.com/uk/press/content/presscentre/releases/2006/06/PR03635.mspx

Power, R. (2001), "2001 CSI/FBI Computer Crime and Security Survey", Computer Security Issues & Trends, vol. VII, No.1. Computer Security Institute. Spring 2001.

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

Richardson, R. (2003), "2003 CSI/FBI Computer Crime and Security Survey", Computer Security Institute, Spring 2003.

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