2003

A Correlation Framework for Continuous User Authentication Using Data Mining

Singh, Harjit

http://hdl.handle.net/10026.1/1644

http://dx.doi.org/10.24382/3876

University of Plymouth

A Correlation Framework for Continuous User Authentication Using Data Mining

by

HARJIT SINGH

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

DOCTOR OF PHILOSOPHY

School of Computing, Communications and Electronics

Faculty of Technology

December 2003

AN EPITAPH TO HARJIT SINGH

As his Director of Studies it was a privilege to have supervised Harjit Singh.

He was one of the most intelligent and diligent students I have supervised in my academic career.

Harjit excelled both as an undergraduate, obtaining a 1st Class Honours degree from University of Plymouth, and as a member of the Network Research Group, while researching for his PhD.

With his excellent grasp of the fundamental issues of computing and communications, allied to his highly enquiring mind, Harjit was an exceptionally gifted researcher, and his untimely death is a great loss to all those who have had the pleasure of knowing and working with him.

Dr B.M.Lines
27/4/04

Table of Contents

List of Figures

Acknowledgements

I would like to thank the following people for their invaluable assistance during this research project:

Dr. S.M. Furnell	- Network Research Group, University of Plymouth
Dr. B.M.Lines	- Network Research Group, University of Plymouth

Above all this thesis is dedicated to my family and friends who have helped and supported me throughout the project.

Chapter 1

Introduction and Overview

1. Introduction and Overview

1.1 Introduction

The increasing security breaches revealed in recent surveys and security threats reported in the media reaffirms the lack of current security measures in IT systems. While most reported work in this area has focussed on enhancing the initial login stage in order to counteract against unauthorised access, there is still a problem detecting when an intruder has compromised the front line controls. This could pose a serious threat since any subsequent indicator of an intrusion in progress could be quite subtle and may remain hidden to the casual observer. Having passed the frontline controls and having the appropriate access privileges, the intruder may be in the position to do virtually anything without further challenge. This has caused interest in the concept of continuous authentication, which inevitably involves the analysis of vast amounts of data.

The primary objective of the research is to develop and evaluate a suitable correlation engine in order to automate the processes involved in authenticating and monitoring users in a networked system environment. The aim is to further develop the *Anomaly Detection* module previously illustrated in a PhD thesis [1] as part of the conceptual architecture of an Intrusion Monitoring System (IMS) framework.

1.2 Thesis structure

This thesis begins with Chapter 2, which provides an introduction to the issues surrounding information systems security. It includes an overview of current security measures adopted in dealing with security breaches in a networked system environment, the concepts of computer system security and the generic classes of abuses. It then leads on to the discussion of the correlation strategies and detection algorithms adopted in analysing the audit trails collected in the audit analysis stages, which is the area of interest considered for the research work.

Chapter 3 provides an overview of Data Mining (DM), the Intelligent Data Analysis technique considered for automating the audit analysis stages. It introduces some concepts behind the terminology used for the knowledge discovery phases with reference to the DM approach. It then proceeds to provide a detailed account of the steps comprising the DM process and introduction to some of the techniques used in DM. It also presents some of the algorithms commonly used for the DM process.

Chapter 4 presents the comparative studies carried out on the various learning algorithms used in DM. It includes some initial results of the evaluation work and discussion of these results.

Chapter 5 then proceeds to describe a conceptual intrusion correlation framework for user authentication and monitoring scheme. This forms a major part of the research work carried out, and shows the context in which the underlying behavioural profiling techniques could be applied.

Finally, Chapter 6 concludes the thesis, highlighting the achievements of the research programme and identifying avenues of future research opportunity.

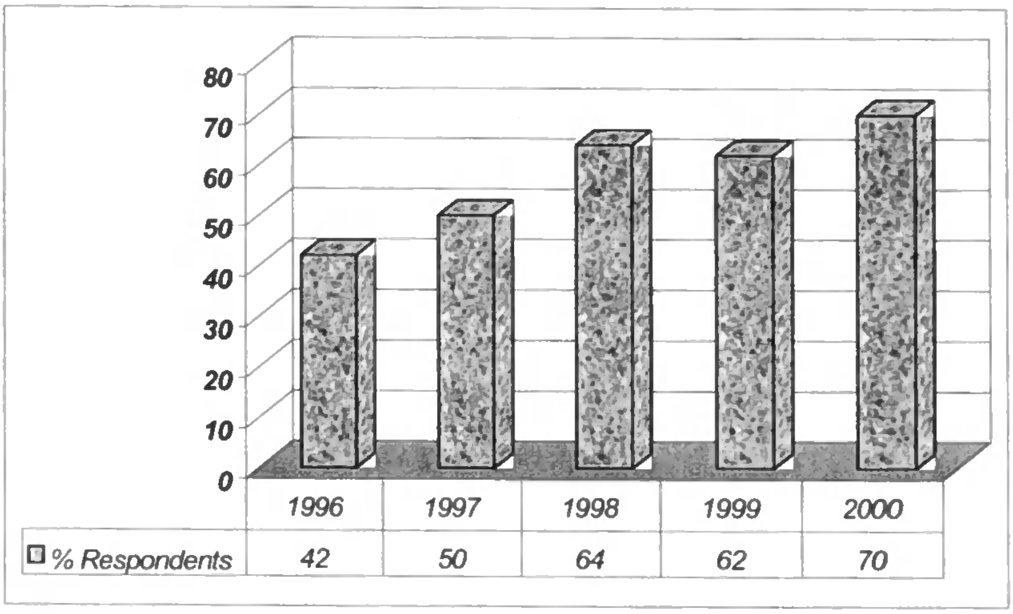
2. User Authentication and Intrusion Detection

2.1 Introduction

In the last few decades, there has been a global expansion of telecommunications networks and the emergence of new transport technologies that have enabled the capabilities for high-speed data transmissions and faster processing. This in turn, coupled with deregulation and privatisation, has contributed to the growth and advancements of computer networks. These developments have inevitably contributed towards the impact of Information Technology (IT) upon our modern society. The phenomenon often referred to as "the information revolution" can vividly be seen from the proliferation of local area networks (LANs) in organisations, academic, business, and research institutions, which in turn are connected to the outside world via the Internet. This ever evolving technology has introduced a new dimension with which many new services in various forms (e.g. email, WWW, FTP, etc.) have been introduced to connect people on a global scale. Gone are the days when the Internet was used primarily for promoting one's product or services, the available infrastructure and services are becoming an integral part of an enterprise and small businesses alike in today's global business environment. The new business opportunities offered for instance, through the use of electronic commerce (e-commerce), is further evidence of the impact of IT upon our everyday lives.

Concurrently, it has introduced the increasing dependency on rapid access and processing of sometime sensitive information whereby computers are an integral part of information management in modern organisations. More importantly the dire consequences of the increasing reliance upon IT and networked systems is the serious impact if someone deliberately sets out to misuse or abuse the system, which inevitably could lead to a range of undesirable repercussions for the affected organisations (e.g. disruption to activities, financial loss, legal liability and loss of business goodwill). A recent study conducted by the US Computer Security Institute (CSI), in collaboration with the FBI, reported that 70% of respondent organisations had detected unauthorised use of their computer systems in the previous 12 months [2] – which represented an 8% increase on previous findings from 1999. Analysing the survey reports over a five year period, as depicted in *Figure 1*, reveals that the percentage of respondent organisations reported unauthorised use of computer systems is increasing.

The increasing security breaches reported in the media reaffirmed these threats. For instance, the attacks as a direct result of cyber crime activities on dot com sites, such as distributed denial-of-service (DDOS) attacks on Yahoo, eBay, Amazon.com and other popular web sites, illustrates how delicate and vulnerable the Internet can be [3]. Even the net's warning centre, the Computer Emergency Response Team (CERT) that alerts people to the activities of malicious hackers, has been attacked [4]. More recently have been the high profile attacks on Microsoft sites, where blueprints of Microsoft's software programs were allegedly accessed by hackers via a Trojan called Qaz – a "worm", which makes copies of itself to spread throughout a network. Once installed, the Qaz program allows hackers unauthorised access to the network by, for example, relaying back to them passwords and other secret information, which is further evidence of the threat [5]. Even more revealing is the staggering fact of the paucity of current security measures.



	1996	1997	1998	1999	2000
▨ % Respondents	42	50	64	62	70

Figure 1: Percentage of organisations reporting abuse over a period of 5 years

Although security breaches have often been perceived to be carried out by external intruders bypassing security policies or exploiting inherent flaws in systems through the Internet, a potentially increasing threat is unauthorised access by insiders [2]. These are abuses carried out by users within an organisation or originating from a legitimate user, having access to the networked computers inside the perimeter of the organisation or via a remote connection to the networked computers. These abuses, if not detected, could result in unauthorised disclosure of information, and the modification or destruction of sensitive data. Typically these are users who misuse their privileges. For instance a Financial Institution employee can change the account details of someone because they did not like that person. Security breaches could also arise from unauthorised users masquerading using the identity of legitimate user's or from *Clandestine* users who can evade access control and auditing. The various classes of abuses as result of both, mischief and malice have been comprehensively categorised by [6] and are as described in *Table 1*. Although these encompass most of the human abuse that could occur, it should be noted that the categorisation does not take into account of any abuses that might be related to software activities (e.g. viruses, worms, etc.). This is understandable when considering the first computer virus reported was allegedly written in 1986 [7], whereas the analysis made was before such incidents had become a commonplace.

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 1: Categories of system abuser

A significant challenge arises in dealing with such internal abuses while most approaches are attained towards tackling issues surrounding the external breaches of security. Therefore this research work is focused towards addressing these issues of internal security breaches. This chapter proceeds first to describe current approaches adopted for the security mechanisms in place with reference to the user authentication schemes employed to protect the IT systems. This forms the basis of some of the discussion in relation to the shortcomings of the current methods adopted. The latter part of the thesis brings into focus some of the current approaches used to combine data and information from numerous sources to enable the correlation stages in detecting security breaches or threats occurring on the system.

2.2 Security measures

It is clear from the reported breaches that the subject of security in networked systems is a topic of much debate. Although most systems offer some security in the form of user authentication, this thesis is aimed at addressing the issues surrounding the approach adopted and the methodology involved in integration of these approaches within a networked environment. The various concepts adopted in current approaches are firstly introduced, which will enable the discussion on a proposed advanced intrusion correlation framework for user authentication and monitoring in the later chapters to be appreciated. While there are numerous hardware and software tools available on the market, which offer some protection against the various abuses, many of the sophisticated security measures are still being developed. The existing approaches adopted in personal computers (PCs) or as network-wide security policy are therefore aimed at maintaining an appropriate level of security in a networked systems environment. The approaches adopted to protect these IT systems are generally derived from the principles behind user authentication methods, which are primarily based upon three main principles; specifically something you know (e.g. password or PIN), something you have (e.g. key, smart card or other tokens) and something you are (e.g. biometrics) [8].

2.2.1 Authentication methods

The most common security mechanisms are point-of-entry identification and authentication schemes. These methods are typically based upon simple password schemes and the use of file permissions and system privileges in order to restrict access to certain resources. These are common features found in most systems and have been around for some time. As such they are inevitably susceptible to various forms of attack, from "socially engineered" attacks to automated password cracking tools (e.g. LophtCrack [9]) available on the net. For instance, the password cracking tools can obtain encrypted passwords from stand-alone workstations, networked servers to primary domain controllers. They can even capture and crack encrypted password files from the challenge / response exchanged when one machine authenticates to another over the network. Conversely, human factor contribution to the problems (i.e. not using the schemes correctly) can compromise the protection offered by these schemes. Typically, these are problems related to users using easily guessed passwords, sharing the passwords with others, forgetting their passwords, not changing their passwords and so forth. Studies carried out, as reported in [10, 11], over the last 30 years, have revealed the shortcomings associated with using password schemes for authenticating users on systems.

The access control mechanism, which is an extension of the user authentication schemes can be seen adopted as blocking perimeter entry and exit via firewalls, and also through implementation of authentication devices such as tokens. These are physical devices that users carry with them. The principle behind this approach is analogous to an ATM machine. Since it combines two things to prove the user's identity - something the user has (an authenticator) and something the user knows (a password or PIN) - it is also called a two-factor authentication method. The problems associated with access control mechanism are inherently similar to that of the authentication mechanism on which they are based.

The known vulnerability of the traditional user authentication mechanisms employed has generated interest in enhancing these schemes through the use of biometrics. These techniques are based upon the principle of using a measurable physical characteristic or personal trait to recognise the identity, or verify the claimed identity, of a person through automated means [12]. These schemes are finding a common place as stand alone devices or integrated into a system-wide policy control. The biometrics-based user authentication techniques can be separated into two distinct categories, physiological and behavioural characteristics [13], as described in the section that follow.

2.2.1.1 Physiological characteristics

The biometrics-based authentication schemes derived from physiological characteristic uses biological traits that do not change or that cannot be altered over an individuals lifetime without significant duress. It therefore uses physical characteristics such as fingerprint, facial features, retinal or iris patterns. While they provide a relative stable characteristic for user authentication, an additional device or equipment connected externally to the computer is generally required. This is to enable the system to capture the samples of the biometric, which are stored in a biometric template for future comparison during authentication. Furthermore each scheme needs to be evaluated according to the application intended for, since some biometric are better discriminators that others. They are also relatively expensive compared with traditional authentication methods and often involves a trade-off of user friendliness, speed, accuracy, and so forth.

2.2.1.2 Behavioural characteristics

The behavioural biometrics-based scheme depends on characteristics that are more a reflection of an individual's personal trait. In contrast to physiological characteristics, the behavioural characteristic is learned or acquired over a period of time. Examples of such characteristics used are voice pattern, signature recognition and typing style. While in many cases they do not require any additional hardware, some approaches such as signature recognition still require specialised hardware to capture the samples of the biometric.

The accuracy of biometric authentication schemes, which is critical to successful implementation, is dependent on the schemes used, with each having its advantages and disadvantages. The most commonly discussed performance measures used to attest the accuracy of these devices are the False Acceptance Rate (FAR) and the False Rejection Rate (FRR) [13]. The former is the probability that a biometric will incorrectly identify a user or will fail to reject an impostor while the latter is the probability that a legitimate user will be rejected. Both of these measures require large statistical samples. In addition, FRR and FAR are inversely proportional measurements. For example a higher tolerance setting to make it harder for impostors to gain access could result in legitimate users having difficulties to gain access (i.e. as FAR goes down, FRR rises). Conversely, lowering the tolerance could result in an impostor gaining access to the systems (i.e. as FRR goes down, FAR rises). Most biometric systems have a variable threshold and the FAR and FRR are optimised to have a balanced tolerance. Therefore the setting of any such threshold parameter will always represent something of a compromise and is dependent on the application of the biometric-based authentication scheme selected [12].

2.3 Limitations of current user authentication mechanisms

The use of the various schemes described previously (e.g. passwords, PINs, and tokens) carries the security problem of verifying that the presenter is the authorised user, and not an unauthorised holder. Typically the measures employed are focused on preventative strategies and adopt passive mechanisms to provide the systems with some form of protection against the common threats.

Therefore the security measures employed in systems can be seen to have commonality to provide the system with a defensive mechanism. This could pose a serious security threat in the event where an impostor has successfully breached a user account or misuse is occurring from a legitimate user, since any subsequent indicators of an intrusion in progress could be quite subtle and may remain hidden to the casual observer. Having passed the frontline controls and having the appropriate access privileges, the user may be in the position to do virtually anything without being further challenged. No doubt the defensive mechanisms schemes employed such as traditional user authentication and access controls (e.g. passwords and user / group-based file permissions) schemes act as deterrent to most security threats but it is unable to protect the system from "socially engineered" penetrations and the threat of successfully breached account(s) or misuse occurring from legitimate users.

Access control schemes employed in protecting IT systems act as a gate, and are therefore also susceptible to these threats. For instance, the access control policy implemented in firewalls has largely been misunderstood as a solution to recognise unauthorised access and block it, as well as to provide an airtight perimeter defence against this security breach when considering it is exposed to similar vulnerabilities encountered by traditional user authentication and access control schemes. Despite the perimeter and access controls offered both internally and remotely, once inside a system, the users are free to engage in malicious activities without being further challenged. These threats are not limited to internal infringements of policy but are exacerbated by external security threats. Although most access control schemes, such as employed in firewalls, generate activity logs, enterprises have limited resource to carefully analyse the myriad of data generated.

The above highlights the reality that the measures in place currently do not scale up to the problems related to the after-effect of someone bypassing the security policy or misuse occurring from authorised users. This is due the inherent characteristic of defensive mechanisms, which provide one-off rather then continuous authentication. A survey carried out to find out what would Information System managers change in order to make their security infrastructure adequate as reported by [14], reveals a high proportion of respondents – close to 80% point towards the need to change authentication of users schemes and more accurate monitoring. It is considered that the problems above could be addressed through appropriate monitoring and analysis of user activity within an active session, which may potentially reveal patterns that appear abnormal in relation to their typical behaviour, or which are compatible with the sign of recognised intrusion scenarios. This scheme can then be employed to provide continuous authentication of users as part of a security framework. It is from this perspective that many Intrusion Detection Systems (IDSs) have been conceived and leads on the next subject of discussion prior to advocating an advanced intrusion correlation framework in the later chapters.

2.4 Intrusion Detection Systems

The requirement for IDSs can be attributed to the need to complement existing security measures in order to protect the system against intrusions. Hence it is important to define what constitutes an intrusion in the context of network and computer security, before examining the architectural issues of IDSs. An intrusion can be defined as any set of actions that attempt to intrude or otherwise compromise the integrity, confidentiality, or availability of system or network resources [15]. Intrusion detection could then be defined as the process of identifying and responding to malicious activity targeted at computing and networking resources [16]. Various IDSs have been proposed in order to provide continuous monitoring beyond the first line of defence. Many of the earlier IDSs proposed were an off shoot from an original model proposed by Denning [17], with IDES [18] developed by SRI International, being the de facto standard model practically implemented at that time. Many of these models or prototype systems were based on the assumption that an intruder's behaviour will deviate from that of a legitimate user. Hence IDSs employ various techniques to classify indicators of user behaviour, such as systems resource usage from the collected audit trails. The general trend of IDSs can be categorised based on the data source (i.e. audit trails or network traffic data), and intrusion model employed (i.e. anomaly detection, misuse detection or a hybrid approach). These approaches are generally focused on providing continuous monitoring, which involves analysing vast amounts of audit trails.

2.5 Misuse detection model

The misuse detection model is based on an approach that relies implicitly on past experience of security violations. The rules implemented are encoded with knowledge from known past intrusions, known system weaknesses and security policy measures. Using the encoded rules, the audit trail data is monitored for a match or potential security breach, which upon detection, will be flagged for the attention of the system administrator. For example, if an unauthorised user having gained access to a system running a UNIX OS through a Telnet (Port 23) session enters a command something like "`mail < /etc/passwd`" (a command used to launch the mail program which causes the password file to be emailed to the user / intruder) then this can be caught by examining audit logs resulting from the generated audit trails. The key advantage of IDSs adopting the misuse detection model is that they can accurately and efficiently detect known attacks, which have been coded as rules. Therefore by nature, systems adopting this approach are vulnerable to attacks that have no matching signatures. This approach is similar to common anti-virus software, which is only as good as the latest anti-virus update in protecting systems from known viruses. Moreover organisations could become dependent upon vendor provided rules rather than being able to create rules specific to their requirements. Furthermore updating of rules has to be hand-coded and often relies on hypothesis, which tends to become more knowledge intensive and time consuming. In addition the processing time is proportional to the size of the rule-base. There are several approaches used, which differ in the representation as well as the detection algorithms employed to detect the intrusion patterns.

2.5.1 Expert System

The system is based on inferred knowledge by the experts in the domain from known past intrusion scenarios, which are translated into *if-then* rule-base representation. The current activity is then matched against the rule conditions to determine whether the conditions requisite for an attack as specified in the left hand of the rule, hence the *if* part, constitute intrusions. IDES uses a production-based expert system tool (P-BEST), which is rule-based, to encode information regarding known attack scenarios, known systems vulnerabilities, site-specific security information and expected system behaviour. In addition the rule-based detection is a forward-chaining system, hence the audit records are viewed as facts, which map to rules in the rule-base, as opposed to being driven by goals that the user states. Other known systems such as Discovery [19] uses expert systems written in an Artificial Intelligence (AI) shell, while OSIRIS [20] uses Prolog rules to encode intrusion signatures.

2.5.2 Model Based Reasoning

While the rule-base approach attempts to pattern match audit records to rules, model-based reasoning attempts to combine models of misuse with evidential reasoning to support conclusions about the occurrence of misuse. The approach, originally proposed by [21], seeks to model intrusion at a higher level of abstraction than the audit record. Attack signatures are accumulated in a database, where each attack scenario comprises a sequence of behaviours making up the attack. In order to substantiate or refute current activity as misuse or intrusion, audit trails are examined for evidential information to support the reasoning. The approach permits the selective narrowing of the focus on relevant data, so a subset of collected data is examined rather than the collected myriad of audit trails.

2.5.3 State Transition Analysis

In state transition analysis, attacks on a system are represented as sequence of state transition diagrams to model the monitored system [22]. The rationale behind this approach is that any penetration (e.g. penetration scenarios that can be used to illegally acquire root privileges in a UNIX system) is essentially a sequence of actions that leads the target system from some initial normal state to a target compromised state. The *state* is a snapshot of the system representing the values of all volatile, semi-permanent and permanent memory locations on the system. The attack pattern must satisfy the corresponding system state to transit into the respective state. Arcs are used to represent the event / conditions required for the change in state connected to successive states as depicted in *Figure 2*. The model specifies state variables, intruder actions, and defines the meaning of compromised state. USTAT [23], a system based on this approach, has an inference engine, which uses the audit records as means of monitoring penetration-relevant state changes that occur within the system. Therefore the analysis is focused on state changes, rather than pattern matching raw audit records to attack scenarios.

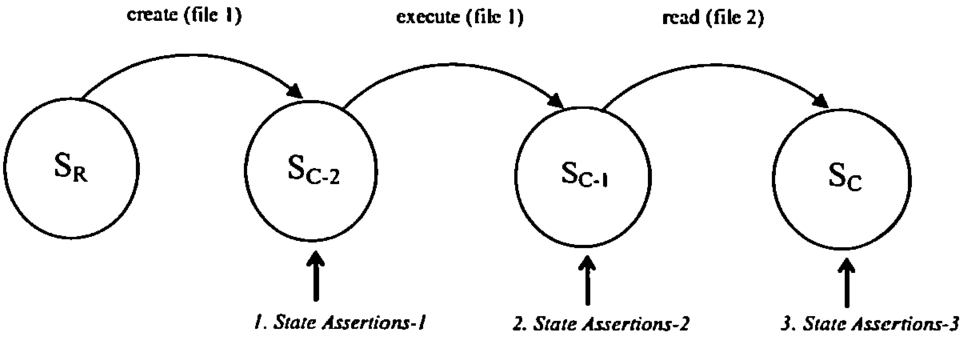


Figure 2: A hypothetical state transition diagram

This analysis approach is similar to the Model-based approach previously described whereby both techniques focuses on a penetration's signature actions rather than the audit records that record the actions. When a sequence actions is found to match the actions represented in a state transition diagram, USTAT then constructs a list of the users who contributed steps within the penetration and instances of this scenarios is then reported to a systems administrator.

2.5.4 Graph Based

While most IDSs described so far are based on detecting intrusions on individual hosts either through distributed audit trail or network traffic collection and centralised analysis, the goal of Graph-based Intrusion Detection System (GrIDS) [24] is to detect and analyse large scale attacks. As such it aggregates network activity of interest, which are then translated into activity graphs. These graphs consist of nodes and directed edges, where the former represents host while the latter represents network traffic between them. They are constructed by user specified rule sets which consists of several sections for example a name, initialisations, preconditions, graph combining rules and assessment and actions. The rule sets are used to decide if two graphs are to be merged or to compute the attributes of the combined graph, and to decide what actions to take if required. For example when a worm infects a network, the network activity associated with its propagation causes GrIDS to build a tree-like graph as depicted in *Figure 3*. This pattern is then recognised as a potential worm using a detection heuristic, which might evaluate the number of nodes and branches in the graph. Counts exceeding a user-specified threshold is then flagged as a "worm". These graphs are evaluated and possibly reported to a systems administrator. The hierarchical architecture of GrIDS allows it to scale to large networks.

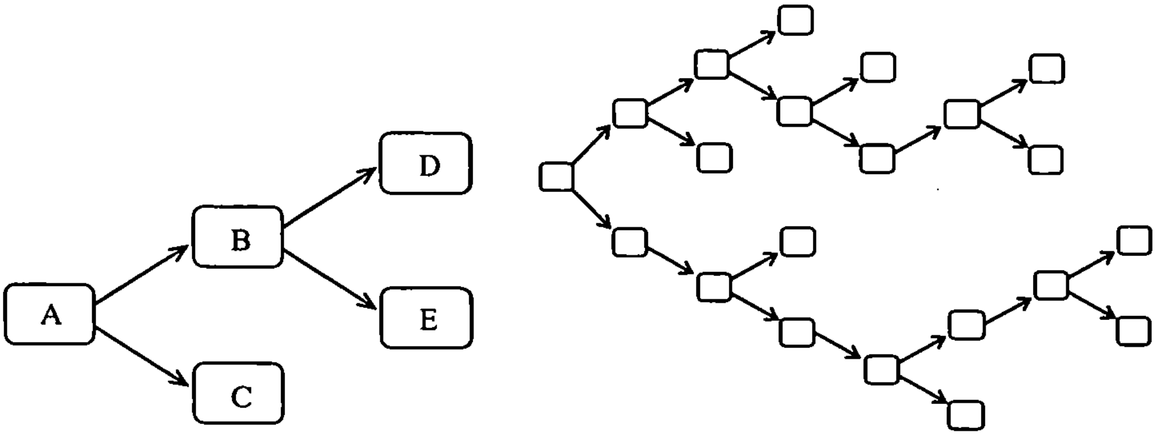


Figure 3: The beginning of a worm graph, and the graph after the worm has spread

2.6 Anomaly detection model

In contrast to IDSs based on the misuse detection model, the anomaly-based intrusion detection model is based on the assumption that misuse or intrusive behaviour deviates from historical norms. The IDSs based on this approach first establishes the normal behaviour, patterns or signatures for users, applications, or other resources of interest in a system such as CPU load, physical memory utilisation, page file, number of processes, and process related information such as creation, activation and termination. Once the profiles of normal behaviour have been established through observing the audit trail records, deviation from these profiles are flagged as an intrusion. While anomaly detection systems can detect unknown intrusion since they do not require any *priori* knowledge about the specific intrusions, the method relies upon intuition and experience in selecting system features. Most anomaly detection IDSs (e.g. IDES, Wisdom and Sense (W&S) [25] and NADIR [26]), are statistical in nature. As such some intrusive behaviour can only be detected by studying the sequential interrelation between events, which can appear to be normal according to the statistical measures. Also, it takes on a "fuzzy" approach since there are no fixed patterns that can be monitored. Furthermore the concept of 'normal' can change over time (e.g. typing style might mature, new applications are used, etc.).

IDES, one of the more influential models developed and implemented by SRI International in the late 1980's which has had quite an impact on the computer security community, uses a profile-based complex statistical methods (e.g. covariance-matrix) in combination with other analysis methods. Multivariate statistical scoring is used where scores are assigned to user sessions to specify the degree to which the user behaves as expected. The profiles store only statistics such as frequency, tables, means and covariances, which represents the description of a subject's normal (i.e. expected) behaviour with respect to the set of behaviour indicative measures (e.g. CPU usage, physical memory utilisation and number of processes). The deductive process is based on evaluating the total usage pattern (i.e. T^2), not just how the subject behaves with respect to each measure considered singly which is expressed in *equation 1*.

$$T^2 = a_1S_1^{\,2} + a_2S_2^{\,2} + \dots\dots + a_nS_n^{\,2} \quad \text{-------}(1)$$

The deductive process is controlled by dynamically adjustable parameters (i.e. a_n), which are specific to each subject measures (i.e. S_n). The single static value denoted by T^2 summarises the degree of abnormality, which is stored in a knowledgebase, is retrieved and compared with the current audit record's vector of indicators of user behaviour variables.

This method of using statistical profile-based anomaly detection is also adopted in the profile engine of other IDSs such as EMERALD [27] and NIDES [28]. SRI International developed the later as a successor to the IDES IDS.

2.6.1 Statistical rule-base

W&S on the other hand is a statistical rule-base IDS that relies on the anomaly detection method to identify system usage patterns that are different from the historical norms. The collected historical audit trails from the OS and processes are developed into rules. The rules are generated from an iterative process of sorting the audit data and examining the frequency of the attribute values. The rules consist of lefthand-side (LHS), which specifies the conditions under which the rule apply and a righthand-side (RHS), which defines what is considered normal under these conditions. The rules incorporated each have grades associated to them, which is a measure of its accuracy. Rules with better (i.e. higher) grades are therefore more specific, or which represent frequent occurring patterns with less variability.

Other methods being researched as anomaly-based intrusion detection models, such as Data Mining [29], are aimed at automating the audit analysis process in identifying system features and latent trends for classifying user behaviour from the collected audit trails. The methodology and algorithms used are central to overcoming the ad hoc and manual approaches adopted in statistical methods as such chapter 3 is devoted to Data Mining, which will be the core methods considered in developing the correlation framework for an advanced authentication scheme, as described in chapter 5.

2.7 Hybrid misuse / anomaly detection

Systems adopting only misuse detection or anomaly detection based IDS model approaches suffer, major problems in detecting certain intrusion patterns. The trade-offs in adopting either approach can be overcome in some ways by combining these detection mechanisms to form a hybrid intrusion detection mechanism. This scheme can be seen adopted by IDSs such as NIDES in order to reduce or elevate the shortcomings of using an independent approach. The approach offers a hybrid as well as independent analysis of monitoring the audit trails. Like IDES, NIDES uses a statistical profile-based module for anomaly detection. The approach employed compares user's short-term profile behaviour to the user's historical or long-term profile to look for deviation occurring, while in parallel a rule-based module generated by the P-BEST tool is used for misuse detection.

Studies reported in [30] give some insight into a practical security approach that can be adopted through the experiment done in modelling the intrusion process. Accordingly it suggests that a typical attacker's behaviour can be categorised into three phases: the *learning phase*, the *standard attack phase*, and the *innovative phase* as depicted in *Figure 4* (taken from [30] for illustration). During the *learning phase*, inexperienced attackers, who are considered to be below some minimal attacking skill threshold, progress on to the *standard attack phase* upon acquiring the relevant skills prior to being able to carry out any malicious activity. Hence they undergo some long-term preparation to attain the relevant knowledge about a target OS limitation, features and vulnerabilities in preparation for the next phase (i.e. the *standard attack phase).*

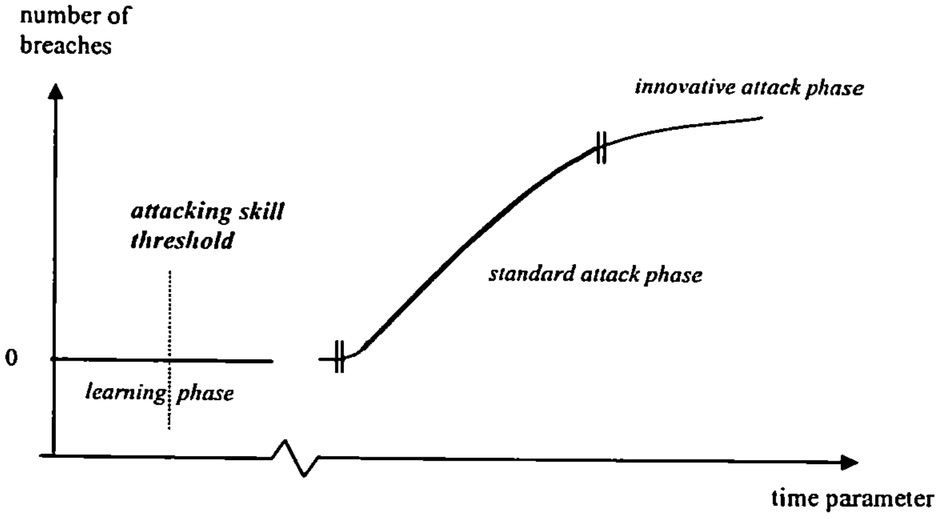


Figure 4: A typical attacking process

In the *standard attack phase*, the attackers are prone to exploit known vulnerabilities using scripts and tools available form the web (e.g. buffer-overflow hack to gain better permissions on a Linux machine). The reported times observed (i.e. during the standard attack) between consecutive breaches from the statistical test data indicates an exponential distribution. This suggests the applicability of traditional methods for reliability modelling (e.g. Markov models). In addition the fairly straightforward actions used to carry out standard attacks suggests a high probability of successful attacks.

Upon exhausting all attempts to attack a target system from the available resources, the attacker moves on to the *innovative phase* where new methods are invented to exploit hitherto unknown system vulnerability. Consequently, the probability for success suggested is expected to be much lower in comparison to the 'standard' attacks as well as a longer time taken for a successful security breach. This could be associated to the trial-and-error factors in inventing innovative approach for a successful breach as well as exploiting or finding new system vulnerabilities. Therefore this would require longer test periods and a larger population of attackers to influence the distribution for this phase besides the attackers experience and learning phase. This reported hypothesis formulated on attacker's behaviour could be used to draw out some important conclusions. For instance while in most cases a misuse detection model can be used to address the 'standard' attacks, anomaly detection model is probably the only method to counteract against the innovative and stealth attacks.

2.8 Shortcomings in current IDSs

Distinguishing user behaviour patterns and classifying it as normal or intrusive is a subtle task. Furthermore exploring the vast amount of audit trail data often yields a small fraction of intrusion or misuse data. While the techniques and detection algorithms used for the *audit analysis* processes varies from the intrusion models employed in IDSs, the shortfalls of these schemes can be addressed in terms of the type of errors that are likely to occur in the system. Therefore this is directly related to the strategies used to correlate the information gleaned from the *audit analysis* stages.

- *False positive errors* – Inherently this is one of the most common problems faced by IDSs. These errors occur due to the misclassification of legitimate behaviour or actions as anomalous. Successive reports of false positives could result in system administrators using this pretext to ignore the IDSs output. This can have serious repercussions when a genuine intrusion is ignored, as well as potentially causing irritation and inconvenience to the legitimate user depending on the type of response mechanism the IDSs employ. Systems incorporating misuse intrusion detection are faced by the subtle task in identifying unique patterns and while it is much more difficult to pick out a valid intrusion attempt, the task is made more cumbersome if a signature also alerts regularly on valid activity. On the contrary, anomaly intrusion detection is prone to this type of errors due to the nature of the method which relies upon intuition and experience in selecting system features. As such any deviation from normal use, which could result from a change of pattern of use by legitimate users, would be flagged as an intrusion and can lead to inundation of these errors on the system.

- *False negative errors* - A more serious threat are when this type of errors occurs. This happens when an intrusive action is not detected and proceeds through the system. This form of error could give a misleading sense of security and, therefore, it is essential to ensure that accurate profiles or signatures of intrusive behaviours are established in order to improve the accuracy of intrusion classification. Attack signatures that are not reflected in IDSs, employing misuse intrusion detection can pose a serious problem if exploited by intruders to gain unauthorised access on the system. There is an extremely good chance that the IDS will not detect it. Likewise this threat is not alien to anomaly intrusion detection since intruders can alter their actions to force *false negative errors* to occur.

Another extremely important issue facing IDSs using current intrusion correlation methods is how much data can the *audit analysis* stages effectively and efficiently analyse. Data that may be logged in an eight-hour period can amount to 3-35MB [31]. While most IDSs tend to filter the raw audit trails, a difficult problem that arises from this is how much can be filtered out without potentially missing an attack.

2.9 Conclusion

It is clear from the above discussions concerning the paucity of current point-of-entry user authentication schemes, whether as a stand-alone device or incorporated as a system wide policy, that there is a need for a more advanced authentication scheme. While IDS's aim to overcome some of these weaknesses by continuously monitoring for signs of unauthorised activity, most of the available IDSs are targeted towards analysing network traffic data in order to protect the system against external abuses. While there are available commercial IDSs using audit trails as the data source such as Axents's Omniguard Intruder Alert [32] and Security Dynamic's Kane Security Monitor (KSM) [33], they work on the basis of attack signatures or misuse detection, which search for compatible signs of recognised scenarios. An intruder's behaviour can vary dramatically and as such, we cannot assume such prior knowledge on all audit data when trying to detect the classes of internal abuses. Hence the need to possibly adopt an anomaly detection approach or a hybrid approach. Most systems currently adopting this approach are optimised using domain expertise to extract only the relevant information from the wealth of available audit data, through ad hoc and manual methods, which can be time consuming and knowledge intensive. This further substantiates the need for an automated approach for the *audit analysis* process to aid in the identification of system features and latent trends that could be integrated into a framework for the purpose of an advanced user authentication scheme. Moreover this is also in relation to the need to analyse the vast amount of generated audit trails in order to obtain user's behaviour profiles.

The need to eliminate the manual and ad-hoc approaches in the data analysis stages in IDSs is attracting interest in applying Intelligent Data Analysis (IDA) techniques. Data Mining, the IDA technique considered have been used in various other domains to address some of the challenges and limitations faced in the *audit analysis* schema currently adopted in the pre-processing stages of IDSs. This is a novel approach yet to be explored in this domain for an advanced user authentication scheme. There is an increasing need for a more coherent paradigm for audit processing in terms of automating the data analysis stages. The current trend of network components providing audit trail or audit logs provides the foundation for IDSs to explore database automated match and retrieval technologies. This can be seen in audit processor components, for instance the SecureView in the Firewall-1 using Data Mart to store the audit trails [16]. This available information could be used for security audit trail analysis in IDSs by utilising the technology in the data analysis stages. The next chapter provides a detailed discussion on Data Mining (DM) algorithms and techniques as an IDA tool to present the key concepts to further the discussion on the proposed advance intrusion correlation framework.

Chapter 3

Data Mining

3. Data Mining

3.1 Introduction

The explosive growth of data in the last decade has consequently outpaced our capabilities for collecting, storing, interpreting and analysing it. This problem is also inherent, as introduced in the previous chapter, to IT security. The process involved in monitoring and analysing user activities to detect the occurrence of intrusions consequently presents the problem of vast quantities of data that may be logged. Current techniques employed as anomaly detection do not scale up to the problem as described in the previous chapter. The approaches used often rely on manual and ad hoc means of analysing the generated data. These methods are focused on extracting only the relevant information that optimises the amount of data analysed. Furthermore the techniques adopted are generally focused on analysing specific data features based on the statistical measures which tends to become knowledge intensive. Therefore there is a need to apply analysis techniques that can automate the process to build models or profiles from the temporal regularities exhibited by users, which could then be used as an input to the anomaly detection process. Besides the automation of these activities it is also essential to ensure that accurate profiles of users are established in order to reduce the false positives and false negative errors. This could be achieved by correlating between adjacent or frequent sequential patterns of user behaviour occurring, which is seemingly intractable using statistical approaches. It is considered important to use a combination of approaches since some intrusions can only be detected by analysing sequential interrelation between audit trails, which an event (e.g. deduction process used in IDES) alone might otherwise appear to be normal according to the statistical measures.

While there is a trend of integrating database technologies into knowledge bases like DBMS, EIS and On-Line Analytical Processing (OLAP) into IDSs, which can provide us with the basic tools for the efficient storage and lookup of large data sets, they do not resolve the issue of how to help system administrators to find patterns in data and analyse large bodies of data. This has led to the quest for a new generation of tools and techniques with the ability to assist humans in analysing the mountains of data intelligently and automatically. Therefore this chapter is aimed at describing these techniques and tools which is the subject of the rapidly emerging field of Data Mining (DM) or Knowledge Discovery in Databases (KDD). The term knowledge discovery in databases was formalised in 1989 [34] in reference to the general concept of being broad and 'high level' in the pursuit of seeking knowledge from data. As such the terms DM and KDD have been used synonymously by many authors to refer to the same knowledge discovery process.

Recent interest in data mining has led to a flurry of new technologies and methodologies based on an amalgamation of techniques from statistics, machine learning, database technology and high performance computing [35]. This chapter aims to provide an introduction to the knowledge discovery process by describing algorithms and techniques that could be integrated into a correlation framework, in order to automate the analysis of the generated raw audit trail data, to the discovery of meaningful patterns and rules in the data. Therefore a detailed overview of the phases involved in the Data Mining methodology is described for the discovery process. The analysis techniques and characterisation of data that can be use to build various data features is included. The latter part of the thesis provides an introduction to some of the methods used in DM, which includes description of some common algorithms used for the *data mining* process.

3.2 The Knowledge Discovery Process

A piece of knowledge is a relationship or pattern among data elements that is potentially interesting and useful. In general, discovery means finding something that is hidden or previously unknown [36]. A knowledge discovery process is then, a system that can automatically determine such patterns and relationships. When a knowledge discovery system operates on a large, real world database or audit logs in our case, it becomes a DM system.

Data Mining can be described as a collection of techniques and methodologies used to explore vast amounts of data in order to find potentially useful, ultimately understandable patterns [37] and to discover relationships. An important element underlining this concept is the focus of building data features from the collected data. This is in contrast to previous techniques employed to analyse the audit trails, which have often used domain expertise to build complex equations in order to capture the relationships between variables or using domain expertise to build the rules to extract only the relevant information. This is important when considering the large amount of variations of systems usage that could be exhibited by different users in networked environment. Therefore relying on domain expertise alone to model this for anomaly detection could prove to be a difficult and time consuming.

The processes followed in data mining are generally the analysis of the system problem, followed by determination of the key data features which will enable the system problem to be resolved [38]. This is to enable the prediction and description of the analysis phases [39] in latter stages of the DM process. For example, in the IT security domain, in order to provide a continuous authentication mechanism, the problem is to determine which behavioural biometric (e.g. keystroke analysis, mouse dynamic, etc.) could be adopted. This is followed by determination of which data features that could be used for instance, users typing speed or common key's used and so forth, which will enable the user to be authenticated on the system.

Prediction typically makes use of existing variables in the database in order to predict unknown or future values of interest. Description focuses on finding patterns describing the data and the subsequent presentation for user interpretation. The relative emphasis of both prediction and description is dependent upon the data mining system used [37]. Several algorithms fulfil these objectives. These algorithms are incorporated into the various DM methods. The process could be represented as depicted in *Figure 5* (Modified from [36] for illustration). It is an iterative and interactive process, involving numerous steps with many decisions being made by the user. The general DM methodology for the discovery process is as follow:

1) Developing an understanding of the proposed application domain, the relevant prior knowledge, and the goals of the end-user.

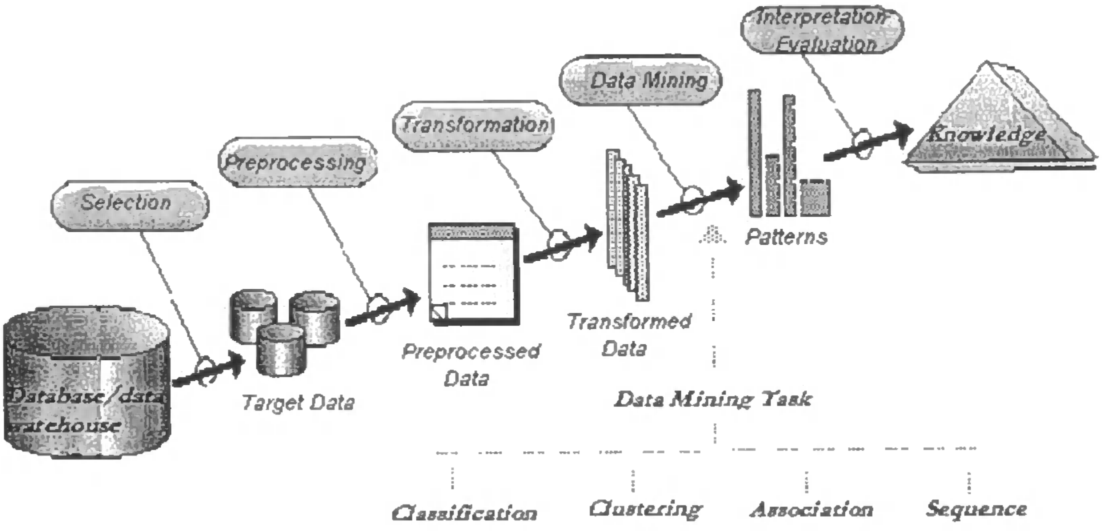


Figure 5: An overview of the steps comprising the DM process

2) Create a target data set.

- Focus on a subset of variables or data samples, on which discovery is to be performed.

- If required recent and historical data can be found, combined, and transformed into an organised target data set repository.

3) Clean and pre-process the data set if required.

- Remove redundant or erroneous data, which the statisticians term 'outliers'.

- Involve processes to quantify continuous features, for instance, as used to generate the audit trail semantic or demarcation of the beginning of each message (e.g. removal of time stamp from audit trails).

- Determine strategies for handling missing values

- Explore data analysis using summarisation tools, graphical methods or statistical analysis.

- Building the database based on the evaluation method used, for instance train and test, cross-validation or bootstrap method (see section 3.4).

4) Find useful features to represent the data depending on the goal of the task by reducing and projecting the data.

- Use dimensional reduction or transformation methods to reduce the effective number of variables under consideration.

5) Decide whether the task of the DM process is classification, association, regression or clustering.

6) Choose the data mining algorithm(s), depending on the DM task.

- Decide on the appropriate models and parameters, which matches a particular data mining method.

7) Search for patterns of interest in a particular representational form or model.

8) Interpret the mined patterns.

- Involves trying to understand the discovered patterns.

- Return to any steps 1-7 for further iteration using different settings if required, removing trivial patterns and presenting the useful patterns to users.

9) Consolidate discovered knowledge.

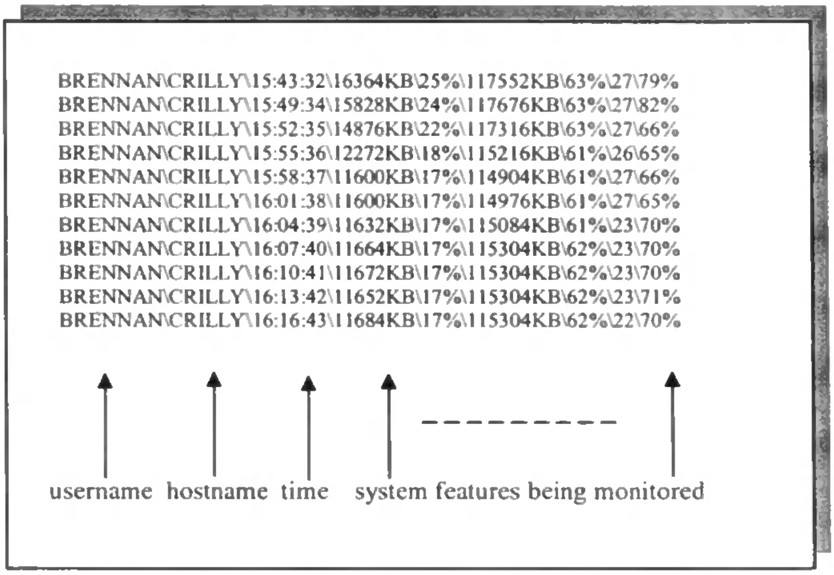
- Incorporate this knowledge into the performance system, or for documentation and reporting.

3.2.1 Data Mining pre-processing stages

The representation of the DM process depicted in *Figure 3* is an analogy to the representation of a supply chain for a manufacturing company. The latter could suffer severe consequences if any short falls or bottlenecks should occur in the chain leading up the end user, while the former will misinterpret the data sets if the initial stages leading up to the data mining task is not clean (containing reliable examples). This is important since data mining depends heavily on the quality [40] and quantity of the data provided. While the vast amount of audit trails generated and stored in audit logs provides an ideal condition to employ DM, which are frequently so large that they not only preclude direct human analysis, they are also very demanding of computer power to find patterns in this myriad of raw data. Therefore in this case it is unusual to mine the audit logs directly when considering the number of possible relationships is immense and the information might be incomplete or corrupted More typically mining is carried out on a subset of the data usually stored as data warehouse or data mart. A data warehouse or data mart is a repository of information, typically historical data that provides an accumulation of data that can be reviewed for analysis. The data is normally organised by subject instead of application which contains only information necessary for decision support processing like for example: computer systems administrators would organise their data by user name, time stamp, host name and event type (i.e. system features being monitored).

The pre-requisite for accurate and efficient data mining is having data with clean or reliable information. In order to obtain this, a successful transformation of data from the raw data set or from the data warehouse is essential. To enable this task, understanding of the application domain, the relevant prior knowledge and the goals of the end-user are required in order to accomplish the selection and preparation of the data. This initial phase of data selection and preparation, is normally termed *pre-processing* and can be used to constrain the search space, and can make patterns in the data more visible [41] in the later stages of data mining, using the relevant domain knowledge. The phase of data preparation and transformation takes a huge amount of time in the knowledge discovery process (typically 80% to 90% of the overall process spend on a DM project).

In analysing audit trails, for instances, generated from computers in a networked environment to monitor for anomalies (see *Figure 6*) occurring on the system. If the goal, is to detect the occurrence of an intrusion on the system, whereby the approach relies on a continuous user authentication scheme, the objective is then to mine the behavioural indicative data from the audit trails to identify the patterns that could be ideally categorised as 'normal' of 'abnormal'. In order to be useful, the recent and historical data can be found, combined and transformed into an organised repository target data set or data warehouse. Based on the task definitions and goals, data are selected from the data warehouse. These data are complied into pre-processed data repository. In order to insure the accuracy of the data, hence, removing audit trails carrying redundant information, cleaning, validation and completion process are performed to develop the target databases (Similar process to building the data warehouse).



```
BRENNAN\CRILLY\15:43:32\16364KB\25%\117552KB\63%\27\79%
BRENNAN\CRILLY\15:49:34\15828KB\24%\117676KB\63%\27\82%
BRENNAN\CRILLY\15:52:35\14876KB\22%\117316KB\63%\27\66%
BRENNAN\CRILLY\15:55:36\12272KB\18%\115216KB\61%\26\65%
BRENNAN\CRILLY\15:58:37\11600KB\17%\114904KB\61%\27\66%
BRENNAN\CRILLY\16:01:38\11600KB\17%\114976KB\61%\27\65%
BRENNAN\CRILLY\16:04:39\11632KB\17%\115084KB\61%\23\70%
BRENNAN\CRILLY\16:07:40\11664KB\17%\115304KB\62%\23\70%
BRENNAN\CRILLY\16:10:41\11672KB\17%\115304KB\62%\23\70%
BRENNAN\CRILLY\16:13:42\11652KB\17%\115304KB\62%\23\71%
BRENNAN\CRILLY\16:16:43\11684KB\17%\115304KB\62%\22\70%
```

Figure 6: Example of generated audit trails collected from networked computers

At this stage if the operation is to validate a hypothesis, then OLAP tools can be employed. If this is a data mining effort, the data are analysed (mined) for patterns using mining algorithm(s) and the results are presented. The results from the data mining are then subjected to other evaluation methods, from for example, presentation of results in a summary text document, or visualisation tools, to determine whether previously unknown information exists. Data Mining techniques typically draw upon methods from fields such as pattern recognition, machine learning, database management, statistics, knowledge acquisition for expert systems and data visualisation.

3.3 Data Mining techniques - building data features

The discovery process involves mining the sorted and graded (refined or examined base patterns) which requires a process involving tools and programs for sampling and assaying the material to determine the predefined objective. The initial sequence at this stage is to determine the data mining task. Data Mining algorithms provide the technology to accomplish data mining tasks. Different data mining tools and different algorithms are optimised based on the predefined data mining task. This involves deciding whether then goal of the DM process is classification, clustering, association or sequential.

3.3.1 Classification / clustering

Classification has two distinct meanings. We may define a set of observations with the aim of establishing the existence of classes, or clusters in data. Or we may know for certain that there are so many classes, and the aim is to establish a rule whereby we can classify a new observation into one of the existing classes. The former type is known as Clustering (referred to as Unsupervised Learning by the AI community), the latter as Classification (referred to as Supervised Learning by the AI and discrimination by the Statistical community) [42].

In detecting a perpetrator on a system for example, sufficient "normal" and "abnormal" audit trails for users or application could be collected. Classification algorithms (e.g. through rule induction) could then be applied to the categorised data sets to learn "classifiers" that can label or predict unseen audit data as belonging to the normal class or the abnormal class. The classifiers could then be integrated into an anomaly detection mechanism to monitor for deviation or anomalies occurring on the system.

The classification rule could be: -

```
if  <condition>
and <condition>
        |
        |
        |
and <condition>

then normal

else

then abnormal
```

This rule could then be used to apply to new data sets for classification.

If this is a clustering task the process is like classification except that the classes are not normally known before hand. For example a set of attack signatures can be grouped together according to some similar criteria. This technique will divide the audit trails according to their signature patterns, thus creating a set of groups, which have the maximum similarity within them and the maximum difference between them, creating a profile of possible attack signatures. These clusters could then be used to predict attacks, which might be missed out by IDSs employing for instance statistical approaches, which are mainly used for evaluation of a given hypothesis, or for the systematic fitting of a given class of models [43]. In other words clustering involves defining homogeneous from heterogeneous data sets.

3.3.2 Association

Association algorithm looks for relationships related to a particular event occurring. These algorithm attempts to generate rules, which state that, when a particular event occurs then another type of event also occurs in a certain percentage of cases. Association attempts to generate rules or discover correlation in data and is expressed in *equation 2:*

$$X \Rightarrow Y, \text{ where X and Y are sets of items } \text{-------}(2)$$

This means that an event or a transaction of a database that contains X tends to contain Y.

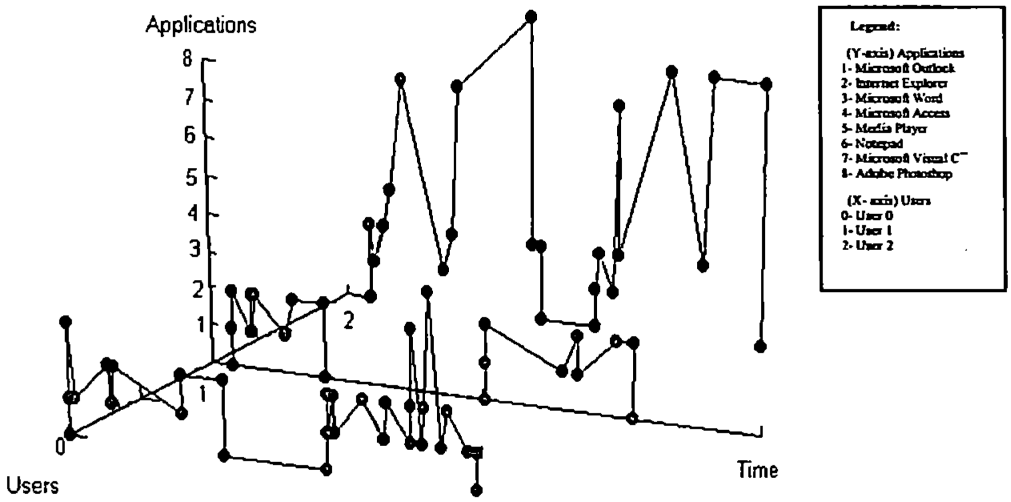


Figure 7: Graphical representation of applications run by users

Correlation of system features, for example, the correlation between applications frequently occurring from the interaction (i.e. in an active session) of users with the system, can serve as the basis for constructing normal usage profiles. This pattern, once identified for example as depicted in *Figure 7* for User 1, which is the legitimate user in this case, could be incorporated into an anomaly detector framework in conjunction with other key indicators of user behaviour. Therefore the legitimate user patterns when compared against other usage patterns of users interacting with the system could potentially be used to identify unauthorised access or intrusion.

Typically the association rule derived, from data mining on a subset of the audit log containing the generated audit trail could be: -

```
75% of all audit trails that contain application A and
application B also contains application C.
```

The number '75%' refers to the confidence factor, a measure of the predictive power of the rule.

3.3.3 Sequential

Sequential looks at events occurring in a sequence over time or time-ordered sequences. This could be expressed through the following: for E ϵ N, E is a set of event types, while N is a set of positive integers (or referred to as natural numbers in the study of set theory) and is denoted as following: N = {1,2,3,.....}. An event is a pair (A, t), where A ϵ E is an event type and t represents the time of the event or occurrence of an event. This is followed by predefined sets of possible intrusion classes where, C is a set of intrusion classes and I ϵ C, I is an intrusion type, hence for example:

```
90% of the time, if the event (A, t) occurs, it is
followed by intrusion type I.
```

3.4 Methods of comparison

In order to test the classifiers or rules prediction accuracy, the sample data sets are normally split into two parts a training set and a testing set. The algorithm is subjected initially with the training set to build "classifiers" and then the unseen test data or testing data is used to test the classifiers accuracy in predicting or labelling the data correctly. Typically the three common evaluation methods used are train and test, cross validation and bootstrap [42].

3.4.1 Train and test

This technique is generally used for large sample sizes, typically, those greater than 1000 samples. Using this technique a classification rule is tested on a sample of data whose classification is known (but unknown to the classifier). The predicted and true classification on this test data gives an unbiased estimate of the error rate of the classifiers. There is a slight loss of efficiency because the classifiers are not trained on the complete data set, but nevertheless since this technique is used for large data sets, this is not generally perceived as a problem.

3.4.2 Cross validation

This techniques in normally used for moderate sample sizes, typically, those with sample size of between 100 and 1000 samples. In its most elementary form, cross-validation consists of dividing the data sets into *m* sub samples. Each sub samples is predicted via the classification or predicted rule from the remaining (i.e. *m*-1) sub samples. The method is intended to avoid the possible bias introduced by relying on any one particular division of the train and test samples. In this way the error rate is estimated efficiently and in an unbiased way. The rule finally used is calculated from the average score over the different partitions.

3.4.3 Bootstrap

This method is the best approach to examine the error comparison for small data sets, typically, those less than 100 samples. The general aim of this method is to re-use the original data sets, of size *n*, in order to obtain a new data set (of size *n*) by re-sampling with replacement. In the context of classification, the bootstrap approach is to replicate the whole classification experiment a number of times and to estimate bias from these replicate experiments. In order to estimate the error rate in a sample (i.e. of size *n*), a large bootstrap replicate samples *B* are created, each being a replicate of the original sample. These sample sizes are taken randomly from the original by sampling with replacement. This means that some sample points will be omitted (on average 1/e = 37% of data will not appear in the bootstrap samples) and some data point will appear more than once in the bootstrap samples. The average bootstrap rates of the bootstrap sample are then combined in order to give and estimate of the error rate for the original rule.

3.5 Data Mining algorithms

There are a number of methods and data mining algorithms available, which depends strongly on the kind of data, domain and the intended use for the mined knowledge. Is the model intended to be predictive or explanatory? Should the patterns discovered be understandable by people, or is reliability the most important consideration? Any Data Mining algorithm is composed of 3 general [39] components: model representation, model evaluation and search method. The model should represent flexible limits and assumptions clearly, so that patterns can be discovered; the model should have predictive validity - which can be based on cross validation; and, the search should optimise the model evaluation criteria (classification, association and etc) given the observed data and the model representation. The 'mining' tools or search 'engines' are usually automated smart programs incorporating some form of artificial intelligence in the relational database. These programs detect predefined patterns and alert users of variations. There are four main categories of data mining algorithms, Neural Networks, Machine Learning, Statistical and Hybrid (multitude of combinations of different classes of algorithms) [42].

3.5.1 Neural Networks

The field of neural network emerged from diverse sources namely from the fascination of mankind with understanding and emulating the human brain.

Rumelhart and McClelland introduced the back propagation supervised learning algorithms or Multilayer perceptrons (MLP) [40]. Like the brain, the neural network based on the MLP consists of layers of neurons. These are collections of connected nodes with inputs, outputs and processing at each node as depicted in *Figure 8*. A number of hidden processing layers exists between the visible input and output layers. A neural network can be trained for example, to identify the risk of intrusion from a number of factors (see *Figure 8)*. The neural model has to train the net on a training data set and then use it to make predictions. The network learns by iteratively adjusting the connection weights until the outputs are sufficiently close to the training data. The problem with neural networks is that no explanation of the results is provided (black box operation) [43]. This inhibits confidence, acceptance and application of results.

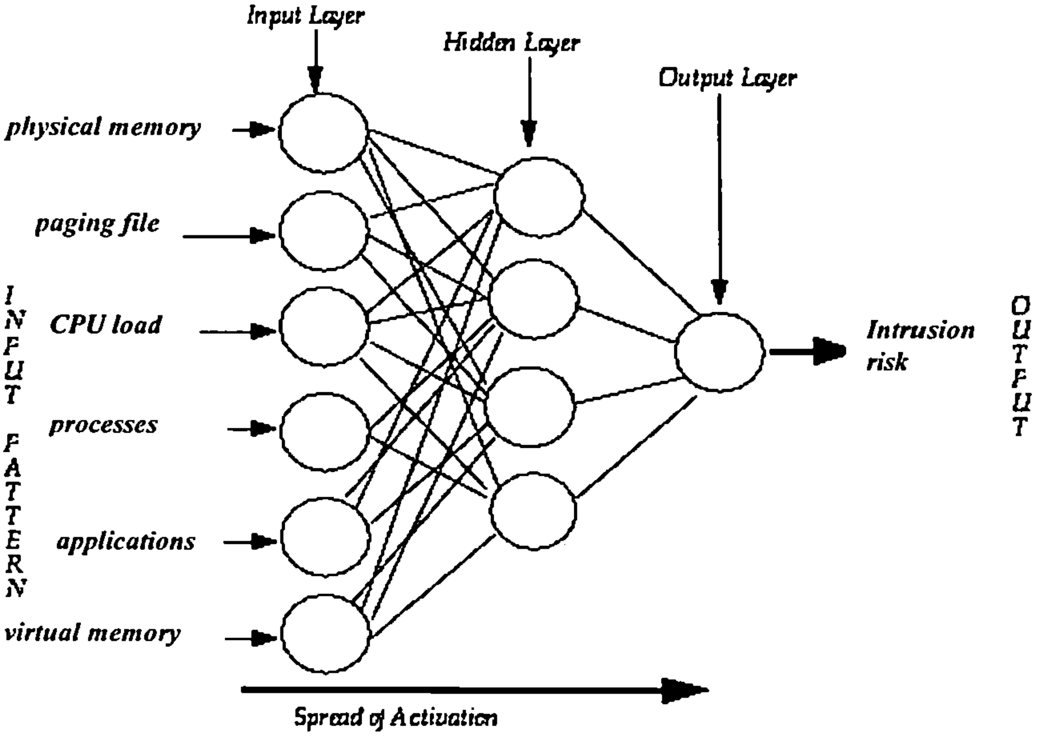


Figure 8: Example of a neural network model

The Radial Base Function (RBF) neural net is another example of a supervised learning algorithm. It consists of a layer of units performing linear or non-linear functions of the attributes, followed by a layer of weighted connections to nodes whose outputs have the same form of target vectors. The RBF has a structure like an MLP with one hidden layer, except that each node of the hidden layer computes an arbitrary function of the inputs (with Gaussians being the most popular). The hidden layer has parameters appropriate for functions being used e.g. Gaussian widths and positions.

3.5.2 Machine learning

The field of machine learning was conceived nearly four decades ago with the bold objective to develop computational methods that would implement various forms of learning, in particular mechanism capable of inducing knowledge [44] from examples of data. The Machine Learning algorithms use automatic procedures based upon logical or binary operations in order to learn a task from a series of examples.

The majority of these algorithms employ decision-tree and rule induction approaches in order to generate classifying expressions simple enough to be understood. These algorithms can be implemented without using expert intervention, although expert knowledge may be used. The algorithms commonly included in this class are C4.5, C5.0 which are decision tree based algorithms and CN2, ID3 and AQ15 which are rule–based algorithms.

3.5.2.1 Decision Trees

This application segregates the data based on values of the variables. This methodology uses a hierarchy of if-then statements to classify data (see *Figure 9*). The major advantage in this application is that it is faster and more understandable than neural nets [42]. However, the major drawback is that data type has to be interval or categorical [37]. Continuous data will then have to be recorded into these two data types, thus bringing out the possibility of concealing significant breakpoints in the data. The *if-then* statements could also be complex, especially if the condition list is long.

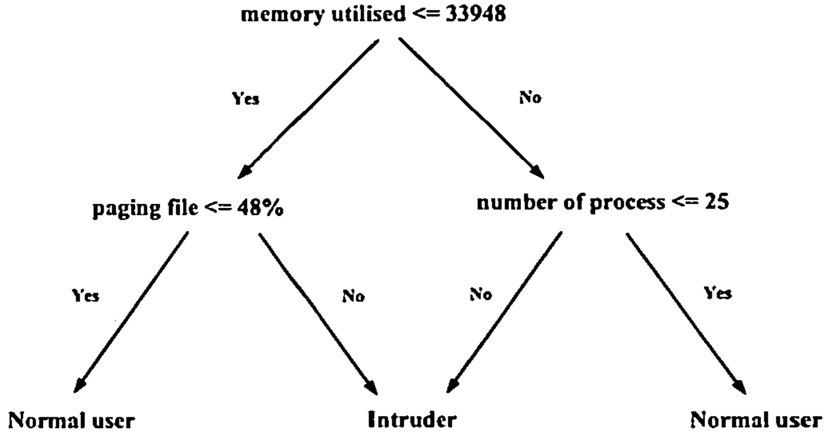


Figure 9: Example of a decision tree modal

C4.5 is an example of a decision tree algorithm [45]. It uses a modified entropy based measure to calculate the gain in information. In other words, entropy is used to measure how informative a node is, by splitting the data at various decision boundaries. The first stage of C4.5 generates a decision tree as shown in *Figure 9*. Each level of the decision tree represents a split of the data set. Examining each possible split of the data, the attribute which best splits the data is selected. C4.5 uses a bottom up search, based on iteratively merging of groups. The second stage of the C4.5 algorithm prunes the decision tree for generalisation. That is, C4.5 evaluates the consequences of merging every pair of group. This process stops when two value groups remain or until the gain cannot be further improved by merging.

OC1 is another example of a decision tree Machine Learning based algorithm [46]. Unlike C4.5, which makes its decision on various boundaries based on single attributes (termed *oblique* decision), OC1 uses linear combination (i.e. it creates a new variable by combining other variables, for example, if $Y = 2X_1 + 7X_2 + 3X_{3,}$ then Y is a linear combination of the variables X_1, X_2, and X_3), of attributes in decision-making termed *non-oblique* decision. Therefore it builds decision trees that contain linear combination of on or more attributes at each internal node. These trees then partition the space of examples with both, oblique and axis-parallel splits. OC1 incorporates a number of features intended to support flexible experimentation on real and artificial data sets.

3.5.2.2 Rule Induction

A set of non-hierarchical sets of conditions will be generated, which will then be used to predict values for new data items. Certain software applications tend to evaluate and refine the rule-set, by selecting the best rules for prediction so as to avoid rules that overlap. The rules used for prediction are more general and more powerful than decision trees, utilising predictive forests (with many partial decision trees) with extended ranges of values [44]. These predictive models are fully transparent and provide full explanations for their predictions.

In keystroke analysis, for instance, a potential problem could be to learn rules from the audit trails to authenticate users on the basis of a known typing pattern. Typically, the keystroke digraph latencies, which is the intervals between the keyboard character pair, or digraph typed (e.g. the time between releasing one key and pressing the next) and the mean inter-keystroke times (i.e. the average time of all keystroke patterns) could be used as the attributes or discriminators in this case. An induction technique can then be used to produce a symbolic classification model, which could generates a rule stating:

```
if    average inter-keystroke <= 25ms, latency between
      digraph EF <= 35ms, and latency between
      digraph CD <= 48ms,

then  <legitimate user on the system>
```

CN2 is an example of this class, which also uses an entropy-based measure to guide the choice of rules [48]. However, it uses a test of statistical significance to decide whether the improvement is worth accepting. It generates rules using a generic-to-specific search strategy, which terminates when no specialisation produces an acceptable gain.

3.5.3 Statistical Approaches

Two main phases of work on classification methods based on statistical approach can be identified, these are the "classical" phase concentrated on derivatives of Fisher's [42] early work on linear discrimination and the "modern" phase which exploits more flexible classes of models. This modern phase attempts to provide an estimate of the joint distribution of the features within each class, which can in turn provide a classification rule, for example k-Nearest Neighbour (k-NN) algorithm. The statistical approach provides a probability of being in each class rather than simply a classification. They are generally characterised by having an explicit underlying probability model. In addition, it is usually assumed that the user of this technique has a strong statistical background, hence some human intervention is assumed with regard to variable selection, transformation and overall structuring of the problem.

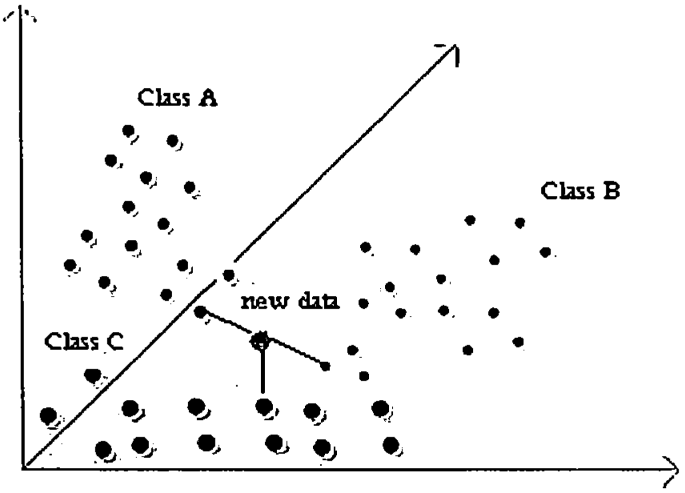


Figure 10: Example of the k-NN classifies based on euclidean distance

The k-NN is an example of a statistical algorithm, in which each sample with n attributes is viewed as a point in n-dimensional space as illustrated in *Figure 10*. The distance between each new test point and all previously seen training points, distance is calculated. The point with smallest distance is termed the nearest neighbour. The exemplar is assigned to the class that is most popular among the K-neighbours, in this case A, B or C. K is usually chosen to be odd when it is required that equal probabilities are not possible. The distance between exemplars is usually taken to be the Euclidean distance.

COG is another example of a statistical algorithm. In contrast to k-NN, COG effectively calculates the "centre of gravity" of all the members of each class within the data set. Therefore any new point within the nearest class centre is then predicted from the established points.

3.6 Conclusion

The nature of monitoring involves the complexity of learning patterns from audit trails to detect intrusive and suspicious activities on computer systems. As such, the task has often been carried out through manual and ad hoc approaches, typically, through complex statistical methods, to build models or rules, in order to establish normal behaviour patterns profiles that can be used against to detect anomalies. Therefore it is clear from the discussions above, that data mining techniques and algorithms have the potential to provide a systematic approach to this problem. The data mining techniques in building data features through the DM algorithms can be used to automate the processes involved to compute consistent patterns from the audit trails. This is important since the patterns can be used to guide the data gathering process, the feature selection process as well as discovering intrusion patterns.

While statistical anomaly detection algorithms in current IDSs are being built on to address the issue of scalability, an important issue that arises is how data features such as correlation between adjacent or frequent sequential patterns of user behaviour can be extracted. This is considered important as shown in the examples using DM techniques because some intrusions can only be detected by analysing sequential interrelation between audit trails. Therefore this is further evidence of the potential of using DM techniques that can be applied to audit trails to learn "classifiers" that can detect these patterns.

Another important element that can be observed is the interpreted rules obtained from the data mining process. The systems features outlined by the classifiers to detect anomalous behaviour can be used to detect known intrusions. The potential of using the DM techniques for automating or semi-automating the means of analysing large quantities of data, is presented in chapter 4 through the comparative studies carried out on the various learning algorithms used in DM. It then proceeds to describe a conceptual advanced intrusion correlation framework for user authentication and monitoring using these DM techniques, which forms the basis for future research work.

Chapter 4

A Comparative Study of Behavioural Profiling Techniques

4. A Comparative Study of Behavioural Profiling Techniques

4.1 Introduction

The chapters to date have addressed the need for a continuous user authentication scheme, and the prerequisites of achieving this objective through appropriate monitoring and analysis of user activity within an active session. As this would involve analysing vast amount of audit trails and learning temporal regularities of users, a dilemma faced by current approaches, we purposed the use of data mining techniques and methodology by introducing the principles and concepts in chapter 3. This was in order to provide the background to further the discussion on a conceptual correlation framework in the next chapter. Besides the issues surrounding the *audit analysis* stages, it is also essential to gather as much information as possible pertinent to a user's interaction with the system. This is in order to build user profiles to distinguish between similar behavioural patterns of users that could occur. This brings about the importance of behavioural indicative data, an important issue yet to be considered in this thesis, although it has been introduced briefly through the examples used to elaborate some of the principles of DM.

This chapter proceeds with some results on the comparative studies carried out on the various learning algorithms used in DM to build temporal regularities from the resource usage of users in an active session. Some results and implications of the method are provided. Although some reported work has been carried out to analyse network traffic data using DM [49], none has been carried out to compare the feasibility and effectiveness of learning techniques (e.g. neural network, machine learning, etc) for the analysis of audit trails, nor using DM for the purpose of continuous user authentication. Therefore the comparative studies carried out are considered novel in this area and part of the work described in the proceeding sections has been published [50, 51]. The potential of using DM is further substantiated with a comparative study carried out using a statistical approach for detecting deviation from a user's historical keystroke profile captured under a multi-tasking windowed environment.

4.2 Comparative study on behavioural profiling using Data Mining

The work aimed to assess the use of DM to automate the data analysis process in identifying system features and latent trends for classifying user behaviour from the collected audit trails. This approach is based on the assumption that a user's behaviour has regularity and that using the classifiers these temporal regularities can be modelled. The patterns reflected in the DM algorithm classifiers (e.g. through rule induction), could then be used to recognise deviation, if it occurs, from normal use. Using this analogy, anomalous behaviours can then be categorised as a possible unauthorised user or use of that system. For the purpose of this investigative work, the audit trail data analysed was collected from networked computers on a participating local area network (LAN) running

Microsoft's Windows NT 4.0. This represents new work in the context of using Windows NT as the OS to study the implications of a continuous user authentication scheme. The networked computers were restricted to single users in order to create the user behavioural profiles. In order create these profiles, an independent agent was installed locally on ten networked computers that participated in these trials to enable the auditing of the users interaction with the system. This was done using a system-wide hook and Windows API functions. The system-wide hook was used to allow a specified code block (hook-function) to receive the appropriate Windows messages (e.g. SHELL_WINDOWCREATED for notification of top level shell or applications created) irrespective of the target application (i.e. it is possible for a hook function residing in a system DLL to receive notifications for all currently running applications). Windows API function, specifically, the GlobalMemoryStatus command was used to obtain the system's current usage of both physical and virtual memory. The collected audit trails were stored remotely on the server instead of locally in order to overcome the possible limitations of local storage capacity.

The audit trail collected is based on the assumption that users performing their regular tasks will impose similarly regular demands upon system resources. Hence system features such as CPU usage, physical memory utilisation, paging file, number of processes and process related information such as creation, activation and termination, etc, involved for continuous monitoring of user interaction with the system are audited. Similar system features have been used in other reported work [18]. However, previous work has focused on statistical and neural network (although no reported work as been published using the learning algorithms studied). The audit trails generated are polled at fixed intervals except in the case of process related information in order to reduce the load on the networked computers. A user's behaviour profile can be uniquely identified by

```
<user name, absolute time, date, hostname, event1,....
eventn>
```

which is the semantic used for the audit trail where, *events$_n$*, denotes the system features being monitored. Notes relating to the associated experimental trials are presented in Appendix B.

4.2.1 Methodology for audit trail analysis

The methodology used is derived from the four main activities of DM; *selection*, *pre-processing*, *data mining* and *interpretation*, and is as depicted in *Figure 11*. The collected audit trail is split into various sample sizes. These subsets form the target data sets, which will undergo the analysis to identify patterns and to test specific hypotheses. The cleaned data, containing both categorical and numerical data, is then subjected to analysis by the DM algorithms. The Intelligent Data Analysis (IDA) Data Mining Tool [52] is used to analyse the sample data sets. This incorporates algorithms from the fields of Statistical, Machine Learning and

Neural Networks. Six algorithms, k-NN, COG, C4.5, CN2, OC1 and RBF were chosen for this investigative work (see the previous chapter for description of these algorithms).

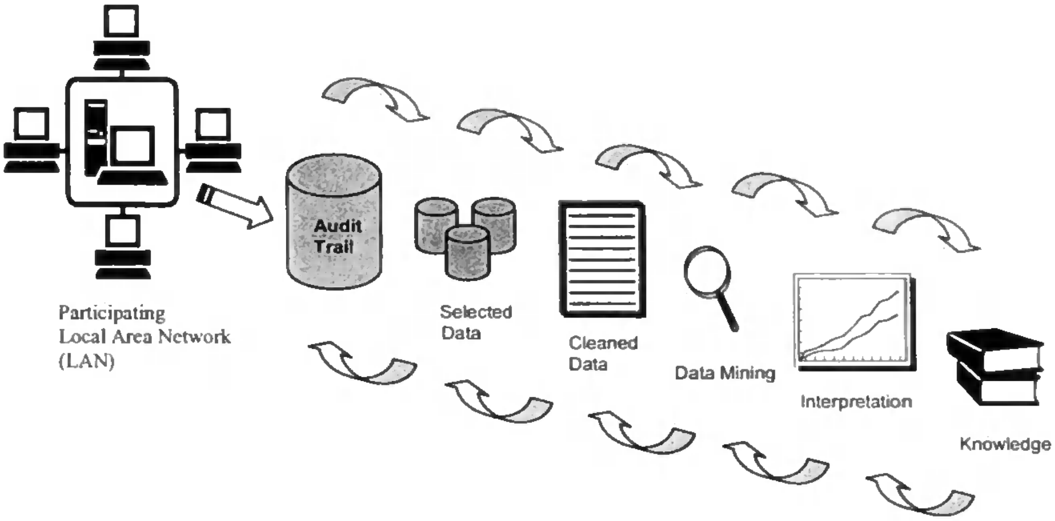


Figure 11: Methodology for behavioural profiling

4.2.2 Evaluation criteria

Depending on the evaluation method (as described in the previous chapter), the sample data sets collected from the networked computers are split into various ratios. For the purpose of this work the train and test evaluation method was used. This method was chosen since the sample sizes analysed were greater than 1000 samples. The data sets were split into ratios of 9:1, 8:2 and 7:3, hence into two parts; a training set and a testing set. The varying data sets ratios were used to avoid the possible bias introduced by relying on any one particular division of the data samples. This approach is similar to cross validation evaluation methods but instead of taking the average score, the lower accuracy exhibited by the three sample ratios was used. This was in consideration of the security implications if average scores were used, since it would not give a clear indication of the actual accuracy in classifying users behaviours, in this case, as being intrusive. Therefore, the error rate could be estimated efficiently and in an unbiased way. The algorithm or classifier is subjected initially with the training set and then the classification accuracy is tested using the unseen data set or testing set. The results give an indication of the error rate (or false positives) and the overall classification accuracy of the trained algorithms.

4.2.3 Initial results

The high classification accuracy obtained from these initial results as depicted in *Figure 11*, for some of the DM algorithms, suggest that the algorithms can learn models or classifiers from the temporal regularities exhibited by users to classify the unseen audit data as belonging to the legitimate users. These have important implications for the purpose of continuous monitoring and user authentication, since the result shows the classifiers ability to automated the process involved in extracting the latent trend which otherwise would have to be modelled through traditional statistical approaches. The high accuracy obtained for some of the algorithms also indicates that the data features monitored could be used, in combination with other behaviour indicative data, as the basis to build user profiles for later comparison during the authentication process.

In terms of the DM algorithms performance, the comparative studies carried out using the four data sets having different proportions of sample with a fixed number of classes (i.e. 10 users in this case), suggest that Machine Learning and Statistical-based algorithms are better for these types of data sets (see *Figure 12*). C4.5 and OC1 decision tree based algorithms in particular, out-performed the CN2 rule-based and RBF algorithms.

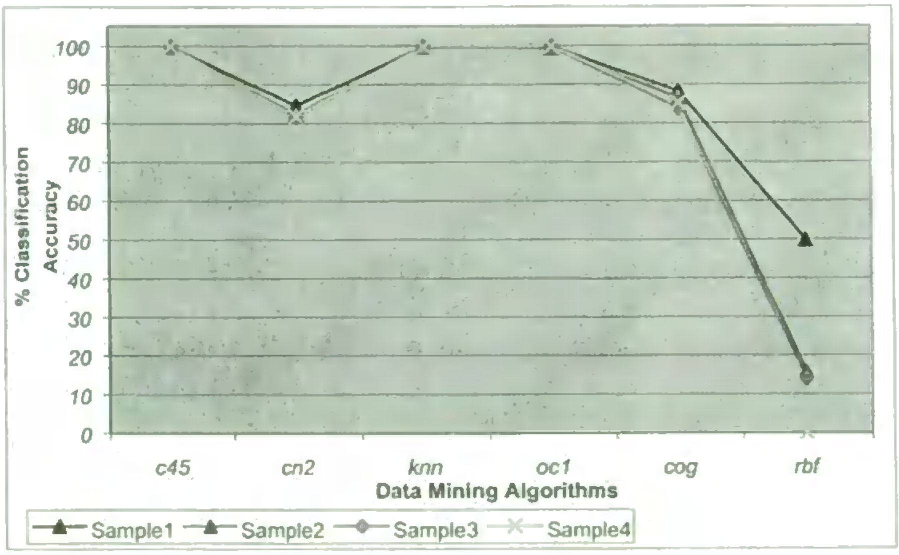


Figure 12: Percentage classification accuracy of selected DM algorithms

The output results generated from the *data mining* process using the DM tool, for example when using C4.5, could be as depicted in *Table 2* (this is an extraction from the list of rules generated). However despite the slower classification times observed using k-NN in comparison to C4.5, the classification accuracy obtained shows some significance for further investigative work. Amongst the statistical algorithms, k-NN faired better then COG, but is slower in comparison to the classification times observed. The classification accuracy obtained overall depicts RBF classification accuracy as inverse proportional to the sample sizes. These results support other reported work that suggests RBF is commonly suited to situations where there are only a small number of classification classes [42].

Rules	Interpretation
Attr 5 <= 25 : 1 (1082.0)	If number of process <= 25 Then User 1
Attr 5 > 25 :	If number of process > 25 And
| Attr 3 <= 48 :	If paging file is <= 48% And
| Attr 4 <= 33948 : 2 (1084.0)	If memory utilised <= 33948 Then User 2
| Attr 4 > 33948 : 3 (431.0)	else
	If memory utilised > 33948 Then User 3

Table 2: Rules generated by the C4.5 algorithm

In addition to the consistency in classifying the data sets and the overall average classification accuracy, as depicted in *Figure 13*, our initial investigations also identified that C4.5 has an overall quicker train and test time and outputs explicit rule representation, which could be easily integrated in a correlation framework for real-time monitoring.

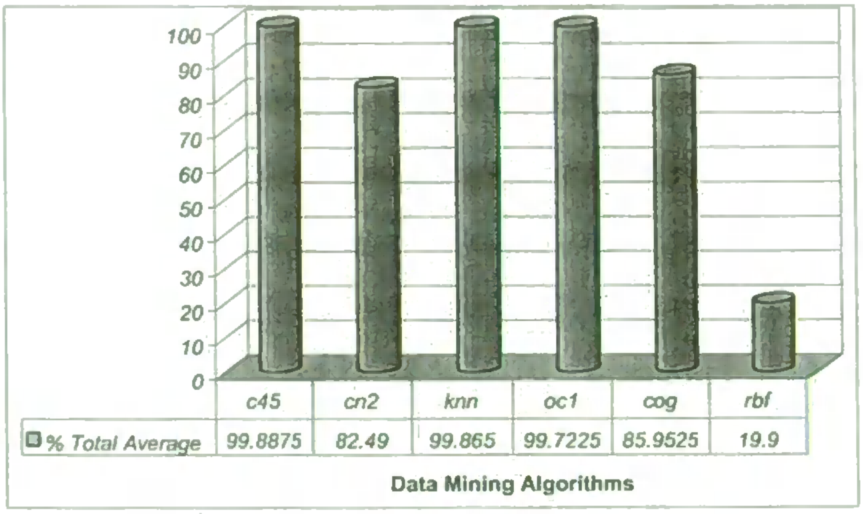


Figure 13: Total percentage average classification accuracy of selected DM algorithms

4.3 Comparative study on keystroke data analysis using data mining and statistical approach

Keystroke analysis is an example of a biometric, which uses inter-keystroke latencies (time between keystrokes) to differentiate between users. The idea of using keystroke for user authentication is not new, and there have been published papers in this area. However, previous work has been based on neural network [53], bayes classifiers [54], and statistical [55] approaches to analyse the keystroke data. Furthermore these approaches have been based on enhancing the initial login-stages for user authentication where the user's typing was monitored in a controlled environment. As such, no research work specifically focussed on continuous dynamic keystroke analysis and in an uncontrolled environment has been reported. Therefore, it is considered that there exists the scope for using DM techniques. While in the previous section the aim was to show the feasibility and effectiveness of DM learning algorithms in building temporal regularities in user behaviour, this section is intended to build upon the findings by comparing against statistical approaches. Therefore it is not intended as a through analysis of keystroke latencies for user authentication. The experiment was designed to allow keystroke data to be collected under the Microsoft Windows NT environment.

In order to collect the required data, an independent agent installed locally on the networked computers was used for acquiring keystroke notifications across all applications running within a users' active session. This was done using a system-wide hook to receive the appropriate Windows messages (i.e. similar approach to the agent described in the pervious section). For the purpose of this work, the keystroke was monitored between releasing one key and pressing the next. Therefore the WM_KEYUP for the key-up event and WM_KEYDOWN for the key-down hook functions residing in a system DLL was used to receive keystroke notifications for all currently running applications. A total of ten users were profiled and the audit trail generated contained the following attributes:

```
<first character, second chararacter, digraph latency>
```

In order to determine accurate digraph latencies, it was also necessary to implement a high-accuracy timer as the default timers available do not offer adequate accuracy for the millisecond latencies expected

4.3.1 Statistical approach

Some pre-processing strategies had to be implemented to eliminate extreme short / long digraph latencies that may adversely affect the distribution of digraph times and, any digraph pair whose latency fell outside a nominal range was excluded from the archived data. For the purposed of this experiment the range was restricted to times above 40ms and below 750ms. These thresholds are based on the original experiments carried out by [1] and are designed to eliminate samples where two keys may have been accidentally struck together (thus, producing an infeasibly small latency) or, where the user may have made a pause in their typing and thus introduced an unnaturally large inter-keystroke latency. Following the pre-processing, and due to the limited set of data, analysis focussed on the 4 main users who provided the largest profiled data sets in order to best illustrate the trends observed. The experimental data for each user was then processed off-line to calculate the mean and standard deviation values for each unique digraph pair.

In the event that any digraph pair had a standard deviation greater than its mean value, the digraph samples were sorted and the top / bottom 10% were then removed with subsequent re-calculation of the mean and standard deviation values – this was only attempted where at least ten samples were available for the digraph pair. The reason for this additional step was to remove digraph samples where the latencies would have an adverse affect on the standard deviation (i.e. the distribution of samples was tightened by removing extreme outliers). Once a set of digraph pairs was produced (with corresponding mean / standard deviation digraph latency values), the user's profile was further constrained by filtering out digraph pairs where the sample count fell below a nominal threshold value. Our experiments fixed this value at fifty samples; however, the software used for analysis allows a variable threshold. A summary of the profiles generated by this method is shown in *Table 3*.

User	Unique Digraph Pairs	Filtered Digraph Pairs	Average Inter-keystroke Time
User A	466	122	151ms
User B	405	51	145ms
User C	412	89	206ms
User D	461	127	162ms

Table 3: Summary of user profile statistics

Once a user profile was generated, the profile was evaluated by comparison with the users' raw keystroke data. This allowed the test profile to be evaluated using the users' own data (to test the False Rejection Rate – FRR) and against other users' keystroke data (to test the False Acceptance Rate – FAR).

As there is likely to be significant variation in a users' own session data, a compensatory factor was applied to the standard deviation that could be varied in a "live" environment according to the security needs of the organisation (see *equation 2*). This factor allowed the number of standard deviations from the mean to be adjusted. For the purposes of this experiment, four weightings were considered namely 0.5, 1, 1.5 and 2. This produced an acceptable digraph range.

digraph range = mean ± (standard deviation * weighting factor) ------------(2)

4.3.2 Statistical approach - results

When viewing the preliminary results as depicted in *Figure 14* if we consider the four users A, B, C and D and follow the vertical columns of data, we can see a clear peak for each users data when compared with their own profile. This is most noticeable for user C where a significant peak is observed (50% of all digraphs accepted) compared with 35% when user B's digraph data was tested against the same profile.

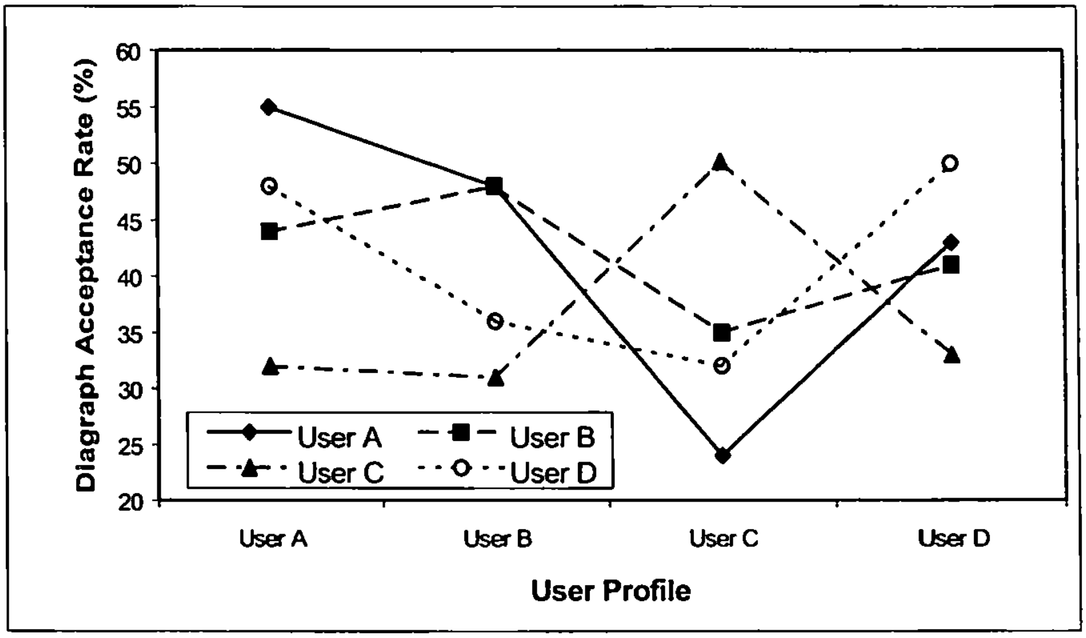


Figure 14: User profile comparisons

Although there was a clear correlation between user C's profile and data, if we consider user A, there was a high FAR for data from users D and B (impostors) when compared with user A's profile. We can also see that in user B's profile the impostor "user A" achieved the same acceptance rate (48%). It is clear from these results that an additional measure of acceptance / rejection is required. To further test the FAR / FRR of the test system, the analysis software monitored the number of consecutively rejected digraph pairs – representing the highest alert level of the system as depicted in *Figure 14*.

When considering (see *Figure 15*) we can identify two distinct trends. Firstly, the solid line plots the digraph acceptance rate for all user data sets against user C's profile. Here we can see a clear peak correlating to user C's own data and corresponding reductions in the acceptance rates for the other users' data. Secondly, the dashed line indicates the highest alert level detected by the analysis software. This is simply a record of the highest count of consecutively rejected digraph times (excluding non-profiled digraph pairs). Again, we can see a correlation between user C's own data when compared with their profile and corresponding increases in the alert level as impostor data sets are compared with the target profile.

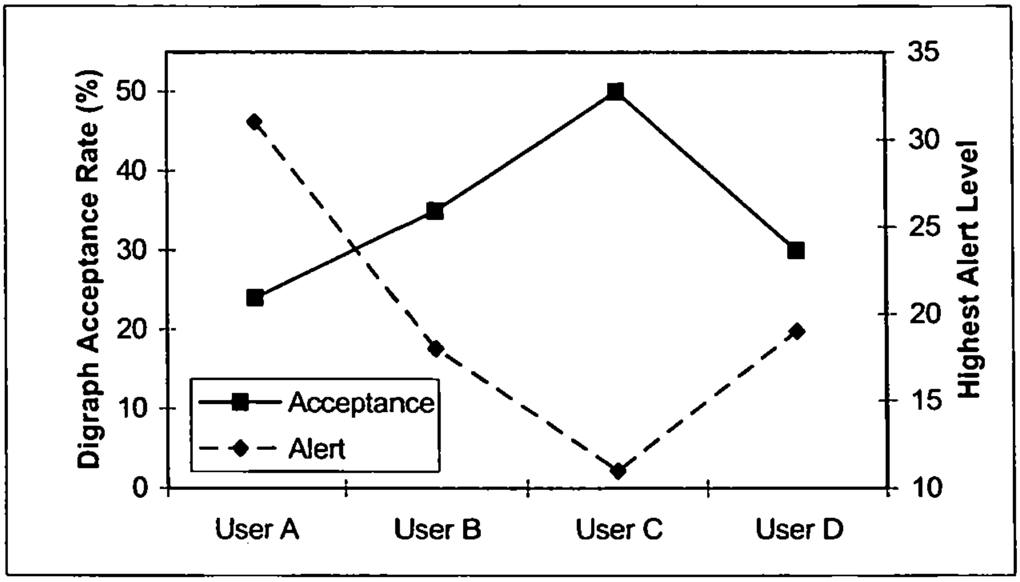


Figure 15: Single user profile comparison

4.3.3 Data Mining approach

The methodology described in the previous sections, using traditional statistical approaches to learn temporal regularities requires a significant level of manual intervention in the data analysis stages. Further, it is time consuming when considering the amount of data generated from a single session or multiple sessions and the number of users on a system. From this we can see the need to automate some of the *audit analysis* stages or pre-processing stages. The methodology used to analyse the raw keystroke data using DM follows the similar principle as described in *section 4.2* which is derived from the four main activities of DM; *selection*, *pre-processing*, *data mining* and *interpretation*. For the purpose of this work, the data sets were split into a ratio of 9:1 hence into two parts; a training set and a testing set (using train and test evaluation method). The raw clean data sets will undergo analysis using IDA Data Mining Tool (same algorithms were used as in the previous work) The algorithm or classifier is subjected initially with the training set and then the classification accuracy is tested using the unseen data set or testing set. The results give an indication of the error rate (or FAR) and the overall classification accuracy of the trained algorithms.

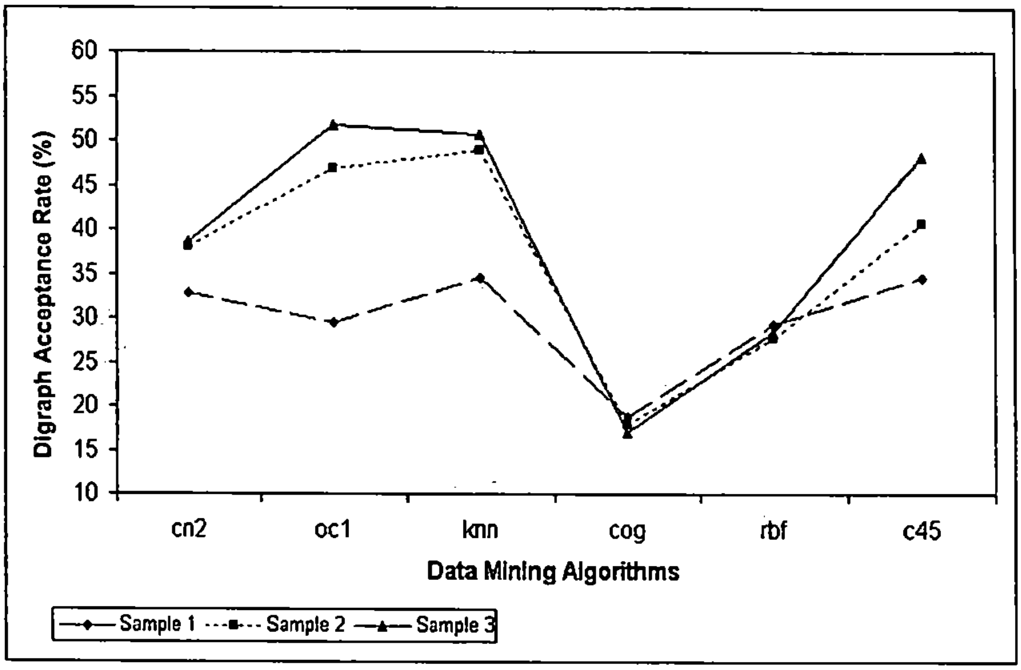


Figure 16: Varying sample sizes with fixed number of classes and attributes

The percentage acceptance rate obtained is encouraging as depicted in *Figure 16*, when considering the acceptance rate achieved for the highest algorithm is 53%. This is in consideration of the time factor involved in comparison to the statistical approach and the amount of domain expertise input to the process, which only resulted in an absolute difference of 2% from the highest percentage acceptance rate (i.e. 55%) obtained in the statistical analysis. Furthermore the acceptance rate obtained increases proportionally (unlike the statistical approach which is restricted in the sample size analysed), except for the COG and RBF algorithms. This is important when considering the size of data being analysed and hence eliminates the ad-hoc approaches adopted using traditional statistical methods.

The initial results suggest that Machine Learning (OC1 and C4.5) and Statistical (k-NN) based algorithms are suitable for these types of data sets. Despite the results, more work needs to be carried out in order to correlate the results to a specific or group of algorithm(s), in order to obtain a higher percentage of classification accuracy. Nevertheless it is clear from the comparative study carried out that DM algorithms have the potential to automate the process of discovering the temporal regularities from the data sets, which would otherwise rely heavily upon intuition and experience in building this model using other approaches (e.g. statistical approach).

4.4 Conclusion

This chapter has presented a series of results from the preliminary statistical analysis of multi-application keystroke data. This has been contrasted with a DM approach to the production of a unique user profile. Whilst the results from this stage of the research are not as encouraging as we had hoped for, they have shown a potential for the use of continuous user authentication. However, it is also clear that a simple statistical approach does not provide sufficient distinction between users. The DM approach is limited due to the nature of the data gathered and will also require further research. It is proposed that further work will investigate the usefulness of trigraph keystroke combinations (timings for three consecutive keystrokes) and the possible use of word-graph timings (timings for frequently occurring words).

The methodology and algorithms utilised in this chapter provide the foundation which could be integrated into an intrusion correlation framework, as discussed in the next chapter.

Chapter 5

A Correlation Framework for Continuous Authentication

5. A Correlation Framework for Continuous Authentication

5.1 Introduction

This chapter begins by describing the conceptual architecture of the Intrusion Monitoring System (IMS) previously described in [1]. It then leads into discussion of how this can be used to support a novel correlation module to enable behavioural profiling for continuous user authentication.

5.2 Intrusion correlation framework

An intrusion correlation can be referred to the interpretation, combination, and analysis of information from all available sources about target system activity for the purpose of intrusion detection and response [16]. While various schemes and detection algorithms employed in IDSs have been described in chapter 2, no work has been reported on integrating the techniques of DM into a correlation framework to enable continuous authentication of users on a system. Therefore this section is aimed on introducing the proposed advanced correlation framework using Data Mining, which would be an integral part of the Intrusion Monitoring System (IMS). A conceptual description of the concept that underlie the IMS architecture is first provided, which will enable the proposed advanced correlation framework in the latter sections to be understood.

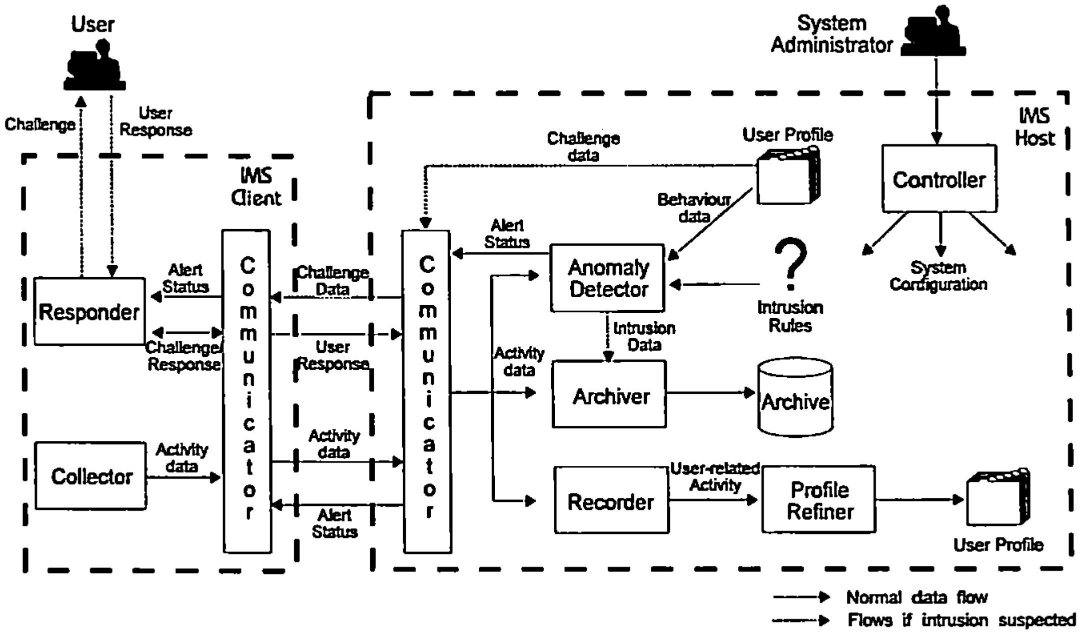


Figure 17: Intrusion Monitoring System (IMS) architecture - conceptual architecture

The IMS as depicted in *Figure 17* is a conceptual architecture based on a client / server relationship for real time intrusion monitoring. The purpose of the Clients is to collect the required data relating to user and process activity and respond to any suspected intrusion detected by the Host. All behaviour profiles, generic rules and such like are maintained securely at the Host, which also handles all of the analysis and the main bulk of other processing associated with the supervision.

At a lower level, the Host and Clients systems will be comprised of a number of modules, each handling a different aspect of the overall intrusion monitoring task (as illustrated in *Figure 15*) and are defined in the sections that follow.

5.2.1 Anomaly Detector

The *Anomaly Detector* analyse user and process activity for signs of suspected intrusion, comparing it against the behaviour profiles that apply to the current user's (claimed) identity as well as against generic intrusion rules. In practise, this module will be comprised of a number of further sub-components, which will be expanded upon in the latter section of the thesis, each handling a specific aspect of anomaly detection and behaviour monitoring (e.g. keystroke analysis, resource usage, etc.). The detector maintains an *alert status table*, with entries existing throughout the life of each user-initialised session of process to indicate the level of detected anomalies and thereby the confidence of potential intrusion. Each entry contains the basic information (similar to the audit trails semantic previously described), which is examined and updated each time activity data relating to the user / process is analysed.

5.2.2 Profile Refiner

The data collected is desirable to be optimised for updating behavioural profiles besides to analyse for detecting deviation of normal use. It is envisaged that the data generated from user-related activity would provide the IMS as the basis for automatic update of user profiles, recognising that behaviour may legitimately alter over time (e.g. improvements of typing ability, imposing different resource demands as a result of new applications accessed, etc.).

5.2.3 Recorder

The *Recorder* handles the short-term storage of user-related activity data during the period of user session and focuses specially upon the collection of data relating to the profiled characteristics of a given user (e.g. collection of resource usage for user and process activity). Upon termination, the information will be used as input to the *Profile Refiner*, provided that the session was not considered anomalous. In the event of a proven anomaly, *Recorder* can discard its stored information for the session.

5.2.4 Archiver

The *Archiver* collects data relating to all system activity and stores it in a long-term archive (in the same manner as a traditional audit trail), providing a more permanent record of activities and suspected anomalies. The storage will occur regardless of whether sessions / processes are regarded as anomalous and details of all security relevant events will be archived. However, in order to conserve storage space, it may be desirable in some scenarios to only record details of certain types of event.

5.2.5 Collector

The *Collector* represents the interface between the IMS and the existing information system / applications, with the responsibility for obtaining information on all relevant user and system activity. The module is envisaged to operate in such a way as to encompass, but be independent of, all system applications. In the context of the experience to date, the application that was written and installed on local workstations fulfilled the role of the collector.

5.2.6 Responder

The *Responder* centres around the continuous monitoring of the alert status transmitted by the Host, with increases in the level triggering certain actions. The module would reside in the IMS client and handles the task of responding to the anomalies detected by the Host.

5.2.7 Communicator

The *Communicator* provides the network communications interface between the Host and Client(s) systems operating on the local network. As such, the functionality of this module is duplicated on both side of the link. The principal functions would include transmitting user and process information to the Host and then subsequently keeping the Client(s) informed of the current alert status.

5.2.8 Controller

The *Controller* module enables the System Administrator to configure the operation of the IMS system on both the client and host side. In addition, other features (such as profile management) would be provided under the auspices of the Controller module.

5.3 Advanced Intrusion Correlation (AIC) framework

The approaches investigated for user behavioural profiling from the collected user and process activity audit trail could be used as the basis to provide identification and authentication of users and monitoring for signs of suspected intrusion. Our initial results show the potential of developing and integrating the DM techniques investigated into the Anomaly Detector and Profile Refiner modules of the IMS. Previous work by [56] has been focused on adapting the concepts of continuous monitoring (using the principles of IMS) to enhance the OS user identification and authentication schemes, while the proposed AIC framework, as depicted in *Figure 18*, is to address the issues of *audit analysis* which has been described previously. It is envisaged that the AIC will act as an agent within the *Anomaly Detector* of the IMS.

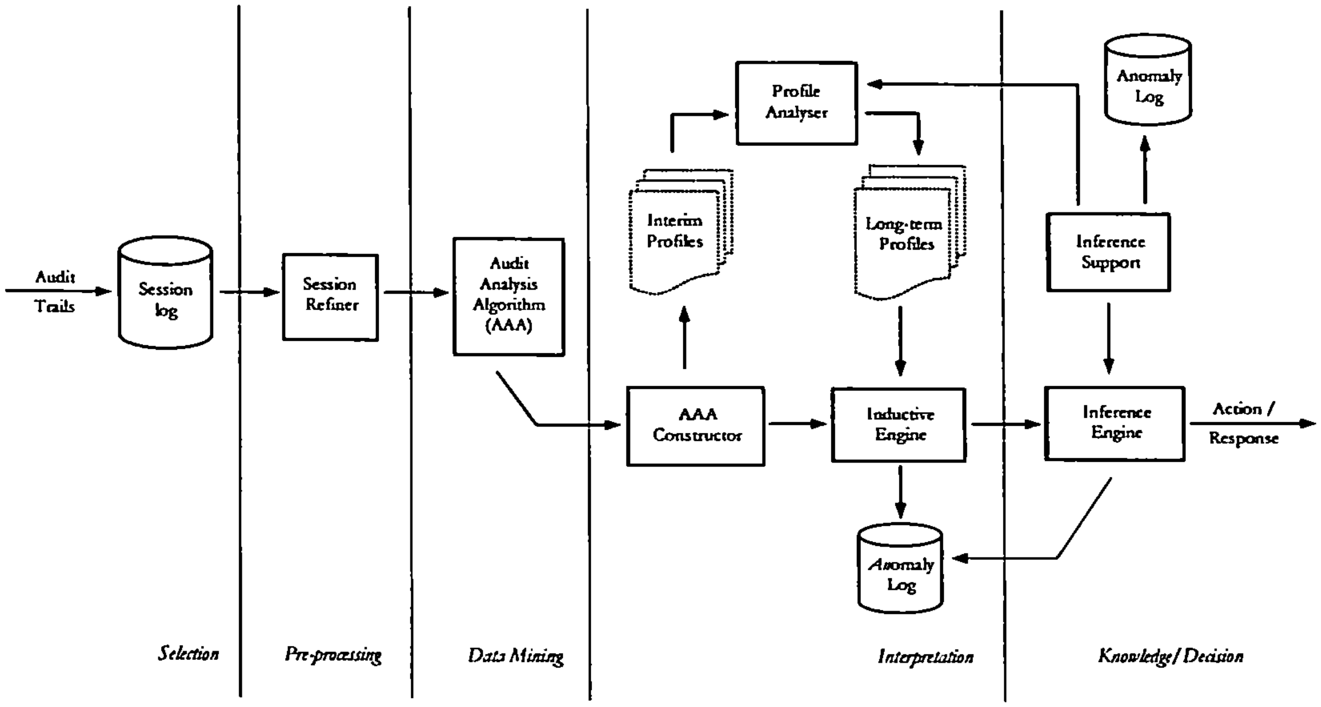


Figure 18: Proposed Advance Intrusion Correlation (AIC) framework for user authentication and monitoring

The concepts behind the AIC framework are in some ways similar to the principles of inductive reasoning [57]. Where the goal is to arrive at a decision (i.e. if deviation is occurring) from a limited set of information (i.e. behaviour indicative data) available due to the inherent problem of gleaning specific information from audit trails. The key aspects of this design are defined in the sections that follow.

5.3.1 Session Log

The *Session Log* would provide a temporary storage to the generated audit trails. The data stored would be restricted to only relevant data pertaining to the behaviour indicative data (e.g. resource usage data, keystroke data, etc.). This is in contrast to the functionality of the IMS *Archiver* module, which provides a permanent storage to all system activity, although it can potentially be a source for a permanent record to provide evidential support should the need transpires. This would enable a reduced access time taken for the *Audit Processing* module to select the relevant features for analysis in the later stage.

5.3.2 Session Refiner

The *Session Refiner* involves preparing the target data set prior to undergo analysis. Typically it may involve converting the data into acceptable format, demarcation of the beginning of each message (e.g. time stamp, absolute time, etc.) or may involve processes to quantify continuous features, for instance as used to generate the audit trail semantic. These stages can be used to constraint the search space and make patterns or relationships in the audit trails more visible in the later stages of the AIC processes.

5.3.3 Audit Analysis Algorithms

The *Audit Analysis Algorithms (AAA)* incorporating the DM algorithm(s) will be used to identify system features, patterns and latent trends for classifying user behaviour. Data features such as correlation between adjacent or frequent sequential patterns of user behaviour occurring will be analysed. The information gleaned from using these algorithms will be used to identify the temporal regularities of user's behaviour, which will be reflected in the user's profiles in the latter stages.

5.3.4 AAA Constructor

The *AAA Constructor* would refine the inferred association rules or classification rules from the *AAA* engine. The various types of patterns exhibited in the data would be cleaned (i.e. remove redundant data), combined, and transformed into an understandable syntax. This will later be stored in the *Interim Profile*, which would provide a temporary repository.

5.3.5 Profiler Analyser

The concepts behind the *Profiler Analyer* are adapted from the *IMS Profiler Refiner* and would have a similar functionality in the proposed AIC framework.

Similarly the audit trails generated will be optimised as input to the *Inductive Engine* to detect deviation from normal use and as a source for updating user profiles which inherently will change over time. Depending on how often similar patterns are exhibited by user(s), these changes will be reflected in the *Long-term Profiles* repository.

5.3.6 Inductive Engine

The *Inductive Engine* would enable the detection of any deviation occurring on the system. The *Long-term Profiles* of users will be compared against the generated audit trails to detect for anomalies. Anomalies detected would be stored in the *Anomaly Log,* which will provide the basis for detected deviations to be further analysed in order to reduce the probability of false positives error prior to reporting the conclusion inferred through the *Inductive Engine.*

5.3.7 Inference Engine

The *Inference Engine* would be used to identify reoccurring valid behavioural patterns that are being flagged as anomalies. These will be filtered out in order to reduce the potential of high false positive errors by correlating previously known anomalies logged in the *Anomaly Log* to the current active anomaly detected and from known information input through the *Inference Support* component.

5.3.8 Inference Support

The *Inference Support* would be used to improve the inferred facts from other sources. System Administrators or Security Officers can input this information (e.g. public holidays, staff on sick leave, etc), which would otherwise take a longer time, through normal circumstances, to infer and thus detect anomalies occurring. Furthermore it will be used to provide input to the *Profile Analyser* where deemed necessary for instance, to disable profiling when user is away as countermeasure against the possibility of unauthorised user introducing new temporal behaviour which would effect the legitimate user's profile. It would also enable any modifications (i.e. removing or adding anomalies logged) or up keeping required in the log files of the *Anomaly Log*.

5.4 Conclusion

The notion of a real-time intrusion detection architecture, such as the IMS, with automated processes to protect the systems from unauthorised users is interesting. At the same time the underlining technology to achieve this capability presents a challenging area in IT security. While there is a tendency to equate complex statistical analysis with correlation or the detection mechanism behind this architectures, the methodology developed in analysing the audit trails, which is advocated through the proposed AIC framework, has the potential to provide an important contribution to development of the intrusion detection systems besides the development of the IMS.

Chapter 6

Conclusions

6. Conclusions

6.1 Introduction

The research has concentrated on identifying techniques in order to provide continuous monitoring for user identification and authentication. This work is to support the development of the correlation module previously described in chapter 5 as part of the conceptual architecture of the IMS framework.

Current approaches in this area have, as described in chapter 2, been focused on improving the initial login stages. Although this acts as a deterrent to most "socially engineered" penetrations on the system, abuses resulting from internal security threats and repeated abuses occurring from successfully breached systems cannot be detected using these forms of user authentication methods. This problem could be associated to the inherent characteristic of a one-off authentication scheme. The problem could be addressed through appropriate monitoring and analysis of user activity within an active session. Intrusion detection systems employing these schemes, such as the proposed IMS, have the potential to provide an important contribution in order to protect the systems against the form of security threats, described previously, by providing the system with continuous user authentication and monitoring capabilities. However, the difficulty in adopting these approaches, as described in the earlier sections of the thesis, arises from the *audit analysis* stages. This is due the techniques currently employed in analysing the vast amount of audit trails generated in order to classify the information collected as "intrusive action" or "normal" against some form of predefined conditions. The techniques adopted to address these problems rely heavily on intuition and experience, coupled with ad hoc and manual methods to build the correlation modules, which are generally represented through complex statistical-based detection algorithms.

There is consequently, a clear need for methods to automate some of this process by incorporating pro-active approaches as used in intelligent data analysis techniques. This has led to the investigation of data mining techniques as a possible approach, as an intelligent data analysis method for integration into an advanced intrusion correlation framework proposed in chapter 5.

6.2 Achievements of the research programme

The work was carried out in stages, where the early part of the work was focused towards identifying key indicators of user behaviour in order to provide the input source for the correlation purpose. The latter part of the work was divided into two main areas. The first was focussed on studying the DM algorithm classifiers in automating the processes involved in the *audit analysis* stages. It involved studying the classifier abilities in discovering the latent trends from the generated audit trails. This was in order to build the user profiles for the authentication process. The second area was focussed towards studying the various data features (e.g. association, sequential, etc.) that could be extracted from the audit trails. This was employed to provide a stronger correlation between user's behavioural profiles and the user's interaction with the system.

6.2.1 Time ordered sequence

The work identified attributes (e.g. frequent pattern usage of applications run by a user) that could be correlated over a period of time (as depicted in *Figure 6*) to identify a legitimate user. Auditing the applications that users run could provide a distinctive pattern of the user's interaction with the system. This pattern, once identified, could be incorporated into an anomaly detector framework in conjunction with other key indicators of user behaviour. Therefore the legitimate user patterns when compared against other usage patterns of users interacting with the system could potentially be used to identify unauthorised access or intrusion.

The audit trail data observed resembled a time ordered sequence such that $X < Y < Z$ where, if X, Y and Z are events of certain attribute, Z must occur first followed by Y and then X. These attributes represent a novelty in this area because some intrusions can only be detected by analysing sequential interrelation between audit trails, since each event alone can appear to be normal according to the standard statistical measures, upon which, previous work has been based.

6.2.2 Frequent sequential patterns

The investigative work carried out was further extended to study attributes that could be used to correlate systems usage to the respective user(s) as considered in chapter 4. This is primarily based on the assumption that users performing their regular tasks will impose similarly regular demands upon system resources. Therefore using these attributes the legitimate user patterns could be modelled and compared against other usage patterns of users interacting with the system, which could potentially be used to identify unauthorised access or intrusion. While most statistical anomaly detection algorithms use similar system features, the detection algorithms are based upon statistical measures, which involve complex statistical measures obtained through manual and ad-hoc approaches. Therefore the work proceeded to study the DM methodology, which is an intelligent data analysis technique, a novel approach yet to be explored in this domain for an advanced user authentication scheme. This was in order to correlate between adjacent or frequent sequential patterns of user behaviour occurring that is seemingly intractable using the statistical approaches described in the earlier chapters.

The DM techniques were also studied in terms of the feasibility and effectiveness of the learning algorithms in automating the discovery process of the temporal regularities exhibited by users (a task normally carried out by domain experts by monitoring the audit trails). As such the comparative studies carried out was aimed at studying the heuristics of the DM algorithms, where part of the work has been published [50] and represents novel work in this area.

6.2.3 Learning temporal regularities of user behaviour

The methodology (i.e. the AIC framework) proposed in chapter 5 to automate the process of learning temporal regularities exhibited by users behaviour was further substantiated through a comparative study carried out against statistical approaches, where part of the work has been published [51]. The keystroke latencies for a user in an active session were collected and the analysis for both the methods was carried out off line (as described in chapter 4). The results obtained showed the potential of using DM algorithms to automate the process of discovering the temporal regularities in comparison to the manual and ad hoc means using statistical approaches. Furthermore the time taken to build the temporal regularities from analysing the keystroke latencies is much quicker than statistical approach, which is important when considering the size of the samples analysed.

6.3 Opportunities for future research

Whereas pervious work in this area has been focused on developing the DM algorithm for domain specific problems, no work to date has integrated these techniques into a correlation framework. Therefore the AIC framework represents a novel approach as an underpinning technology to enable the automation of the processes involved in the audit analysis stages. Hence, future research work could build upon the key areas researched in this project. This is in consideration of the encouraging results obtained from the studies carried out using the DM techniques and methodologies for the audit analysis stages. The results showed the applicability of using these techniques in order to automate the process involved in monitoring user's activities for detecting anomalies occurring on the system. However, the studies also suggest that the accuracy of the correlation process is dependent on the input data source. Therefore the studies identified the need to build different data features from the same data samples to enable a better correlation of user's behaviour from the generated audit trails. The work also identified the need to possibly use behavioural biometrics (i.e. keystroke patterns) in combination with other key indicators of user behaviour (e.g. resource usage, application a user runs, etc.) as the input data source for the correlation process in order to improve the classification accuracy (i.e. as "intrusive" or "normal").

6.3.1 Development of prototype AIC framework

Until now the work has been based on off-line analysis, where the data was collected from participating computers in a networked environment and the analysis was carried out using the IDA tool. Therefore the work will focus on the design and development of the *Session log*, *Session Refiner*, *Audit Analysis Algorithms (AAA)*, and the *AAA Constructor* components of the proposed AIC framework. The developed components, when integrated, will enable the automation of the *selection*, *pre-processing* and *data mining* stages of the correlation processes involved. As well as automating the processes involved it will also act as a platform to enable the on-line analysis to be carried out in order to test the selected authentication schemes classification accuracy. Experiences and findings from the development and implementation of these main components will inform the specification for the correlation framework and to provide a full specification for the IMS architecture anomaly detector module.

6.3.2 Session Log

The generated audit trails from the monitoring process of user activity within an active session on Window's NT (i.e. as the target OS), will be stored as historical records in the *Session Log*. These logs will be stored on the host, which would provide an input source to other process in later stages of the analysis, as well as to reduce the processing load on the client computers. This represents new work in the context of using Windows NT as the OS to study the implications of a continuous monitoring / supervision schemes. It is envisaged that an appropriate DBMS will used to enable the processing time taken for the pre-processing stages to be reduced as well as to allow the match and retrieval capabilities of database technologies to be studied for real-time analysis. It is considered that the experience and findings could provide an important contribution for on-line analysis applications, such as the IMS, when trying to provide an efficient storage and lookup of large data sets.

6.3.2.1 Session Refiner

The DM data analysis techniques studied from the *pre-processing* stages will be integrated into this component in order to constrain the search space. This will make patterns or relationships in the audit trails more visible in the later stages of the AIC processes, which would otherwise be conceived through re-iteration of the analysis stages. The techniques used will provide an important contribution to the pre-processing stages, since previous approaches adopted have been based on filtering the data collected from the monitoring / supervision process itself. This approach is aimed at analysing the data more efficiently rather than focusing on reducing the amount of data analysed. However, it is envisaged that the strategies adopted in this process will be dependent on the data features analysed (e.g. association, sequential, etc.).

6.3.2.2 Audit Analysis Algorithms

Further analysis will be carried out to identify suitable DM algorithms that can be used to identify system features, patterns and latent trends for classifying user behaviour more accurately. The work will also consider parallel implementation of the classification algorithms (e.g. C4.5, k-NN, etc.) with association algorithms. This work is considered novel since previous work has focused on using anomaly detector's based on classification of an event rather then looking at correlation between adjacent patterns as previously described in chapter 2.

6.3.2.3 AAA Constructor

In order to build more accurate user behaviour profiles, the various types of patterns exhibited, inferred from the *AAA* engine, will have to be cleaned (i.e. remove redundant data), combined, and transformed into an understandable syntax. Therefore studies on short term and long term profiles of user behaviour will be carried out to enable the task of selecting appropriate data features from the *data mining* process in building data features for latter stages of the AIC framework.

6.3.3 Evaluation

In order to determine the classification accuracy of the prototype AIC framework in detecting deviation from normal use, qualitative testing of the framework will be tested in a live session. This will also allow further false positive / false negative errors occurring to be calculated and subsequently compared with both the individual measures and the overall performance of the prototype system. It is considered that this form of testing would not only provide reliability statistics, but also a subjective evaluation by users.

6.3.4 Behavioural indicative data

Categorising user behaviour patterns and classifying it as normal or intrusive is a subtle task. Furthermore exploring the vast amount of audit trail data can often only yield a small fraction of intrusion or misuse. Besides managing these tasks, the proposed AIC framework will have to limit the errors that could occur from misclassification of user behaviour such as false positive errors (which classifies legitimate behaviour or actions as anomalous), and false negatives (where an intrusive action is not detected and proceeds through the system). Therefore it is essential to gather as much information as possible pertinent to a user's interaction with the system in order to distinguish between similar behavioural patterns of users that could occur. The work will further the studies on a combination of key indicators of user behaviour to determine the classification accuracy improvement from previous work. As such combination of behavioural biometrics such as keystroke latencies and system resource usage will be considered for the purpose of continuous authentication besides using other key behaviour indicators. This will represent a novel contribution in this area since previous work only considered system feature's attributes for the input to the correlation engine.

6.3.4.1 Building data features

The work could build upon the work carried out in this project to study the correlation between user's behaviour profiles and interaction with the system. Data features that could be used to model these behaviours based on the DM techniques (e.g. associations, sequential, etc.) will be further studied.

6.3.5 Profile refinement

A full specification for the profile refinement components, specifically the *profile analyser* component of the AIC framework will be provided from the basis of the strategies used in the development of the prototype AIC framework. The techniques for user profile refinement will be investigated to provide automated updating depending on how often similar patterns are exhibited by user(s). It is envisaged that some form of data mining inductive learning algorithm will be required to enable these processes [42].

While the work is focused on behavioural profiling and advanced intrusion correlation framework using Data Mining techniques, it should be noted that this research would be conducted in conjunction with other IMS-related work within the Network Research Group. Other researchers will be considering the issues of misuse detection, incident response and user authentication.

6.4 Conclusion

While there is a tendency to equate complex statistical analysis with correlation or the detection mechanism, this thesis has presented the results to date from the comparative studies carried out using DM. The methodology used and the classification accuracy obtained in this initial investigative work suggests that DM techniques could be integrated into a correlation framework for continuous authentication. The high classification accuracy obtained and fast response time exhibited in classifying the user behaviour by some of the DM algorithms, when considering the vast amount of audit trail analysed, further demonstrates the potential of applying DM techniques within a real-time application. Whereas previous work in this area has been focussed on developing the DM algorithms for domain specific problems, no work to date has integrated these techniques into a correlation framework. The methodology developed in analysing the generated audit trails, which is advocated by the proposed correlation framework, has the potential to provide an important contribution to the development of a correlation framework for the purpose of continuous user authentication.

References

References

1. S.M. Furnell, "Data security in European healthcare information systems" PhD Thesis, University of Plymouth, U, 1995
2. Computer Security Institute, "2000 CSI/FBI Computer Crime and Security Survey", Vol. 6, No.1, SPRING-2000.
3. B. Betts, "Digital Forensic:crime scene", March 2000, http://www.infosecuritymag.com/articles/march00/cover.shtml
4. M. Ward, "Web warning centre in net attack", BBC news, 24th May 2001, http://news.bbc.co.uk/hi/english/sci/tech/newsid_1348000/1348820.stm
5. G. Bradberry and Nick Nuttall, "Hackers Steal Microsoft Secret Codes", The Times, Saturday October 28 2000, 1-2.
6. J.P. Anderson, "Computer Security Threat Monitoring and Surveilance", James P. Anderson Co., Fort Washington, PA(Apr.)-1980.
7. P. Oldfield, "Computer Viruses Demystified", ISBN 0-9538336-0-7, Sophos Plc-2001, 16-18.
8. T. Escamilla, "Intrusion Detection: network security beyond the firewall", ISBN 0-471-29000-9, Wiley-1998, 3-27.

9. B. Heskett, "A New Windows password Cracker", Cnet News.com, 13th February, 1998, http://news.cnet.com/news/0-1003-200-326537.html

10. D. Klein, "A Survey of, and Improvement to, Password Secutity", Proc. of the USENIX Second Security Workshop, Portland, Oregon, 5-14.

11. E.H. Spafford, "Opus: Preventing Weak Password Choices", Secure Computing, UK, January 1990.

12. J. Ashbourn, "Biometrics: advanced identity verification", ISBN 1-85233-243-3, Springer-Verlag-2000, 45-64.

13. D.E. Denning, "Information Warfare and Security", ISBN 0-201-43303-6, Addison Wesley-1999, 297-319.

14. L. Liebmann, "Security Myths and Reality", NetworkWorld – white papers central, http://www.nwfusion.com/whitepapers/security/.

15. R. Heady, G. Luger, A. Maccabe and M. Serville, "The Architecture of a Network Level Intrusion Detection System", Technical Report, Computer Science Department, University of New Mexico, August 1990.

16. E. Amoroso, "Intrusion Detection: an introduction to internet surveillance, correlation, trace back, traps, and response", ISBN 0-9666700-7-8, Intrusion.Net Books-1999, 15-18.

17. D.E. Denning, "An Intrusion-detection Model", IEEE Transaction on Software Engg., vol. SE-13, Feb 1987, 222-232.

18. T.F. Lunt, "IDES: an intelligent system for detecting intruders", Proc. of the Computer Security, Threat and Countermeasures Symposium, November 1990 Rome, Italy.

19. W.T. Tener, "Discovery: an expert system in the commercial data security environment", Proc. Fourth IFIP TC11 International Conf. On Computer Security, North-Holland, Dec 1986.

20. A. Baur and W. Weiss, "Audit Trail Analysis Tool for System With High Demands Regarding Security and Access Control", Technical ZFE F2 SOF 42, Seimens Nixdorf Software, 1988.

21. T.D. Garvey and T.F. Lunt, "Modal Based Intrusion Detection", In Proc. of the 14th National Computer Security Conference, 1991, 372-385.

22. P. Porras and R. Kemmerer, "Penetration State Transition Analysis: a rule-based intrusion detection approach", 8th Annual Computer Security Application Conf., 1992, 220-229

23. K. Ilgun, R. A. Kemmerer, P.A. Porras, "State Transition Analysis: a rule-based intrusion detection approach", IEEE Transactions on Software Engg. vol. 10, no.Y, 1995, 181-199.

24. S. Staniford-Chen et al, "GrIDS-A Graphical Based Intrusion Detection System for Large Networks" Proc. of the 19th National Information Systems Security Conf., 1996, 361-370.

25. H.S. Vaccaro and G.E. Liepins, "Detection of Anomalous Computer Session Activity", Proc. Symposium on Research in Security and Privacy, May 1989, 280-289.

26. J. Hochberg et al., "NADIR: an automated system for detecting network intrusion and misuse", Computers and Security, vol. 12, no. 3, 1993, 235-248.

27. P.A. Poros and P.G. Neumann, "EMERALD: event monitoring enabling responses to anomalous live disturbance", Proc. of the National Information System Security Conf., 1997.

28. R. Jagannathan et al., "System Design Document Next-Generation Intrusion Detection Expert System (NIDES)", Technical Report, SRI International, SRI Project 3131, 1993.

29. W. Lee and S. Stolfo, "Data Mining Approaches for Intrusion detection", Proc. 7th USENIX Security Symposium, 1998.

30. E. Jonsson and T. Olovsson, "A Quantitative Model of the Security Intrusion Process Based on Attacker Behaviour", IEEE Transactions on Software Engg. vol. 23, no. 4, 1997.

31. J. Frank, "Artificial Intelligence and Intrusion Detection: current and future direction", Proc. of the 17th National Computer Security Conf., October 1994.

32. Symantec Corporation, "Intruder Alert", http://enterprisesecurity.symantec.com/products/products.cfm?ProductID=48

33. Intrusion.com, "The Kane Security Monitor Real Time Security Auditing for Microsoft NT Server & Workstatione", http://www.cstl.com/html/info/idi/ksm.htm.

34. W.J. Trybula, "Data Mining and Knowledge Discovery" Annual Review of Information Science and Technology (ARIST)", vol. 32, 1997, 197 – 225.

35. C.E. Broadley, T. Lane and T.M. Stough, "Knowledge Discovery and Data Mining" American Scientist, vol. 87, 54-61.

36. U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy, "Advances In Knowledge Discovery and Data Mining" AAAI / The MIT Press, -1996, ISBN 0-262-56097-6, 1-31.

37. U.M. Fayyad, "Data Mining and Knowledge Discovery - Making Sense Out of Data", IEEE Expert-1996, vol. 11, no. 6, 20-25.

38. J. Angstenberger, K. Lieven and R. Weber, "Intelligent Methods for data Mining", Proc. UNICOM Seminar on Data Mining, 37-67.

39. P. Adriaans and D. Zantinge, "Data Mining", Addison Wesley-1996, ISBN 0-201-40380-3.

40. P.R. Limb and G.J. Meggs, "Data Mining: tools and techniques", BT Technology Journal, vol.12, no. 4 October 1994, 32-41.

41. S.S. ANAND, D.A BELL and J.G. HUGHES, "The Role of Domain Knowledge in Data Mining", 4th International ACM Conference on Information and Knowledge Management, Baltimore, USA, 1995, 37-43.

42. D. Michie, D.J. Spiegelhalter and C.C. Taylor, "Machine learning, Neural and Statistical Classification", Ellis Horwood-1994, ISBN 0-13-106360-X.

43. H.S. Teng, K. Chen and S.C. Lu, "Adaptive Real-time Anomaly Detection Using Inductively Generated Sequential Patterns", Proc. of the IEEE Symposium on Research and Privacy, May 1990, 278-284.

44. R.S. Michalski, I. Bratko and M. Kubat, "Machine Leamining and Data Mining: methods and applications", Wiley-1998, ISBN 0 471-97199 5.

45. J.R. Quinlan, "C4.5 Programs for Machine Learning", Morgan Kaufmann Publishers-1993, ISBN 1-55860-238-0.

46. S. Murthy, S. Kasif, S. Salzberg, and R. Beige, "OC1: randomised induction of oblique decision trees", Proc. of the 11th Nat. Conf. on AI AAAI-93, Washington D.C., 1993, 322-327.

47. E. Simoudis, "Reality Check for Data Mining", IEEE Expert - October1996, 26-33.

48. P. Clark and R. Boswell, "Rule Induction with CN2: some recent improvements", Proc. of the 5th European Conference (EWSK-91), 151-163.

49. C. Warrender, S. Forrest and B. Pearlmutter, "Detecting Intrusion Using Calls: altemative data models", Symposium on Security and Privacy, 1999.

50. H. Singh, S. Fumell, B. Lines and P. Dowland, "Investigating and Evaluating Behavioural Profiling and Intrusion Detection Using Data Mining", Proc. of the MMM & ACNS, St. Petersburg, Russia, 2001, 153-158.

51. P.S. Dowland, H. Singh and S.M. Fumell, "A Preliminary Investigation of User Authentication Using Continuous Keystroke Analysis", To be published as part of the Proc of the IFIP 8th Annual Working Conference

on Information Security Management & Small Systems Security, Las Vegas, 2001.

52. H. Singh, K.E. Burn-Thornton and P.D. Bull, "Classification of Network State Using Data Mining", Proc. of 4th IEEE International MICC & ISCE conf. 1999, vol.1, 183-187.

53. S. Furnell, P. Morrissey, P. Sanders and P. Stockel, "Information Systems Security: facing the information society of the 21st century", ISBN 0-412-78120-4, Chapman & Hall-1996, 283-293.

54. S. Bleha, C. Silvinsky, B. Hussein, "Computer-Access Security Systems Using Keystroke Dynamics", Transactions on pattern analysis and machine intelligence, vol. 12, no. 12, 1990.

55. R. Joyce, G. Gupta, "Identity Authentication Based on Keystroke Latencies", Communications of ACM, vol.33, February 1990.

56. P.S. Dowland and S.M. Furnell, "Enhancing Operating System Authentication Techniques", Proc. of the Second International Network Conference 2000 (INC 2000), UK, 253-261.

57. J. Durkin, "Expert Systems Design and Development", Prentice Hall-1994, ISBN 0-02-330970-9, 90-130.

APPENDIX A

LIST OF ABBREVIATIONS

AAA	Audit Analysis Algorithms
AIC	Advanced Intrusion Correlation
API	Application Programming Interface
DLL	Dynamic Link Library
DM	Data Mining
FAR	False Acceptance Rate
FRR	False Rejection Rate
IDA	Intelligent Data Analysis
IDS	Intrusion Detection System
IMS	Intrusion Monitoring System
KDD	Knowledge Discovery in Databases
LAN	Local Area Network
MLP	Multi Layer Perceptron
OLAP	On-Line Analytical Processing
OS	Operating System
RBF	Radial Base Function

APPENDIX B

EXPERIMENTAL TRIALS

The pages that follow present notes made during experimental trial work, to support the behaviour profiling studies described in Chapter 4.

Trial 1: Behavioural Profiling

These initial trials were aimed at exploring the available tools at hand and also to look at possible long-term shortcomings in terms of resources available in this area. The initial test was targeted towards host-based Intrusion Detection Systems (IDS). The NT operating system (OS) was selected for these initial trials since it is a common operating system used in the NRG. Audit trails were generated using the NT Performance Monitoring (PerfMon) tool, which was configured to monitor selected performance level of system elements.

The research is focused upon continuous monitoring of user(s) upon penetrating the first line of defence such as user login authentication and access control schemes. The nature of continuous monitoring involves analysis of vast amount of audit trails hence the use of Data Mining (DM) techniques, which will be expanded upon in this report. In order to create profiles of users, the PerfMon tool was optimised to generate alerts or audit trails when a threshold is exceeded. Hence the first part of the trials was to select appropriate attributes or objects in order to classify each system resource and to determine normal usage pattern of system resources. The initial thresholds were based on Microsoft's Windows NT Serve 4.0 Enterprise Technologies Delivery Guide. As a result, the attributes as depicted in ***Figure 1*** were selected and optimised upon succession of trial and error using the recommended thresholds for various counters and objects.

Objects	Counters	Threshold (Alert if)
Memory	% Commited Bytes in Use	>20%
	Available Bytes	<12MB
	Page/sec	>25
Processor	% Processor Time	> 70%
	% Privileged Time	> 70%
	% User Time	> 70%
Logical Disk	% Disk Time	>40%
Physical Disk	% Disk Time	>40%
Process*	% Processor Time	>50%
	% User Time	>50%
System	% Total Processor Time	>60%
	% Total User Time	>65%
	% Total Privileged Time	>70%
Object	Processes	>50

*The Process instance selected for this object is optimised to monitor web browsers.

Figure 1: Attributes and threshold used for the initial trials

Some of the audited objects in these trials have been used in the Next-Generation Intrusion Detection Expert System (NIDES) which stores audit trails containing activity for each user and measures such as CPU usage. Both the Adaptive Intrusion Detection System's (AID) agents on the monitored hosts and Distributed Intrusion Detection System's (DIDS) host event generator (HEG), incorporates measures of execution of a process (start and end of session, read and write to file of device and etc) produced by the OS audit records for behavioural profiling. The generated audit trails for the purpose of this work incorporates profiling functionality and captures deviation from normal behaviour of using system resources. The former is based on anomaly detection model. This approach lessens the subtle task of gathering the *priori* knowledge about the sort of behaviour that would compromise system security.

Exceeded Threshold Values	Objects	Counters	User

Table 1: Structure of table entry

The collected audit trails are pre-processed to form a target data to undergo analysis using the Data Mining algorithms. This stage of per-processing was used to select audit trails containing reliable information or clean data. Two profiles were created, one representing a known user and the other an unknown user **(refer to appendix)**. The data entries are structured into a table entry as depicted in *Table 1*. In this stage also the data is first converted into acceptable format to the DM software. The data imported into excel spreadsheet from the PerfMon tool is saved as a text tab delimited file (*.txt).

The Intelligent Data Analysis (IDA) DM tool used for the data mining process incorporates algorithms from the fields of Statistical, Machine Learning and Neural Network. Due to the modified shell scripts used in this latest IDA version, some of the functions previously exhibited (when using older Linux version at SoC) could not be enabled. Hence for this work the CN2 rule based Machine Learning algorithms was selected. The data sets were split into ratios of 9:1, hence into two parts a training set and a testing set. This evaluation method is a common technique used and is refereed to as train and test. The evaluation method used depends on the selected data or sample size.

The algorithms or classifier is subjected initially with the training set and then the classification accuracy is tested using the unseen data set or testing set. The results give an indication of the error rate and the overall classification accuracy of the trained algorithm. The algorithm selected for these trials uses similar optimised criteria as described in the published paper, hence this in not included in the report. The overall accuracy rate obtained for this initial test was in the range of between 82% - 88%. The classification accuracy is encouraging in this case when considering normal usage pattern examples were not included in the data sets.

It is envisaged that some time will be spent on the IDA tool prior to running some more trials. This is in order to troubleshoot the IDA tool shell scripts hence, to replicate the ability to generate rules when using other Machine Learning algorithms. Once this problem is overcome the next stage would be to reiterate the test and compare the classification accuracy using algorithms from different fields and also to compare the results when more users profiles are generated instead of the 2 that was used for this initial trials. The next stage of work will also include re-examining the structure of the table entry to further improve classification accuracy.

```
Examples: profile1.001.data
Algorithm: UNORDERED
Error_Estimate: LAPLACIAN
Threshold: 0.00
Star: 5

*UNORDERED-RULE-LIST*

IF    88.24 < attr_1 < 100.73
THEN  class = c1  [125 0]

IF    attr_1 > 108.20
  AND attr_2 = "Pages_sec"
THEN  class = c1  [17 0]

IF    attr_1 > 44.03
  AND attr_3 = "PhysicalDisk"
THEN  class = c1  [24 0]

IF    28.26 < attr_1 < 43.23
THEN  class = c1  [24 1]

IF    63.25 < attr_1 < 71.98
THEN  class = c1  [15 0]

IF    48.83 < attr_1 < 62.55
  AND attr_2 = "Pages_sec"
THEN  class = c1  [12 0]

IF    44.03 < attr_1 < 55.17
  AND attr_2 = "Disk_Time"
THEN  class = c1  [12 0]

IF    45.63 < attr_1 < 48.13
THEN  class = c1  [7 0]

IF    25.56 < attr_1 < 26.46
THEN  class = c1  [6 0]

IF    6914048.00 < attr_1 <
8890368.00
THEN  class = c1  [34 4]

IF    attr_1 > 9981952.00
THEN  class = c1  [4 0]

IF    4571136.00 < attr_1 <
5683200.00
THEN  class = c1  [7 0]

IF    26.66 < attr_1 < 27.86
THEN  class = c1  [4 0]

IF    attr_1 > 74.24
  AND attr_2 = "Disk_Time"
THEN  class = c1  [19 0]

IF    6866944.00 < attr_1 <
7401472.00
THEN  class = c1  [11 1]

IF    9408512.00 < attr_1 <
9629696.00
THEN  class = c1  [6 0]

IF    5705728.00 < attr_1 <
6064128.00
THEN  class = c1  [3 0]
```

```
IF    6785024.00 < attr_1 <
6842368.00
THEN  class = c1  [2 0]

IF    56.78 < attr_1 < 62.22
THEN  class = c1  [9 0]

IF    85.58 < attr_1 < 87.19
THEN  class = c1  [2 0]

IF    45.04 < attr_1 < 45.43
THEN  class = c1  [4 0]

IF    6623232.00 < attr_1 <
6707200.00
THEN  class = c1  [5 0]

IF    9730048.00 < attr_1 <
9779200.00
THEN  class = c1  [2 0]

IF    6584320.00 < attr_1 <
6596608.00
THEN  class = c1  [2 0]

IF    9932800.00 < attr_1 <
9971712.00
THEN  class = c1  [2 0]

IF    9693184.00 < attr_1 <
9699328.00
THEN  class = c1  [1 0]

IF    9803776.00 < attr_1 <
9844736.00
THEN  class = c1  [1 0]

IF    4327424.00 < attr_1 <
4423680.00
THEN  class = c1  [1 0]

IF    9844736.00 < attr_1 <
9869312.00
THEN  class = c2  [0 2]

IF    25.46 < attr_1 < 25.56
THEN  class = c2  [0 1]

IF    4046848.00 < attr_1 <
4327424.00
  AND attr_2 = "Available_Bytes"
THEN  class = c2  [3 6]

IF    attr_1 < 3704832.00
  AND attr_2 = "Available_Bytes"
THEN  class = c2  [1 2]

IF    9971712.00 < attr_1 <
9981952.00
THEN  class = c2  [2 3]

IF    8890368.00 < attr_1 <
9408512.00
THEN  class = c2  [3 4]

IF    attr_1 < 25.46
THEN  class = c2  [1 1]
```

```
IF    9629696.00 < attr_1 <
9693184.00
THEN  class = c2  [2 2]

IF    9900032.00 < attr_1 <
9932800.00
THEN  class = c2  [3 4]

IF    26.46 < attr_1 < 26.66
THEN  class = c2  [1 1]

IF    6064128.00 < attr_1 <
6309888.00
THEN  class = c2  [4 4]

IF    6389760.00 < attr_1 <
6422528.00
THEN  class = c2  [4 4]

IF    9699328.00 < attr_1 <
9730048.00
THEN  class = c2  [2 2]

IF    6438912.00 < attr_1 <
6459392.00
THEN  class = c2  [1 1]

IF    6492160.00 < attr_1 <
6543360.00
THEN  class = c2  [1 1]

IF    45.45 < attr_1 < 45.63
THEN  class = c2  [2 2]

IF    9779200.00 < attr_1 <
9803776.00
THEN  class = c2  [1 1]

IF    27.86 < attr_1 < 28.26
THEN  class = c2  [1 1]

IF    6576128.00 < attr_1 <
6584320.00
THEN  class = c2  [1 1]

IF    6596608.00 < attr_1 <
6623232.00
THEN  class = c2  [2 2]

IF    44.61 < attr_1 < 44.75
THEN  class = c2  [1 1]

IF    6707200.00 < attr_1 <
6725632.00
THEN  class = c2  [1 1]

IF    42.96 < attr_1 < 44.03
  AND attr_3 = "LogicalDisk"
THEN  class = c2  [1 1]

IF    4423680.00 < attr_1 <
4571136.00
  AND attr_2 = "Available_Bytes"
THEN  class = c2  [1 2]

IF    80.78 < attr_1 < 85.58
  AND attr_2 = "Pages_sec"
THEN  class = c2  [2 2]

IF    71.58 < attr_1 < 74.24
  AND attr_3 = "LogicalDisk"
THEN  class = c2  [0 1]

IF    9881600.00 < attr_1 <
9893888.00
THEN  class = c2  [2 1]

IF    6735872.00 < attr_1 <
6785024.00
THEN  class = c2  [3 2]

IF    8435712.00 < attr_1 <
8523776.00
THEN  class = c2  [2 2]

IF    42.96 < attr_1 < 43.62
  AND attr_3 = "PhysicalDisk"
THEN  class = c2  [1 1]

IF    6842368.00 < attr_1 <
6866944.00
THEN  class = c2  [0 1]

IF    87.19 < attr_1 < 88.59
  AND attr_2 = "Pages_sec"
THEN  class = c2  [0 1]

IF    100.57 < attr_1 < 108.20
  AND attr_2 = "Pages_sec"
THEN  class = c2  [2 3]

IF    40.34 < attr_1 < 41.00
THEN  class = c2  [1 1]

IF    62.55 < attr_1 < 63.25
THEN  class = c2  [1 1]

IF    6903808.00 < attr_1 <
6914048.00
THEN  class = c2  [4 1]

IF    8570880.00 < attr_1 <
8652800.00
THEN  class = c2  [1 1]

IF    55.17 < attr_1 < 56.78
  AND attr_3 = "LogicalDisk"
THEN  class = c2  [1 1]

(DEFAULT)* class = c1  [378 72]
```

* The majority rules which covers the training examples.

[c1 c2]- The training examples covered by rules in each classes.

Attributes;
attr1: Thresholds)
attr2: counters
attr3: Objects

Classes;
c1 - HSINGH
c2 - UNKNOWN

Trial 2: Behavioural Profiling

A tool was developed using Microsoft Visual C++ to enable the collection of data for the purpose of this research work. This was to overcome the limitations of NT Performance Monitoring (PerfMon) tool such as setting of alert thresholds, obtaining active application title bar, limited attributes and complexity involve in configuring PerfMon tool for collection of data on networked computers. **Trial 1** data (report submitted on 14/5/00) was obtained using PerfMon tool. The tool developed has two executable one of which acts has a secondary shell and is used to retrieve the active application title bar. The secondary shell is executed once every 3 minutes similarly, the executed primary shell retrieves system resource data once every 3 minutes and the table entry is based on the attributes as depicted in ***Table 1***, the local machine name was used as the class. The results obtained for these trials are divided into two parts, analysis of system resource data and application pattern profiling of individual user(s).

Memory Load	TotalPhysical Memory	Available Memory	%Avail. Memory	Total PageFile	Avail. PageFile	%Avail. PageFile	TotalVirt. Memory	Avail. Virtual Memory	Number of Processes

Table 1: Attributes used for Trial 2

(i) Analysis of system resource data

Data were collected from Paul's, Li Chia's and my networked computers. The data sets were split into ratios of 9:1 and evaluated using the train and test evaluation method. The data sets were divided into 2 files, lcpd (Li Chia and Paul Dowland) and lcpdhs (Li Chia, Paul Dowland and Harjit Singh), 2406 and 2886 sample sizes. The c45, cn2, knn and oc3 algorithms were used for the data mining process. The initial results obtained as depicted in ***Table 3*** suggests that the attributes chosen are suitable for classifying or profiling individual behaviour patterns. The Interpretation process or activity stage used to analyse the rules obtained (refer to ***Appendix***) contradicts this assumption since the data sets were recursively partitioned using only static attributes such as Available Virtual Memory (attr9), Total Pagefile (attr5) and Total Physical Memory (attr2). Hence the Data Mining processes were re-iterated using 5 attributes instead of the 10 attributes initially used. The attributes selected were dynamic and is as depicted in ***Table 2.***

Available Memory	%Avail. Memory	%Avail. PageFile	Virtual Memory	Number of Processes

Table 2: Attributes used for Trial 2

The data sets with varying sample sizes were maintained in two separate files, lcpd1 and lcpdhs1. The results obtained are illustrated in ***Table 3***. The rules obtained (refer to ***Appendix***) are encouraging since the values and attributes selected, reflects on individuals behaviour pattern.

alg(s)	data	samp	attr	class	eval	acc%	secs
c45	lcpd	2406	10	2	tr9\10	100.00	0
cn2	lcpd	2406	10	2	tr9\10	100.00	3
oc3	lcpd	2406	10	2	tr9\10	100.00	0
knn	lcpd	2406	10	2	tr9\10	100.00	1
c45	lcpdl	2406	5	2	tr9\10	100.00	0
cn2	lcpdl	2406	5	2	tr9\10	100.00	2
oc3	lcpdl	2406	5	2	tr9\10	100.00	0
knn	lcpdl	2406	5	2	tr9\10	100.00	1
cn2	lcpdl	2406	5	2	tr9\10	100.00	2
c45	lcpdhs	2886	10	3	tr9\10	100.00	1
cn2	lcpdhs	2886	10	3	tr9\10	100.00	7
oc3	lcpdhs	2886	10	3	tr9\10	100.00	0
knn	lcpdhs	2886	10	3	tr9\10	100.00	1
c45	lcpdhsl	2886	5	3	tr9\10	100.00	0
cn2	lcpdhsl	2886	5	3	tr9\10	100.00	2
oc3	lcpdhsl	2886	5	3	tr9\10	100.00	1
knn	lcpdhsl	2886	5	3	tr9\10	100.00	0

Table 3: Results of classification accuracy

(ii) Application pattern profiling of individual user(s)

The data collected over a 3 days period as illustrated in ***Table 4***, suggests that out of the 7 applications, that have been used by either users, on average 78% of the time a user starts the day by checking his or her email. From the table, it also exhibits short-term patterns of individual users developing over time. These results suggest that the initial theory of incorporating pattern of applications used is a viable attribute in detecting masqueraders if to be incorporated in the IMS.

Next stage of work

1. Investigating behaviour profiling using other attributes such as % cpu usage, etc. for more accurate profiling.
2. Investigating correlation of time intervals between same applications used.
3. Investigating appropriate 3D visualisation of data.
4. Collection of data from other networked computers such as administrative staff and lecturers computers.
5. Write up of initial research work for possible publication in conference proceeding/journal.
6. Continue to trouble-shoot the IDA tool to enable other algorithms to function. This will be done on a separate PC obtained on loan from SoC.

	Li Chia			Paul Dowland			Harjit Singh		
	Day 1	Day 2	Day 3	Day 1	Day 2	Day 3	Day 1	Day 2	Day 3
Applications	Microsoft Outlook	Microsoft Outlook	Internet Explorer	Microsoft Outlook	Microsoft Outlook	Microsoft Outlook	Internet Explorer	Microsoft Outlook	Microsoft Outlook
	Internet Explorer	Internet Explorer	Microsoft Outlook	Microsoft Word	Internet Explorer	Internet Explorer	Microsoft Outlook	Internet Explorer	Internet Explorer
	Microsoft Outlook	Microsoft Outlook		Microsoft Outlook	Microsoft Outlook	Microsoft Outlook	Internet Explorer	Microsoft Outlook	Microsoft Visual C++
	Internet Explorer	Internet Explorer		Internet Explorer	Internet Explorer	Internet Explorer	Microsoft Word	Notepad	
	Microsoft Outlook	Microsoft Outlook		Microsoft Outlook	Microsoft Outlook		Notepad	Internet Explorer	
	Internet Explorer	Internet Explorer		Internet Explorer	Internet Explorer		Microsoft Outlook	Microsoft Visual C++	
	Microsoft Outlook			Microsoft Outlook	Microsoft Outlook		Internet Explorer	Internet Explorer	
	Internet Explorer			Internet Explorer	Internet Explorer		Notepad	Microsoft Visual C++	
				Microsoft Outlook	Microsoft Outlook			Adobe Photoshop	
				Internet Explorer	Internet Explorer				
					Microsoft Word				
					Microsoft Outlook				
					Microsoft Access				
					Internet Explorer				
					Microsoft Outlook				
					Internet Explorer				
					WindowsMedia Player				
					Microsoft Outlook				
					Internet Explorer				
					Microsoft Outlook				

Table 4: Application pattern profiling

Rules from c45 Algorithm

File: lcpd.data

```
Read 2165 cases (10 attributes)
Decision Tree:
attr9 = 2075036: 2 (1082.0)
attr9 = 2079244: 1 (1083.0)
```

File: lcpd1.data

```
Read 2165 cases (5 attributes)
Decision Tree:
attr5 <= 25 : 1 (1082.0)
attr5 > 25 : 2 (1083.0)
```

File: lcpdhs.data

```
Read 2597 cases (10 attributes)
Decision Tree:
attr2 > 64948 : 2 (1084.0)
attr2 <= 64948 :
|   attr2 <= 64940 : 3 (431.0)
|   attr2 > 64940 : 1 (1082.0)
```

File: lcpdhs1.data

```
Read 2597 cases (5 attributes)
Decision Tree:
attr5 <= 25 : 1 (1082.0)
attr5 > 25 :
|   attr4 <= 65 : 2 (1084.0)
|   attr4 > 65 : 3 (431.0)
```

Rules from cn2 Algorithm

File: lcpd.data

```
IF    attr_2 = "64948"
THEN  class = c1  [1082 0]

IF    attr_2 = "97716"
THEN  class = c2  [0 1083]

(DEFAULT) class = c2  [1082 1083]
```

File: lcpd1.data

```
IF    attr_5 < 27.00
THEN  class = c1  [1083 0]

IF    attr_5 > 27.00
THEN  class = c2  [0 1082]

(DEFAULT) class = c1  [1083 1082]
```

File: lcpdhs.data

```
IF    attr_5 < 187366.00
THEN  class = c1  [1083 0 0]

IF    attr_2 > 81332.00
THEN  class = c2  [0 1082 0]

IF    attr_2 < 64944.00
THEN  class = c3  [0 0 432]

(DEFAULT) class = c1  [1083 1082
432]
```

File: lcpdhs1.data

```
IF    attr_5 < 26.50
THEN  class = c1  [1083 0 0]

IF    attr_3 < 189596.00
  AND attr_5 > 29.50
THEN  class = c2  [0 1056 0]

IF    attr_3 < 191284.00
  AND attr_5 > 28.50
THEN  class = c2  [0 1082 0]

IF    attr_3 > 193916.00
THEN  class = c3  [0 0 432]

(DEFAULT) class = c1  [1083 1082
432]
```

Results from oc3 Algorithm

File: lcpd.data

```
Unpruned decision tree
241 testing examples

accuracy = 100.00 #leaves = 2.00
      max depth = 1.00

Category 1: accuracy = 100.00
(120/120)
Category 2: accuracy = 100.00
(121/121)
```

File: lcpd1.data

```
Unpruned decision tree
241 testing examples

accuracy = 100.00 #leaves = 2.00
      max depth = 1.00
Category 1: accuracy = 100.00
(120/120)
Category 2: accuracy = 100.00
(121/121)
```

File: lcpdhs.data

```
Unpruned decision
289 testing examples

accuracy = 100.00 #leaves = 3.00
      max depth = 2.00

Category 1: accuracy = 100.00
(120/120)
Category 2: accuracy = 100.00
(120/120)
```

```
Category 3: accuracy = 100.00
(49/49)

Tree 1: Accuracy=100.00 #leaves=3
Tree 2: Accuracy=85.33  #leaves=2
```

File: lcpdhs1.data

```
Unpruned decision
289 testing examples

accuracy = 100.00 #leaves = 3.00
      max depth = 2.00

Category 1: accuracy = 100.00
(121/121)
Category 2: accuracy = 100.00
(120/120)
Category 3: accuracy = 100.00
(48/48)

Tree 1: Accuracy=100.00 #leaves=3
Tree 2: Accuracy=85.33  #leaves=2
```

Classification of Network State Using Data Mining

H. Singh, K.E. Burn-Thornton* & P.D. Bull*

* Data Mining Group, School of Computing, University of Plymouth, 9 Kirkby Place, Plymouth, Devon, PL4 8AA, UK. Tel: +44 (0) 1752 232711; 232621 Fax: +44 (0) 1752 232540 Emails: hsingh@soc.plym.ac.uk, kburn-thornton@plym.ac.uk Web: http://www.tech.plym.ac.uk/soc/research/dmg

* Wavetek Wandel & Goltermann (WWG) Communications Test Solutions, Eurotech House, Burrington Way, Plymouth, Devon, PL5 3LZ, UK. Tel: +44 (0) 1752 765326 Fax: +44 (0) 1752 783000 Email: phil.bull@wago.de*

ABSTRACT

The emergence of new transport technologies coupled with deregulation and privatisation has contributed to the contiguous growth of telecommunications networks particularly in terms of both the intricacy and size of the network. The rapid growth in network size, and intricacy, is of a concern to those who are involved in Network Management - particularly those involved with network operation, administration, maintenance and provisioning (OAM&P) functions. The integration of the evolving and emerging technologies, and systems, with legacy systems provides additional concerns for those endeavouring to ensure availability of the network resources particularly those required to meet agreed Service Level Agreements (SLAs). In this paper, we discuss the potential use of Data Mining algorithms and techniques, for classifying the Network State, and hence whether SLAs are being met, by analysing performance indicative data collected from networks using the Synchronous Digital Hierarchy (SDH) as an exemplar underlying transmission system.

Keywords
Data Mining, Network Management, Telecommunications, Service Level Agreement (SLA), Knowledge Discovery, Network State, Alarm Correlation, Synchronous Digital Hierarchy (SDH), Networks, Pro-active Management, Quality of Service (QoS).

1. INTRODUCTION

The setting up of SLAs demands a high standard of network availability and performance through improved quality management systems. The current practises of network management can overload network management architectures and thus agreed quality of service (QoS) [1] defined in the SLA contracts. This is due to the various processes enacted in the Maintenance Function [2] which involves–monitoring, and managing, extremely large amounts of data as a result of the sheer size, and complexity of the networks. Data that is collected in order to achieve QoS, is determined via analysis of the network performance indicators. This can involve analysis of voluminous samples of data collected from monitoring of network performance, and alarms used to resolve or avoid faults [3], which is collected for the maintenance function. Typically, for example, performance information collected routinely for an ATM network every 15 minutes amounts to 15MB [4].

The onset of overloading network operation centre is the availability of the network, which is dependent on the restoration and preventative [5] (e.g. re-routing the network configuration) functions enact in the network management processes. Network elements (NEs) affect each other and consequence or sequential [6] generation of notification messages is inevitable as a result of the occurrence of faults. A single incident, or fault, due to a particular anomaly or defect may trigger multiple generation of notification messages. These notification manifests externally as alarms and depending on the protocol used by the Management systems, an alarm is referred to as a trap or notification [7], where the later using CMIP, while the former using SNMP. When considering the network complexity, which can consist of several hundred to thousands of NEs, this represents a large amount of alarms. Prominently, this can cause alarm inundation of a network operation centre and has strong antecedents in fault localisation. Hence analysis of the alarms enabling, hypothesis of the root cause of the fault to be proposed.

Performance information collected routinely and alarms, in addition to alarm log which, contains information about an alarm event entered in chronological sequence, provides a huge repository of raw data from networks. Using domain expert to analyse this data can be exhaustive and time consuming especially the classification of the state of the network and the discovery of latent trends or patterns to resolve transient problems. Consequently, alarm correlation [8] has been one of the many correlation techniques employed in fault localisation. Alarms, with germane entities or trends are grouped together to form new semantics and the parenthesis is enriched to provide useful information pertaining the generation of the alarms which, may include corrective measures. Hence,

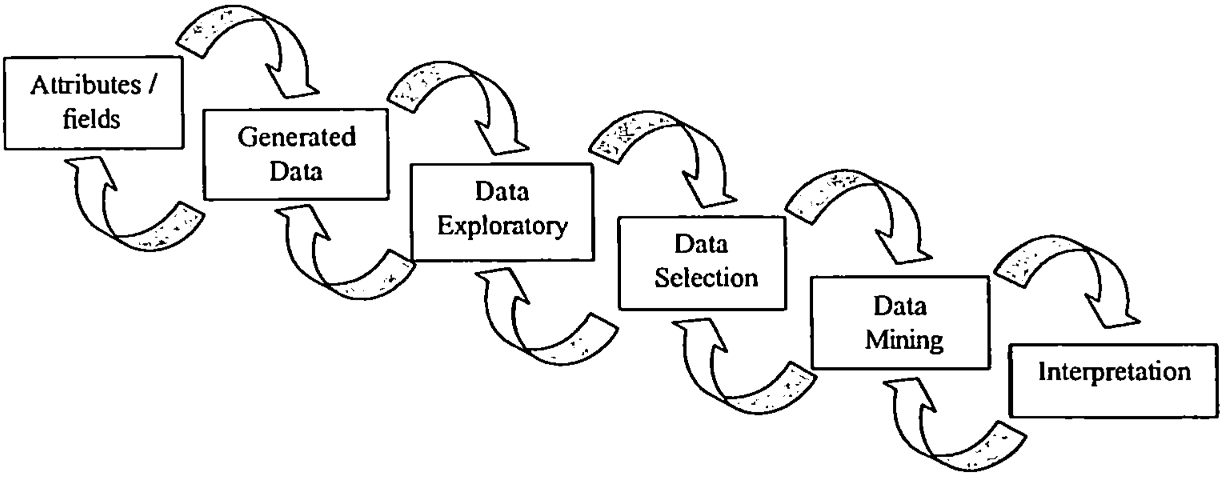


Figure1: Methodology for alarm correlation

this reduces the amount of information displayed and provides plausible information, to enable the subtle task of fault diagnosis.

In this paper we, discuss the potential of Data Mining algorithms, and techniques for the prediction of problems that are likely to persist, as well as those that are likely to degrade network performance – hence enabling classification of the network performance indicators. In section 2, we provide a conceptual introduction to Data Mining. This is followed by description of the methodology used which, is the focus of this work; the Data Mining of SDH network performance indicators in the telecommunication domain. Section 4 provides the results of this investigative work while section 5 provides the conclusion and discussion. Future work is also discussed in section 6.

2. DATA MINING

Data Mining (DM) can be described as a collection of techniques and methodologies used to explore vast amounts of data in order to find potentially useful, ultimately understandable patterns [9] and to discover relationships. DM is an iterative and interactive process, involving numerous steps with many decisions being made by the user. The fundamental goals of data mining are finding latent trends in data, which enables prediction and description [10] of the analysis phases. DM is a rapidly expanding field which, has been exploited in lucrative domains such as in the financial [11] business [12] and communications [13] domains although little reported work has been carried out to determine network state by analysis of the network performance indicators of alarms [14-17].

2.1 Data Mining task

The initial sequence upon acquiring the required understanding of the proposed application domain or *priori* knowledge is to determine the DM task. Different algorithms are optimised based on the predefined DM task. This involves deciding whether the goals of the DM process is classification, association, or sequential [18]. Classification has two distinct meanings. We may aim of classifying new observations into classes from established rules or establishing the existence of classes, or clusters in data [19]. Association attempts to generate rules or discover correlation in data and is expressed:

- X => Y, where X and Y are sets of items.

This means that an event or transaction of database that contains X tends to contain Y.

Sequential looks at events occurring in a sequence over time or time-ordered sequences. This could be expressed through the following:

- E ? N, E is a set of event types, an event pair (A, t), where A ? E is an event type.

Where t represent the time of the event or occurrence of an event. This is followed by predefined sets of fault conditions where:

- F is a set of fault types and C ? F, C is a fault type, hence for example:

90% of the time, if the event (A, t) occurs, it is followed by fault type C.

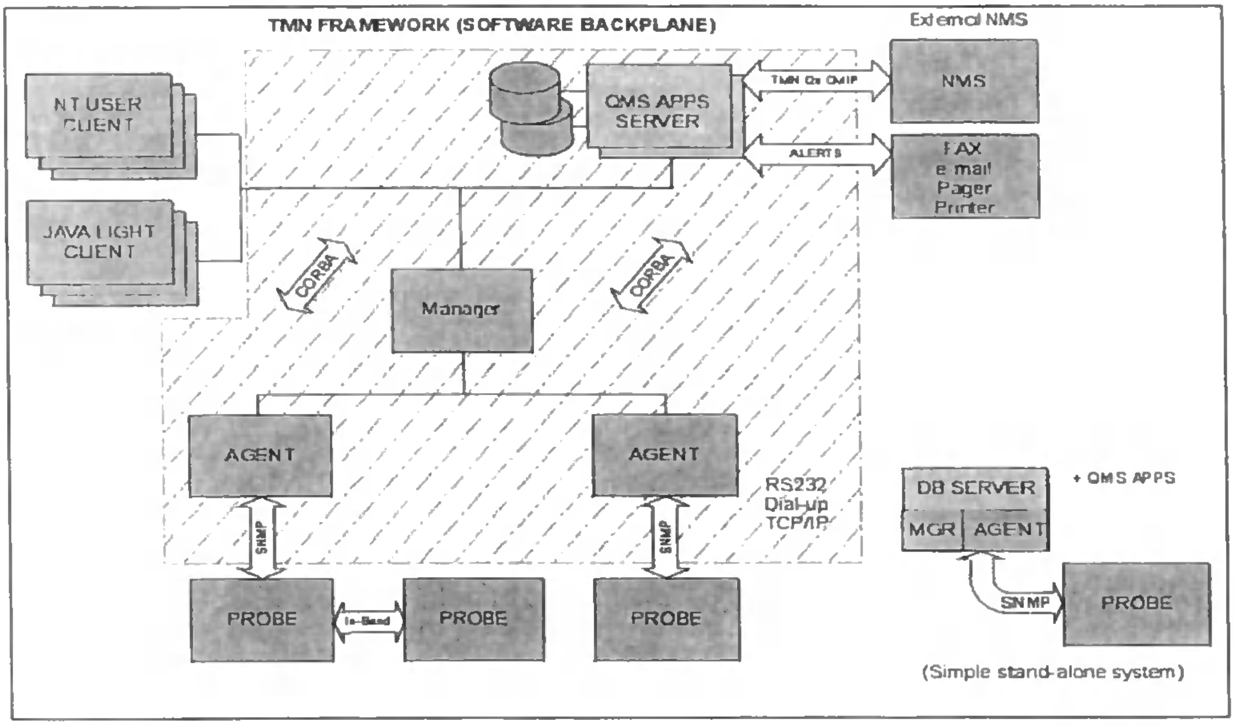


Figure 2: WWG QMS System Architecture

2.2 Techniques and methodology
The subsequent process once the DM task is defined can be derived from the four main activities; *selection, pre-processing, data mining and interpretation*, also known as post-processing.

Selection involves creating a target data set to undergo analysis, paradoxical to the assumption that the complete raw data is presented to the DM software, due to the nature of the data which may present irrelevant attributes or information pertinent to other mien of the domain. The data set selected can be focused on a subset of variables or data samples, of the proposed application domain. The recent and historical data can be found, combined, and transformed into an organised target data set repository.

Pre-processing in DM involves preparing the target data set prior to undergo analysis using the Data Mining software. It may involve converting the data into acceptable format to the data mining software, demarcation of the beginning of each message or may involve processes to discreetise continuous features for instance, as used to generate the alarm semantic. Pre-processing can be used to constrain the search space, and can make patterns or relationships in the data more visible in the later stages of the DM process. The *data mining* process involves subjecting the cleaned (containing reliable information) data to be analysed by the data mining algorithm(s) and results of the analysed (mined) data is presented to the next stage.

Interpretation involves verification of the results, hence analysing the results of the analysis which may include selecting interesting rules based on the DM task. It may also involve re-iterating some of the processes in order to provide further information. The elicited interpreted analysis can be implemented or adapted in a correlation framework or system for pro-active management applications particularly, which requires real-time data as in the telecommunications domain for fault diagnosis.

3. METHODOLOGY

The Data Mining methodology for the determination of network state is determined via analysis of the network performance indicator as depicted in ***Figure 1*** and is derived from the four basic activities carried out during Data Mining activities or processes; selection, pre-processing, data mining and interpretation. For the focus of this work the pre-processing stages, constitute the activities of generating the data and data exploratory on the data set. These are the important stages prior to selecting the target data set, which will ITU-T G703, G704 and domain specific performance indicators.

3.1 Alarm semantic

The semantic of the alarm is based on Wavetek Wandel & Goltermann (WWG) Quality Management System (QMS). QMS fault management concept and semantic is based on ITU-T X.745 and X.733 recommendations respectively. WWG QMS system architecture and relationships between each of the main functions is as depicted in ***Figure 2***. WWG QMS enables the end-to-end visibility of network and service performance to achieve the desired levels of service quality and honour SLA commitments as conceptually depicted in ***Figure 3***.

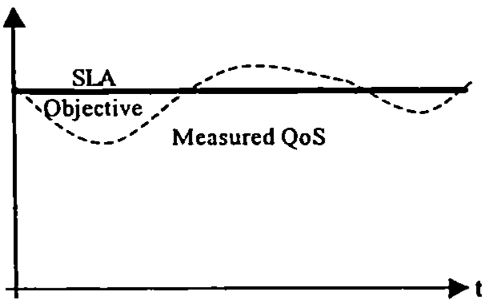


Figure 3: SLA and QoS relationship

3.2 Data Mining algorithms

The Intelligent Data Analysis (IDA) Data Mining Tool [19] used for the data mining process incorporates algorithms from the fields of Statistical, Machine Learning and Neural Networks. The five algorithms chosen for this investigative work are k-NN, C4.5, CN2, RBF and OC1. The generated data sets were split into ratios of 9:1, 8:2 and 7:3, hence into two parts; a training set and a testing set, which is a common technique used known as train and test. The algorithms or classifier is subjected initially with the training set and then the classification accuracy is tested using the unseen data set or testing set. The results give an indication of the error rate and the overall classification accuracy of the trained algorithms.

4. RESULTS

The result of this initial investigative work suggests that the Machine Learning algorithms, C4.5 and CN2 in particular, performed better. This results, is in comparison with the other algorithms used for this initial trials, explicit details on this is beyond the scope of this paper but will be included in future papers. C4.5

undergo further analysis to identify patterns and to test specific hypotheses. The alarm types used in this investigative work are based upon recommendation of

is a decision tree Machine Learning based algorithm and uses a modified entropy measurement to calculate the gain in information. Entropy is used to measure how informative a node, by splitting the data at various boundaries. The output result of the *data mining* process upon subjecting the data to undergo analysis by the DM tool could be in the following form:

```
Attribute 6 = Location 15: 1 (2.0)
Attribute 6 = Location 13:
| attribute 4 = Direction_1: 1 (5.0/2.0)
| attribute 4 = Direction_2: 2 (3.0)
```

This output can be paraphrased as:

```
If attribute 6 = Location 15 then 1
If attribute 6 = Location 13 then
   If attribute 4 = Direction_1 then 1
   else
   If attribute 4 = Direction_2 then 2
```

The (m/n) gives both the number of correctly and incorrect mapped data, entered in the node's leaf.

CN2 is a rule based Machine Learning algorithm, which, like C4.5 belongs to the general class of recursive partitioning algorithms. The CN2 algorithm for this investigative work was optimised to generate *unordered* rule set using the Laplace statistical significance prediction in generating the rules. The former enables the search in each class to be re-iterated, removing only covered examples of that class when a rule has been found. Hence, having found a good conjunct of attribute or complex, the rule 'if <complex> then predict <class>' is added to the end of the rule list. The latter tends to avoid the undesirable 'downward bias' of entropy [21]. The rule for example could be as following:

```
IF    attribute 1 = "Type_44"
  AND  attribute 6 = "Location6"
THEN  class = c3   (9 6 3)
```

The (a b c) in this case gives the training examples covered by rules in each classes, for this example three classes were used.

5. CONCLUSION AND DISCUSSION

The growth in network size, integration of multiple sub-networks and many different vendors' equipment, makes the task to elicit the required *priori* knowledge a difficult task. Consequently, the task to disambiguate the alarms based on hypothesis tends to become more knowledge intensive since, the alarm semantic are not explicit enough to provide important information required for the diagnosis of fault and can be ambiguous. Hence, additional information is required which even to the most erudite domain expert, this proves to be a subtle task. The development of efficient and efficacious tools as well as methodology, are

6. FUTURE WORK

The initial methodology, based on the results of the trials which, will be used, will be further developed and integrated into a correlation systems framework or intelligent data analysis tool. This could be incorporated, into or form, part of a network monitoring or test equipment to enable pro-active network management.

ACKNOWLEDGEMENT

The authors would like to thank Wavetek Wandel & Goltermann (WWG) Communications Test Solutions of Plymouth (UK), DTI and the EPSRC for the financial support for this project.

REFERENCES:

[1] D. Sanchez, D. Guerrero and A. Vina, "Modelling Legacy PDH Equipments from General Models to Real TMN Solutions", proceedings of MOMS'98, 186-195.

[2] A. Mahdi, K.E. Burn-Thornton and P. Bull, "Expert Diagnosis Engine for Remote Test Management of Telecommunications Network", proceedings of INC'98, ISBN 1-84104-016-8, 179-184.

[3] G. Jakobson, M. Weissman, "Alarm Correlation", IEEE Networks-1993, Vol.7, No.6, 52-59.

[4] K.E. Burn-Thornton, J. Garibaldi and A. Mahdi, "Pro-Active Network Management using Data Mining", GLOBECOM '98, 1-9.

[5] R. Gardner and D.A. Harle, "Expert Data Mining For Alarm Correlation in High-Speed Networks", proceedings of IITT EXPERTSYS '97, 145-150.

[6] H. Mannila, H. Toivonen and V.A. Inkeri, "Discovery of Frequent Episodes in Event Sequence", Data Mining and Knowledge Discovery-1997, Vol1, No.3, 259-289.

[7] S. Aidarous and T. Plevyak, "Telecommunications Network Management into the 21st Century Techniques, Standards, Technologies and Applications", IEEE Press-1993, ISBN 0-7803-1013-6, 1-17.

[8] T.A.M. De Castro and J.M.S. Nogueira, "An Alarm Correlation System for SDH Networks", Proceedings of IT'98, 492-497.

[9] U.M. Fayyad, "Data Mining and Knowledge Discovery; Making Sense Out of Data", IEEE Expert-1996, Vol.11, No.6, 20-25.

[10] P. Adriaans and D. Zantinge, "Data Mining" Addison-Wesley-1996, ISBN 0-201-40380-3.

[11] C. Westphal and T. Blaxton, "Data Mining Solution, Methods and Tools for Solving Real-World Problems", Wiley-1998, ISBN 0-471-25384-7, 531-585.

required in order to allow a pro-active or to compliment human interpretation of some of the functions enacted for the purpose of network management. The initial results, from this investigative work provides a framework for some of these tasks.

[12] F. Giannotti, G. Manco, M. Nanni, D. Pedreschi and F. Turini, "Integration of Deduction and Induction for Mining Supermarket Sales Data", proceedings of PADD'99, ISBN 1-902426-04-5, 179-93.

[13] R. Sasisekharan and V. Seshadri, "Data Mining and Forecasting in Large-Scale Telecommunications Networks", IEEE Expert Intelligent Systems and Their Applications-1996, Vol.11, No.1, 37-43.

[14] R. Gardner and D.A. Harle, "EventFlow: A Technology for Alarm Correlation in High-Speed Networks", proceedings of INC'98, ISBN 1-84104-016-8, 171-178.

[15] S. Hajela, "Alarm Management in Telecommunications Networks", Hewlett Packard Journal-1996, Vol.47, No.5, 22-30.

[16] A.T. Bouloutas, S. Calo and A. Finkel, "Alarm Correlation and Fault Identification in Communications Networks", IEEE Transactions on Communications-1994, Vol.42, No.2/3/4, 523-533.

[17] G. Jakobson, G. Weihmayer and M. Weissman, "A Domain Oriented Expert System Shell for Telecommunications Network Alarm Correlation", proceedings of 2nd IEEE Network Management and Control Workshop-1993, Vol.2, 365-380.

[18] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy, "Advances in Knowledge Discovery and Data Mining" AAAI Press / The MIT Press-1996, ISBN 0-262-56097-6, 1-31.

[19] D. Michie, D.J. Spiegelhalter and C.C. Taylor, "Machine Learning, Neural and Statistical Classification", Ellis Horwood-1994, ISBN 0-13-106360-X, 6-16.

[20] K.E. Burn-Thornton and L. Edenbrandt, "Myocardial Infarction-Pinpointing the Key Indicators in the 12 lead ECG Using Data Mining", Journal of Computers and Medicine-1998, Vol.31, No.31, 293-303.

[21] P. Clark and R. Boswell, "Rule Induction with CN2:Some recent Improvements", proceedings of the 5th European Conference (EWSK-91), 151-163.

APPENDIX C

PUBLICATIONS

During the course of this research project, the author has written to 4 published papers, as detailed below.

1. H.Singh, K.E.Burn Thornton, P.D.Bull. 1999. "Classification of Network State Using Data Mining", *Proceedings of 4th IEEE International MICC & ISCE Conference-1999* Vol. 1, pp183-187, 1999.

2. H.Singh, S.M.Furnell, B.Lines, P.S.Dowland. 2001. "Investigating and Evaluating Behavioural 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.

3. P.S.Dowland, H.Singh, S.M.Furnell. 2001. "A Preliminary Investigation of User Authentication Using Continuous Keystroke Analysis", *Proceedings of the IFIP 8th Annual Working Conference on Information Security Management & Small Systems Security*, Las Vegas 27-28 September, 2001.

4. H.Singh, S.M.Furnell, P.S.Dowland, B.Lines and S. Kaur. 2004. "A Correlation Framework for Continuous User Authentication Using Data Mining", accepted for inclusion in *Proceedings of the Fourth International Network Conference (INC 2004)*, Plymouth, UK, 6-9 July 2004.

In addition, the author has contributed to the following conference poster presentations:

1. H.Singh. 2000. "Behavioural Profiling and Intrusion Detection Using Data Mining", Second International Network Conference (INC 2000), University of Plymouth, Plymouth, UK, 3-6 July 2000.

2. P.Dowland, S.Furnell, G.Magklaras, M.Papadaki, P.Reynolds, P.Rodwell and H.Singh. 2000. "Advanced Authentication and Intrusion Detection Technologies", Britain's Younger Engineers in 2000, House of Commons, London, 4 December 2000.

Copies of these materials are bound within this appendix of the thesis.

Investigating and Evaluating Behavioural Profiling and Intrusion Detection Using Data Mining

Harjit Singh, Steven Furnell, Benn Lines and Paul Dowland

Network Research Group, Department of Communication and Electronic Engineering,
University of Plymouth, Drake Circus, PLA 8AA, United Kingdom
{hsingh, sfurnell, blines, pdowland}@plymouth.ac.uk

Abstract. The continuous growth of computer networks, coupled with the increasing number of people relying upon information technology, has inevitably attracted both mischievous and malicious abusers. Such abuse may originate from both outside an organisation and from within, and will not necessarily be prevented by traditional authentication and access control mechanisms. Intrusion Detection Systems aim to overcome these weaknesses by continuously monitoring for signs of unauthorised activity. The techniques employed often involve the collection of vast amounts of auditing data to identify abnormalities against historical user behaviour profiles and known intrusion scenarios. The approach may be optimised using domain expertise to extract only the relevant information from the wealth available, but this can be time consuming and knowledge intensive. This paper examines the potential of Data Mining algorithms and techniques to automate the data analysis process and aid in the identification of system features and latent trends that could be used to profile user behaviour. It presents the results of a preliminary analysis and discusses the strategies used to capture and profile behavioural characteristics using data mining in the context of a conceptual Intrusion Monitoring System framework.

Keywords

Data Mining, Intrusion Detection Systems, Knowledge Discovery, Behavioural Profiling, Intelligent Data Analysis.

1 Introduction

The increasing reliance upon IT and networked systems in modern organisations can have a calamitous impact if someone deliberately sets out to misuse or abuse the system. Systems may be affected by internal and external categories of abuser, as a result of both mischief and malice, leading to a range of undesirable consequences for the affected organisations (e.g. disruption to activities, financial loss, legal liability and loss of business goodwill). A recent study conducted by the US Computer Security Institute (CSI), in collaboration with the FBI, reported that 70% of respondent organisations had detected unauthorised use of their computer systems in the previous 12 months [1] – which represented an 8% increase on previous findings from 1999. The level of reported incidents highlights the paucity of security measures in current systems and, hence, the need for more comprehensive and reliable approaches. In particular, it can be suggested that traditional user authentication and access controls (e.g. passwords and user/group-based file permissions) are not sufficient to prevent determined cases of abuse or re-occurrence, in the case of successfully breached account(s), and misuse occurring from a legitimate user. Having passed the frontline controls and having the appropriate access privileges, the user may be in the position to do virtually anything without being further challenged. However, appropriate monitoring and analysis of user activity within an active session may potentially reveal patterns that appear abnormal in relation to their typical behaviour, or which are compatible with the sign of recognised intrusion scenarios. It is from this perspective that many Intrusion Detection Systems (IDS) have been conceived. Various IDSs [2, 3] have been proposed, which generally can be categorised based on the data source, audit trails or network traffic data, and intrusion model employed, anomaly detection or misuse detection model. The approaches used are generally focused on providing continuous monitoring and involve analysing vast amounts of audit trails, which in an eight-hour period can amount to 3-35MB [4] of data generated.

There is an increasing need for a more coherent paradigm for audit processing in terms of automating the data analysis stages. The current trend of network components providing audit trail or audit logs provides the foundation for IDSs to explore database automated match and retrieval technologies. This can be seen in audit processor components for instance the SecureView in the Firewall-1 using Data Mart to store the audit trails [5]. This available information could be used for security audit trail analysis in IDSs by utilising the technology in the data analysis stages. The need to eliminate the manual and ad-hoc approaches in the data analysis stages in IDSs is attracting interest in applying Intelligent Data Analysis (IDA) techniques. In this paper is discussed the potential of Data Mining (DM) algorithms and techniques as an IDA tool. We use DM to automate the data analysis process in identifying system features and latent trends for classifying user behaviour from the collected audit trails. DM is a rapidly expanding field which, has been exploited in lucrative domains such as the financial [6] and communications [7] sectors. Although some reported work has been carried out to analyse network traffic data [8, 9], none has been carried out in analysing host-based audit trails using DM for the purpose of user authentication, which is the focus of this research work.

2 Data Mining

Data Mining can be described as a collection of techniques and methodologies used to explore vast amounts of data in order to find potentially useful, ultimately understandable patterns [10] and to discover relationships. DM is an iterative and interactive process, involving numerous steps with many decisions being made by the user. The fundamental goals of data mining are finding latent trends in data, which enable prediction and description [11] of the analysis phases. Different algorithms are optimised based on the predefined DM task. This involves deciding whether the goals of the DM process are classification, association, or sequential [10]. Classification has two distinct meanings. We may aim to classify new observations into classes from established rules or establishing the existence of classes, or clusters in data [12]. Association attempts to generate rules or discover correlation in data and is expressed: X => Y, where X and Y are sets of items. This means that an event or a transaction of a database that contains X tends to contain Y. Sequential looks at events occurring in a sequence over time or time-ordered sequences. This could be expressed through the following: for E ? N, E is a set of event types and an event is a pair (A, t), where A ? E is an event type and t represents the time of the event or occurrence of an event. This is followed by predefined sets of possible intrusion classes where, C is a set of intrusion classes and I ? C, I is an intrusion type, hence for example: 90% of the time, if the event (A, t) occurs, it is followed by intrusion type I. The subsequent process, once the DM task is defined can be derived from the four main activities; *selection,. pre-processing, data mining and interpretation*, also known as post-processing [12].

3 Classification of User Behaviour

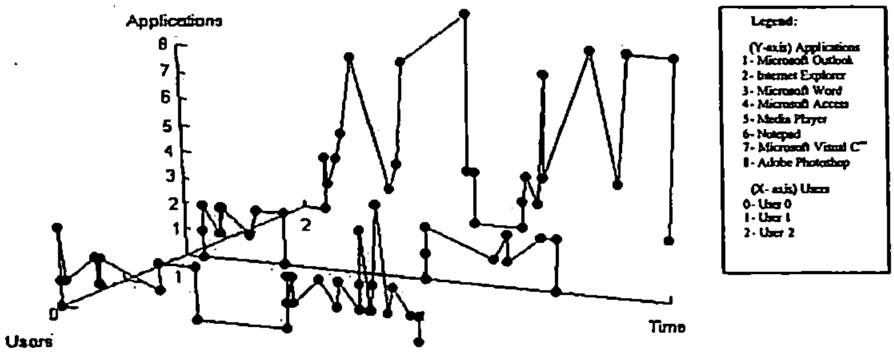


Figure 1. Graphical representation of applications run by users

Distinguishing user behaviour patterns and classifying it as normal or intrusive is a subtle task. Furthermore exploring the vast amount of audit trail data often yields a small fraction of intrusion or misuse. Besides managing these tasks, IDSs have to limit the errors that could occur from misclassification of user behaviour such as false positive or false negative errors. Therefore, it is essential to ensure that accurate profiles of users are established in order to improve the accuracy of intrusion classification. Hence the need to gather as much information as possible pertinent to a user's interaction with the system in order to distinguish between similar behavioural patterns of users that could occur. Auditing the applications that users run could for instance provide a distinctive pattern of the user's interaction with the system as depicted in ***Figure 1.*** Users pattems once identified could be incorporated into an anomaly detector framework in conjunction with other key indicators of user behaviour in order to identify unauthorised access when compared against this distinctive usage pattems. Hence it is essential to collect as much information as possible regarding such behavioural indicators in order to correlate the possibility of intrusion.

4 Methodology

We use DM to extract latent pattems or models of user behaviour from the collected audit trail. This is then reflected in the DM algorithm classifiers (e.g. through rule induction) to recognise deviation, if it occurs, from normal use. This approach is based on the assumption that a user's behaviour has regularity and that using the classifiers this behaviour can be modelled. Using this analogy, anomalous behaviours can then be categorised as a possible unauthorised user or use of that system. The audit trail data analysed was collected from networked computers on a participating local area network (LAN) using an independent agent installed locally in order to audit user interaction with the system. This is based on the assumption that users performing their regular tasks will impose similarly regular demands upon system resources. Hence system features involved for continuous monitoring of user interaction with the system such as resource usage, process-related information such as creation, activation and termination, etc, is audited. Similar system features have been used in other published work [2]. However, previous work was focused on statistical and neural network analysis. A user's behaviour profile can be uniquely identified by: <user name, absolute time, date, hostname, $\text{event}_{1,\ldots,}\text{event}_n$ >, which is the semantic used for the audit trail where, $\textit{events}_n$ denotes the system features being monitored.

4.1 Data Mining audit trail

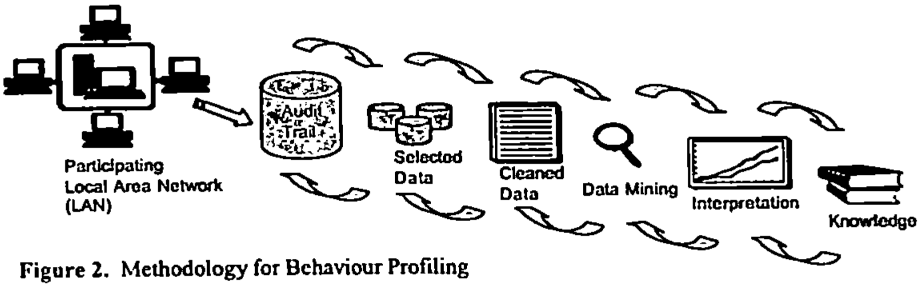


Figure 2. Methodology for Behaviour Profiling

The methodology used is derived from the four main activities of DM; selection, pre-processing, data mining and interpretation, and is as depicted in ***Figure 2.*** The collected audit trail is split into various sample sizes. These subsets form the target data sets, which will undergo the analysis to identify patterns and to test specific hypotheses. The cleaned data, containing both categorical and numerical data, is then subjected to analysis by the DM algorithms. There are a wide variety of DM techniques available, each of which performs more accurately over certain characteristic data sets (e.g. numerical or categorical) and is also relative to the number of variables or attributes and classes. The Intelligent Data Analysis (IDA) Data Mining Tool [13] is used to analyse the sample data sets which incorporates algorithms from the fields of Statistical, Machine Learning and Neural Networks. Six algorithms, k-NN, COG, C4.5, CN2, OC1 and RBF were chosen for this investigative work. For the purpose of this work, the data sets were split into ratios of 9:1, 8:2 and 7:3, hence into two parts, which is a commonly used technique known as train and test. The algorithm or classifier is subjected initially with the training set and then the classification accuracy is tested using the unseen data set or testing set. The results give an indication of the error rate (or false positives) and the overall classification accuracy of the trained algorithms.

5 Results

The initial results obtained from the analysis as depicted in ***Figure 3***, suggest that Machine Learning and Statistical-based algorithms are better for these types of data sets. C4.5 and OC1 decision tree based algorithms in particular, out performed the CN2 rule-based and RBF algorithms. The classification accuracy obtained, using k-NN in comparison to C4.5, shows some significance for further investigative work despite the slower classification times observed. Amongst the statistical algorithms, k-NN faired better then COG but is slower in comparison to the classification times observed. The classification accuracy obtained overall depicts RBF classification accuracy as inverse proportional to the sample sizes. These results support other reported work [12]. In addition to the consistency in classifying the data sets and the overall average classification accuracy, our initial investigations also identified that C4.5 has overall quicker train and test time and outputs explicit rules.

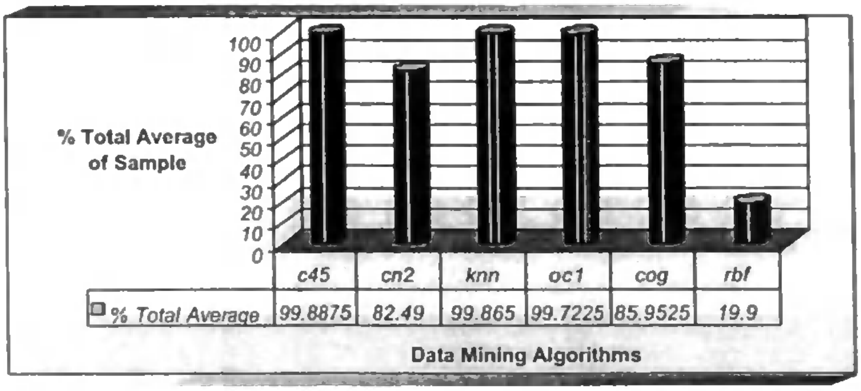


Figure 3. Total percentage average classification accuracy of selected Data Mining algorithms

6 Discussion and conclusion

The classification accuracy obtained suggests that DM techniques could be integrated into an IDS framework in order to provide a mechanism to detect intrusions. The approach used in these initial trials has shown the potential that DM techniques can be used to detect anomalies or intrusions through the behaviour model generated by the DM algorithm's classifiers. The high classification accuracy obtained and fast response time exhibited in classifying the user behaviour by some of the DM algorithms further demonstrates the potential of applying DM techniques within a real-time application for identifying intrusions [14]. Another important element identified is the interpreted rules obtained from the data mining process. The systems features outlined by the classifiers to detect anomalous behaviour can be used to detect known intrusions. The results so far have been based around the classifiers used that are optimised to classify either new observed user behaviour into classes from established rules or establishing the existence of classes using the DM algorithms. While this has been the fundamental goal in our approach, another important aspect of identifying user behaviour from frequent patterns developing over time has yet to be addressed.

References

1. Computer Security Institute, "2000 CSI/FBI Computer Crime and Security Survey", Vol. 6, No.1, SPRING-2000.
2. T.F. Lunt, "IDES: an intelligent system for detecting intruders", Proc. of the Computer Security, Threat and Countermeasures Symposium, November 1990 Rome, Italy.
3. B. Mukherjee, L.T. Herberlein and K.N. Levitt "Network Intrusion Detection", IEEE Network-1994, Vol. 8, No. 3, 26-41.
4. J. Frank, "Artificial Intelligence and Intrusion Detection: current and future direction", Proc. of the 17th National Computer Security Conference, October 1994.
5. E.G. Amoroso, "Intrusion Detection: an introduction to internet surveillance, correlation, traps, trace back, and response", Intrusion.Net-1999, ISBN 0-9666700-7-8.
6. C. Westphal and T. Blaxton, "Data Mining Solution, Methods and Tools for Solving Real-World Problems", Wiley-1998, ISBN 0-471-25384-7, 531-585.
7. R. Sasisekharan and V. Seshadri, "Data Mining and Forecasting in Large-Scale Telecommunications Networks", IEEE Expert Intelligent Systems and Their Applications-1996, Vol.11, No.1, 37-43.
8. W. Lee and S. Stolfo, "Data Mining Approaches for Intrusion detection", Proc. 7th USENIX Security Symposium, 1998.
9. C. Warrender, S. Forrest, B. Pearlmutter, "Detecting Intrusion Using Calls: alternative data models", Symposium on Security and Privacy, 1999.
10. U. M. Fayyad, "Data Mining and Knowledge Discovery: making sense out of data", IEEE Expert-1996, Vol.11, No.6, 20-25.
11. P Adriaans and D. Zantinge, "Data Mining", Addison-Wesley-1998 , ISBN 0-201-40380-3.
12. D. Michie, D.J. Spiegelhalter and C.C. Taylor, "Machine Learning, Neural and Statistical Classification", Ellis Horwood-1994, ISBN 0-13-106360-X, 136-141.
13. H. Singh, K.E. Burn-Thornton and P.D. Bull, "Classification of Network State Using Data Mining", Proc. of the 4th IEEE MICC & ISCE '99,Malacca, Malaysia, Vol.1, 183-187.
14. S.M. Furnell and P.S. Dowland, "A Conceptual Architecture for Real-time Intrusion Monitoring", Information Management & Computer Security-2000, Vol. 8, No. 2, 65-74.

A Preliminary Investigation of User Authentication Using Continuous Keystroke Analysis

P. S. DOWLAND[1], H. SINGH[2], S. M. FURNELL[3]

[1]pdowland@plymouth.ac.uk
[2]hsingh@jack.see.plym.ac.uk
[3]sfurnell@plymouth.ac.uk
Network Research Group
Department of Communication and Electronic Engineering
University of Plymouth
Drake Circus
PLYMOUTH
PL4 8AA
United Kingdom
Tel: +44 1752-233521   Fax: +44 1752-233520

Key words: Keystroke Analysis, Authentication, Biometrics, Data Mining.

Abstract: There has been significant research in to the provision of reliable initial-login user authentication, however there is still a need for continuous authentication during a user session. This paper presents a series of results from the preliminary statistical analysis of multi-application keystroke data. This has been contrasted with a Data Mining approach to the production of a unique user profile. This paper aims to determine which approach provides the best basis for further research. It is determined that the technique offers promise as a discriminator between individuals in an operational context, but further investigation with larger data sets is required with a combination of approaches being considered in order to improve the accuracy.

1. INTRODUCTION

There have been a number of previous studies that have considered the security weaknesses in modem IT system and, whilst various recommendations and technical solutions have been proposed, many still rely on enhancing the initial login-stage mechanism (e.g. via biometric identification, smart cards etc.) [COPE 90, SHER 92, MILL 94]. Whilst this improves the initial authentication judgement, there is still a need for user authentication throughout a session. In most systems there is no further check on a users' identity beyond the initial username/password. Once a user gains legitimate access to IT resources, it is feasible for there to be no further challenge, with the only possibility for detection of a masquerader being the post-event detection of a major incident (i.e. an impostor can masquerade as the valid user without detection or challenge).

To counter this risk, it is suggested that some form of user monitoring is desirable to continuously (or periodically) authenticate the user in a transparent manner. Whilst such monitoring is technically feasible, there are significant issues to be considered in selecting appropriate attributes to assess. This is particularly important, as continuous monitoring must be transparent to the end user in order to minimise any perceived inconvenience (with the exception of appropriate challenges in the event of a significant profile deviation).

This paper specifically considers the problems of continuous user authentication using keystroke digraph latencies. This area has not received much attention and as such, most of the background research is based upon static keystroke analysis [JOBU 89, BROW 93, JOYC 90] (i.e. where the users' typing was constrained). Keystroke analysis is, however, considered by end users as the most acceptable form of continuous authentication [FURN 00]. A GUI environment produces a new challenge, as there is no option to control the users' typing. This can cause problems, as it is difficult to determine in which application individual digraph pairs were entered. This paper will introduce a statistical approach for detecting deviation from a user's historical keystroke profile captured under a multi-tasking windowed environment. Following this initial analysis, a Data Mining (DM) approach will be considered in order to determine the potential for improving user classification. It should be noted that the aim of this paper is to determine which approach provides the best basis for further research and is not intended as a thorough analysis of keystroke latencies for user authentication. Finally, some thoughts on future work are introduced which will be developed further.

2. EXPERIMENT OVERVIEW

Although there have been a series of papers describing the mechanisms for keystroke analysis, the authors have been unable to identify any research specifically focussed on continuous keystroke analysis in which the collection of users typing samples was not artificially constrained in some way through a custom interface (e.g. asking the user to type known strings).

The experiment was designed to allow keystroke data to be collected under the Microsoft Windows NT environment. In order to collect the required data, it was necessary to implement a mechanism for acquiring keystroke notifications across all applications running within a users' active session. As the client systems were running Microsoft Windows NT v4.0, it was necessary to implement a system-wide hook function that would receive keyboard events through the Windows message chain. System-wide hooks allow a specified code block (hook-function) to receive the appropriate Windows messages (e.g. WM_KEYUP for the key-up event) irrespective of the target application (i.e. it is possible for a hook function residing in a system DLL to receive keystroke notifications for all currently running applications). This effectively allowed application keystroke data to be duplicated and directed towards the data logger on the client workstation. Technical details of the implementation of the hook function and its associated support files are beyond the scope of this paper and, as such, have been omitted. There are a number of resources available that provide further information for interested readers [DOWL 00, MICR 00]. In order to determine accurate digraph latencies, it was also necessary to implement a high-accuracy timer (as the default timers available do not offer adequate accuracy for the millisecond latencies expected).

To eliminate extreme short/long digraph latencies that may adversely affect the distribution of digraph times, any digraph pair whose latency fell outside a nominal range was excluded from the archived data. For the purposed of this experiment the range was restricted to times above 40ms and below 750ms. These thresholds are based on the original experiments carried out by the authors [FURN 95] and are designed to eliminate samples where two keys may have been accidentally struck together (thus, producing an infeasibly small latency) or, where the user may have made a pause in their typing and thus introduced an unnaturally large inter-keystroke latency. The output of this pre-processing was a data file containing the following structure:

first_char *second_char* *digraph_latency*

For this experiment a total of ten users were profiled. As the intention was to evaluate the analysis mechanisms without implementing a large-scale trial, tests were carried out using a small set of test subjects. The main limiting factor was the need to collect data over a prolonged period (weeks rather than hours). Despite the small scale of the trial, it still proved difficult to collect sufficient data in order to provide a valid comparison between users. Due to this limited set of data, analysis has focussed on the 4 main users who provided the largest profiled data sets in order to best illustrate the trends observed.

3. STATISTICAL ANALYSIS

Following the pre-processing described in the previous section, the experimental data for each user was then processed off-line to calculate the mean and standard deviation values for each unique digraph pair. In the event that any digraph pair had a standard deviation greater than its mean value, the digraph samples were sorted and the top/bottom 10% were then removed with subsequent re-calculation of the mean and standard deviation values – this was only attempted where at least ten samples were available for the digraph pair. The reason for this additional step was to remove digraph samples where the latencies would have an adverse affect on the standard deviation (i.e. the distribution of samples was tightened).

Once a set of digraph pairs was produced (with corresponding mean/standard deviation digraph latency values), the user's profile was further constrained by filtering out digraph pairs where the sample count fell below a nominal threshold value. Our experiments fixed this value at fifty samples; however, the software used for analysis allows a variable threshold that will be investigated further in the future work described in a later section. A summary of the profiles generated by this method is shown in *Table 1*.

Table 1: Summary of user profile statistics

User	Unique Digraph Pairs	Filtered Digraph Pairs	Average Typing Speed
User A	466	122	151ms
User B	405	51	145ms
User C	412	89	206ms
User D	461	127	162ms

Once a user profile was generated, the profile was evaluated by comparison with the users' raw keystroke data. This allowed the test profile to be evaluated using the users' own data (to test the False Rejection Rate – FRR) and against other users' keystroke data (to test the False Acceptance Rate – FAR).

As there is likely to be significant variation in a users' own session data, a compensatory factor was applied to the standard deviation that could be varied in a "live" environment according to the security needs of the organisation. This factor allowed the number of standard deviations from the mean to be adjusted. For the purposes of this experiment, four weightings were considered, namely 0.5, 1, 1.5 and 2. This produced an acceptable digraph range:

digraph range = mean ± (standard deviation * weighting factor)

When viewing the preliminary results (*Figure 1*), if we consider the four users A, B, C and D and follow the vertical columns of data, we can see a clear peak for each users data when compared with their own profile. This is most noticeable for user C where a significant peak is observed (50% of all digraphs accepted) compared with 35% when user B's digraph data was tested against the same profile.

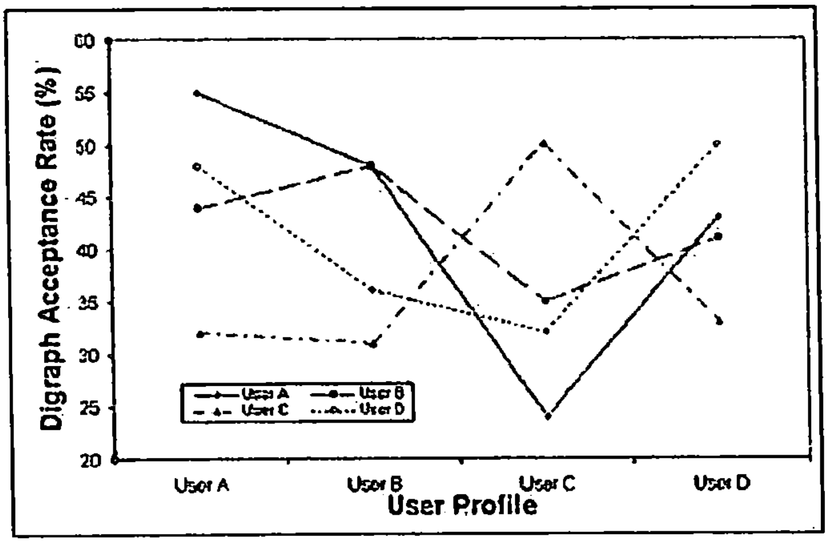


Figure 1: User profile comparisons

Although there was a clear correlation between user C's profile and data, if we consider user A, there was a high FAR for data from users D and B (impostors) when compared with user A's profile. We can also see that in user B's profile the impostor "user A" achieved the same acceptance rate (48%). It is clear from these results that an additional measure of acceptance/rejection is required. To further test the FAR/FRR of the test system, the analysis software monitored the number of consecutively rejected digraph pairs – representing the highest alert level of the system (*Figure 2*).

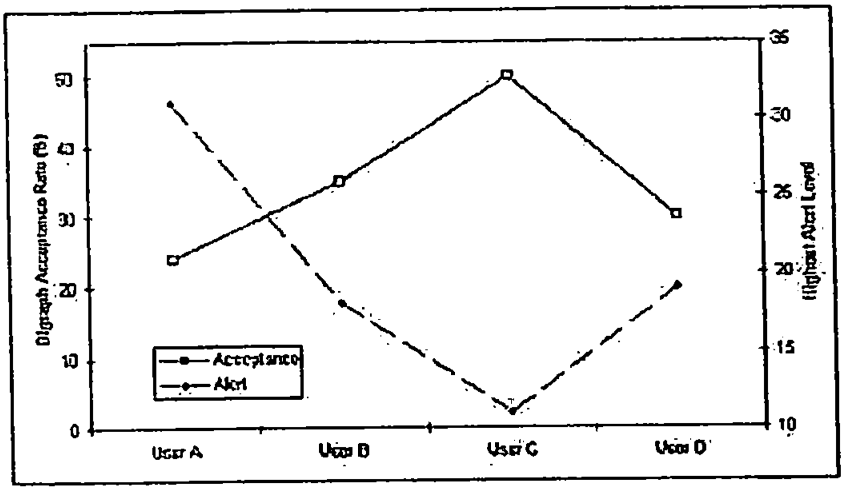


Figure 2: Single user profile comparison

When considering (*Figure 2*) we can identify two distinct trends. Firstly, the top line plots the digraph acceptance rate for all user data sets against user C's profile. Here we can see a clear peak correlating to user C's own data and corresponding reductions in the acceptance rates for the other users' data. Secondly, the lower line indicates the highest alert level detected by the analysis software. This is simply a record of the highest count of consecutively rejected digraph times (excluding non-profiled digraph pairs). Again, we can see a correlation between user C's own data when compared with their profile and corresponding increases in the alert level as impostor data sets are compared with the target profile.

4. DATA MINING ANALYSIS

The methodology described in the previous sections, using traditional statistical approaches, requires a significant level of manual intervention in the data analysis stages. Further, it is time consuming when considering the amount of data generated from a single session or multiple sessions and the number of users on a system. From this we can determine there is a need to automate some of the data analysis pre-processing stages. These stages offer the opportunity to investigate Data Mining (DM) methodology and algorithms, a previously untried approach in this field, in order to eliminate the manual approaches adopted and also to compare the FAR/FRR percentage accuracy with the statistical approach. Data Mining can be described as a collection of techniques and methodologies used to explore vast amounts of data in order to find potentially useful, ultimately understandable patterns [FAYY 96] and to discover relationships. The methodology used to analyse the raw keystroke data is derived from the four main activities of DM; selection, pre-processing, data mining and interpretation [FAYY 96]. DM is an iterative and interactive process, involving numerous steps with many decisions being made by the user. Different algorithms are optimised based on the predefined DM task. This involves deciding whether the goals of the DM process are classification, association, or sequential [MICH 94].

For the purpose of this work, the data sets were split into a ratio of 9:1 hence into two parts; a training set and a testing set, which is a commonly used technique known as train and test. The Intelligent Data Analysis (IDA) Data Mining Tool [SING 99] is used to analyse the sample data sets which incorporates algorithms from the fields of Statistical, Machine Learning and Neural Networks. Six algorithms, k-NN, COG, C4.5, CN2, OC1 and RBF were chosen for this investigative work. The algorithm or classifier is subjected initially with the training set and then the classification accuracy is tested using the unseen data set or testing set. The results give an indication of the error rate (or FAR) and the overall classification accuracy of the trained algorithms.

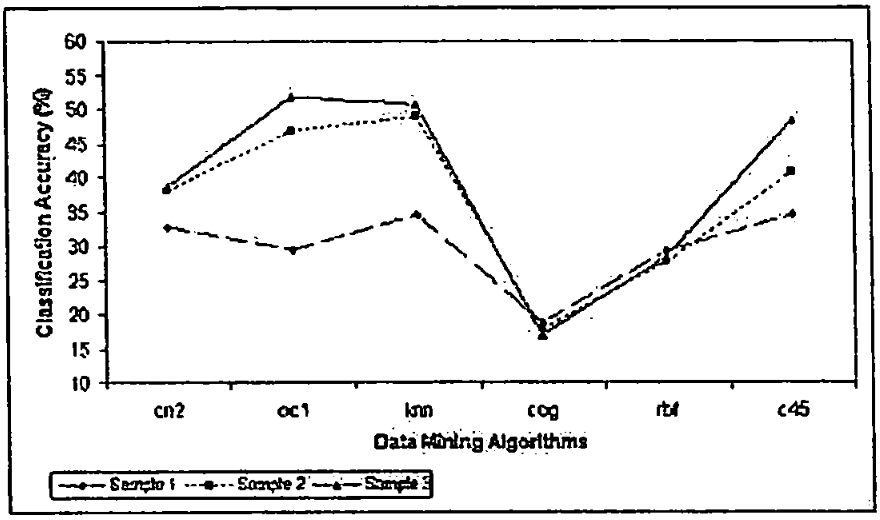


Figure 3: Varying sample sizes with fixed number of classes and attributes

The percentage classification accuracy obtained is encouraging as depicted in (*Figure 3*), which shows that when the sample size is increased, the classification accuracy obtained increases proportionally, except for the

COG a statistical based algorithm and RBF a Neural Network based algorithm. This is important when considering the size of data being analysed and hence eliminates the ad-hoc approaches adopted using traditional statistical methods.

The initial results suggest that Machine Learning (OC1 and C4.5) and Statistical (k-NN) based algorithms are suitable for these types of data sets. Despite the results, more work needs to be carried out in order to correlate the results to a specific or group of algorithm(s), in order to obtain a higher percentage of classification accuracy.

5. CONCLUSIONS

It is clear from the results presented in this paper that there is some potential for continuous user authentication based on keystroke analysis. However, it is also clear that a simple statistical approach does not provide sufficient distinction between users. The DM approach is limited due to the nature of the data gathered and will also require further research. It is proposed that further work will investigate the usefulness of trigraph keystroke combinations (timings for three consecutive keystrokes) and the possible use of word-graph timings (timings for frequently occurring words). Further analysis will be carried out on much larger data sets in order to give a higher statistical reliability -and will also incorporate high-level characteristics (average typing speed and typing error rates) which will provide additional information to the system-characteristic based DM approach being developed in parallel with this research [SING 01]. Other approaches that will be investigated include, consideration of various standard deviation weightings, varying the minimum number of samples for profiled digraphs and varying the inclusion threshold for each sampled digraph. A further possibility for research may be an investigation into a correlation between digraph latencies and the applications in which they were generated (i.e. application specific keystroke profiles).

This paper has presented a series of results from the preliminary statistical analysis of multi-application keystroke data. This has been contrasted with a DM approach to the production of a unique user profile. Whilst the results from this stage of the research are not as encouraging as we had hoped for, they have shown a potential for the use of continuous user authentication. The next phase will concentrate on a combination of techniques to improve the digraph acceptance rate seen in these results.

6. REFERENCES

[COPE 90] Cope J.B.; "Biometric systems of access control"; Electrotechnology; pp71-74; April/May 1990.

[BROW 93] Brown M. & Rogers S.J.; "User identification via keystroke characteristics of typed names using neural networks"; International Journal of Man-Machine Studies; pp999-1014; 1993.

[DOWL 00] Dowland P.S. & Furnell S.M.; "Enhancing Operating System Authentication Techniques"; Proceedings of the International Network Conference 2000 (INC2000); pp253-261; July 2000.

[FAYY 96] Fayyad U.M.; "Data Mining and Knowledge Discovery: making sense out of data"; IEEE Expert; vol. 11; no. 6; pp20-25; 1996.

[FURN 95] Furnell S.M.; "Data security in European healthcare information systems"; PhD Thesis; University of Plymouth, UK; 1995.

[FURN 00] Furnell S.M., Dowland P.S., Illingworth H.M. & Reynolds P.L.; "Authentication and Supervision: A survey of user attitudes"; Computers & Security; vol. 19; no. 6; pp519-539; 2000.

[JOBU 89] Jobusch D.L. & Oldehoeft A.E.; "A survey of password mechanisms: Weaknesses and potential improvements. Part 1"; Computers & Security; p587-603; 1989.

[JOYC 90] Joyce R. & Gupta G.; "Identity Authentication Based on Keystroke Latencies"; Communications of the ACM; vol. 33; no. 2; pp168-176.

[MICH 94] Michie D., Spiegelhalter D.J. & Taylor C.C.; "Machine Learning, Neural and Statistical Classification"; Ellis Horwood; ISBN 0-13-106360-X; pp136-141; 1994.

[MICR 00] Microsoft Corporation; "Monitoring System Events"; 2000; http://msdn.microsoft.com/library/psdk/winbase/hooks_9rg3.htm

[MILL 94] Miller B.; "Vital Signs of Identity"; IEEE Spectrum; February; 1994.

[SHER 92] Sherman R.L.; "Biometrics Futures"; Computers and Security; vol. 11; no. 2; pp128-133; 1992.

[SING 99] Singh H., Burn-Thornton K.E. & Bull P.D.; "Classification of Network State Using Data Mining"; Proceedings of the 4th IEEE MICC & ISCE '99; Malacca, Malaysia; vol. 1; pp183-187; 1999.

[SING 01] Singh H., Furnell S.M., Lines B.L. & Dowland P.S.; "Investigating and Evaluating Behavioural Profiling and Intrusion Detection Using Data Mining"; Proceedings of the International Workshop on Mathematical Methods, Models and Architectures for Computer Networks Security (MMM) 2001; St Petersburg, Russia; 21-23 May 2001.

A Correlation Framework for Continuous User Authentication Using Data Mining

H. Singh[#], S.M. Furnell[*], P.S. Dowland[*], B. Lines* and S. Kaur[+]

[#]School of Computing Science, Middlesex University, White Hart Lane, London, UK
h.singh @ mdx.ac.uk
*Network Research Group, Department of Communication & Electronic Engineering, University of Plymouth, Drake Circus, Plymouth, Devon, UK
info @ network-research-group.org
[+]Institute of Biological Sciences, Faculty of Science, University of Malaya, Kuala Lumpur, Malaysia
sarinder @ um.edu.my

Abstract

The ever-increasing security breaches by both external and internal intruders highlights the lack of security measures in many current systems. Extensive work has been carried out to address this problem, for example by enhancing the initial login stage in order to overcome the security flaws of traditional authentication methods. However, in the event that an unauthorised user compromises a systems initial authentication, the user is in the position to do virtually anything without being further challenged. This has caused interest in the concept of continuous authentication during a user's active session based upon their behaviour characteristics, which inevitably involves the analysis of vast amounts of data. Whereas most reported work in this area uses statistical approaches to model the temporal regularities exhibited by users, this paper presents a series of comparative studies carried out using Data Mining techniques and algorithms. It presents the result of the analysis carried out and discusses a proposed systematic correlation framework for continuous user authentication using the Data Mining methodology adopted in the comparative studies. This paper shows how the correlation framework could be used to automate the analysis of the generated audit data as well as the processes involved in authenticating users in a networked environment.

Keywords

Authentication, Biometric, Data Mining, Behavioural Profiling, Intelligent Data Analysis, Keystroke Analysis.

1. Introduction

The increasing security breaches revealed in recent surveys [1, 2] and security threats reported in the media [3, 4] reaffirms the lack of current security measures in IT systems. While most reported work in this area [5] has focussed on enhancing the initial login stage in order to counteract against unauthorised access, there is still a problem detecting when an intruder has compromised the front line controls. This could pose a serious threat since any subsequent indicator of an intrusion in progress could be quite subtle and may remain hidden to the casual observer. Having passed the frontline controls and having the appropriate access privileges, the intruder may be in the position to do virtually anything without further challenge. This has caused interest in the concept of continuous authentication, which inevitably involves the analysis of vast amounts of data. Although there has been some research [6] in applying the concept of continuous authentication, most of the reported work in this area uses statistical based approaches to model the temporal regularities exhibited by users for the authentication process.

This paper presents some comparative studies carried out using Data Mining (DM) techniques and algorithms. In section 2, we provide details of the comparative studies and results of the analysis carried out using DM algorithms on the generated behavioural data. Although some reported work has been carried out to analyse traffic data using DM [7, 8], none to the knowledge of the authors has investigated the feasibility and effectiveness of learning techniques (e.g. neural network, machine learning, etc.) for the analysis of audit trails. The potential of using DM for the purpose of continuous authentication is further substantiated with a comparative study using a statistical approach for detecting deviation from a user's historical keystroke profile captured under a multi-tasking windowed environment. Since the initial work by Jobusch and Oldehoeft [9], the use of keytstroke analysis has been further investigated using Bayes classifiers [10] and statistical approaches [11] to analyse the keystroke data. It is, therefore, considered that there exists the scope for using DM techniques. The paper then proceeds to describe a proposed correlation framework for continuous user authentication using the DM methodology adopted in the comparative studies. This is considered novel since no work to the author's knowledge have proposed a correlation framework for continuous user authentication using DM.

2. Comparative Study on Behavioural Profiling Using Data Mining

We use DM to extract latent patterns or models of user behaviour from the collected audit trail. This is then reflected in the DM algorithm classifiers (e.g. through rule induction) to recognise deviation, if it occurs, from normal use. This approach is based on the assumption that a user's behaviour has regularity and that using the classifiers this behaviour can be modelled. Using this analogy, anomalous behaviours can then be categorised as a possible unauthorised user or use of that system. The audit trail data analysed was collected from networked computers on a participating local area network (LAN) using an independent agent installed locally in order to audit user interaction with the system. This is based on the assumption that users performing their regular tasks will impose similarly regular demands upon system resources. A number of system parameters were monitored and logged including resource usage and process-related information such as creation, activation and termination. Similar system features have been used in other published work [12], however, this focused on statistical and neural network analysis. A user's behaviour profile can be uniquely identified by: <user name, absolute time, date, hostname, event1,..., eventn >, which is the semantic used for the audit trail where, eventsn denotes the system features being monitored.

The methodology used is derived from the four main activities of DM; selection, pre-processing, data mining and interpretation (Figure 1). The collected audit trail is split into various sample sizes. These subsets form the target data sets, which will undergo the analysis to identify patterns and to test specific hypotheses. The cleaned data, containing both categorical and numerical data, is then subjected to analysis by the DM algorithms. There are a wide variety of DM techniques available, each of which performs more accurately over certain characteristic data sets (e.g. numerical or categorical) and is also relative to the number of variables or attributes and classes. The Intelligent Data Analysis (IDA) Data Mining Tool [13] is used to analyse the sample data sets which incorporate algorithms from the fields of Statistical, Machine Learning and Neural Networks. Six algorithms, k-NN, COG, C4.5, CN2, OC1 and RBF were chosen for this investigation. For the purpose of this work, the data sets were split into ratios of 9:1, 8:2 and 7:3, hence into two parts, which is a commonly used technique known as train and test. The algorithm or classifier is initially subjected to the training set and then the classification accuracy is tested using the unseen data set or testing set. The results give an indication of the error rate (or false positives) and the overall classification accuracy of the trained algorithms.

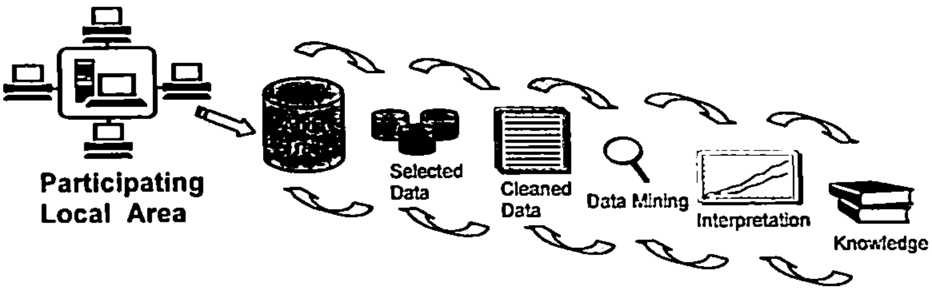


Figure 1. Methodology for Behavioural Profiling

2.1 Discussion of results

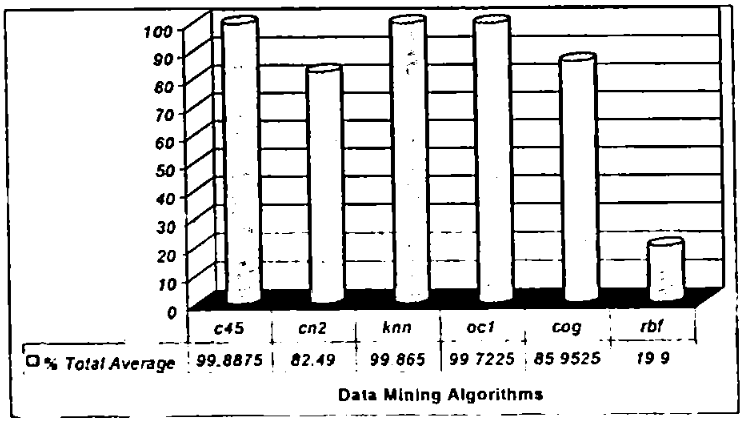


Figure 2. Total percentage average classification accuracy of selected DM algorithms

The initial results obtained from the analysis (Figure 2), suggest that Machine Learning and Statistical-based algorithms are better for these types of data sets. C4.5 and OC1 decision tree based algorithms in particular, out performed the CN2 rule-based and RBF algorithms. The classification accuracy obtained, using k-NN in comparison to C4.5, shows some significance for further investigation despite the slower classification times observed. Amongst the statistical algorithms, k-NN faired better then COG but is slower in comparison to the classification times observed. The classification accuracy obtained overall indicates that RBF classification accuracy is inversely proportional to the sample size. These results support other reported work [14]. In addition to the consistency in classifying the data sets and the overall average classification accuracy, our initial investigations also identified that C4.5 has overall quicker train and test time and outputs explicit rules.

3. Comparative Study on Keystroke Data Analysis Using Data Mining

Keystroke analysis is an example of a biometric that uses inter-keystroke latencies (time between keystrokes) to differentiate between users. While in the previous section the aim was to show the feasibility and effectiveness of DM learning algorithms in building temporal regularities in user behaviour, this section is intended to build upon the findings by comparing against statistical approaches. In order to collect the required data, an independent agent installed locally on the networked computers was used for acquiring keystroke notifications across all applications running within a users' active session. A total of ten users were profiled out of which only 4 users (who provided the largest profiled data sets) were selected. The audit trail generated contained the following attributes: <first character, second character, digraph latency>.

Full details of the statistical method used are detailed in [15]. Therefore only the relevant results are presented here. A summary of the profiles generated by the statistical method is shown in Table 1.

User	Unique Digraph Pairs	Filtered Digraph Pairs	Average Inter-keystroke Time
User A	466	122	151ms
User B	405	51	145ms
User C	412	89	206ms
User D	461	127	162ms

Table 1. Summary of user profile statistics

Once a user profile was generated, the profile was evaluated by comparison with the users' raw keystroke data. This allowed the test profile to be evaluated against the users' own data (to test the False Rejection Rate – FRR) and against other users' keystroke data (to test the False Acceptance Rate – FAR).

3.1 Discussion of results

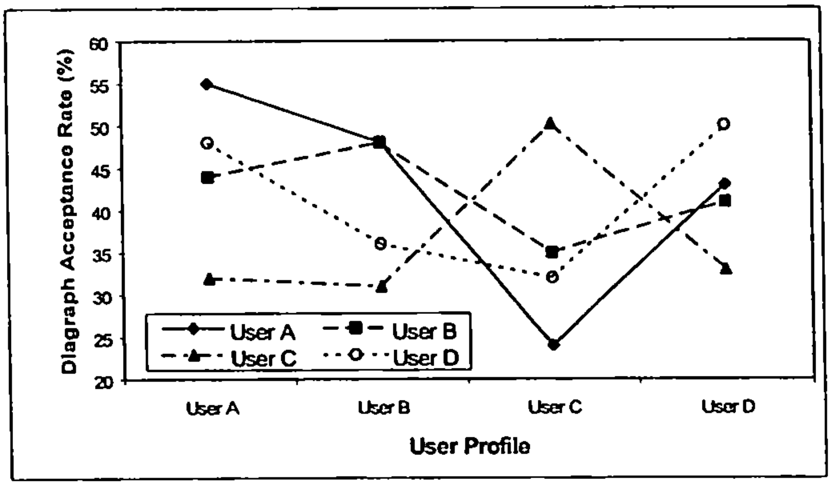


Figure 3. User profile comparisons

When viewing the preliminary results (Figure 3) if we consider the four users A, B, C and D and follow the vertical columns of data, we can see a clear peak for each users data when compared with their own profile. This is most noticeable for user C where a significant peak is observed (50% of all digraphs accepted) compared with 35% when user B's digraph data was tested against the same profile. Although there was a clear correlation between user C's profile and data, if we consider user A, there was a high FAR for data from users D and B (impostors) when compared with user A's profile. We can also see that in user B's profile the impostor "user A" achieved the same acceptance rate (48%).

3.2 Data Mining approach

The methodology used to analyse the raw keystroke data using DM followed a similar principle to that described in section 2. For the purpose of this work, the data sets were split into a ratio of 9:1. The algorithm or classifier is subjected initially to the training set and then the classification accuracy is tested using the unseen data set or testing set. The results give an indication of the error rate (or FAR) and the overall classification accuracy of the trained algorithms. The percentage acceptance rate obtained is encouraging (Figure 4), when considering the acceptance rate achieved for the highest algorithm is 53%. This is in consideration of the time factor involved in comparison to the statistical approach and the amount of domain expertise input to the process, which only resulted in an absolute difference of 2% from the highest percentage acceptance rate (i.e. 55%) obtained in the statistical analysis. Furthermore the acceptance rate obtained increases proportionally (unlike the statistical approach which is restricted in the sample size analysed), except for the COG and RBF algorithms. This is important when considering the size of data being analysed and hence eliminates the ad-hoc approaches adopted using traditional statistical methods. The initial results suggest that Machine Learning (OC1 and C4.5) and Statistical (k-NN) based algorithms are suitable for these types of data sets. Despite the results, more work needs to be carried out in order to correlate the results to a specific or group of algorithm(s), in order to obtain a higher percentage of classification accuracy. Nevertheless it is clear from the comparative study carried out that DM algorithms have the potential to automate the process of discovering the temporal regularities from the data sets, which would otherwise rely heavily upon intuition and experience in building this model using other approaches (e.g. statistical approach). Furthermore the methodology and algorithms provides the foundation, which could be integrated into a correlation framework as presented in the following section.

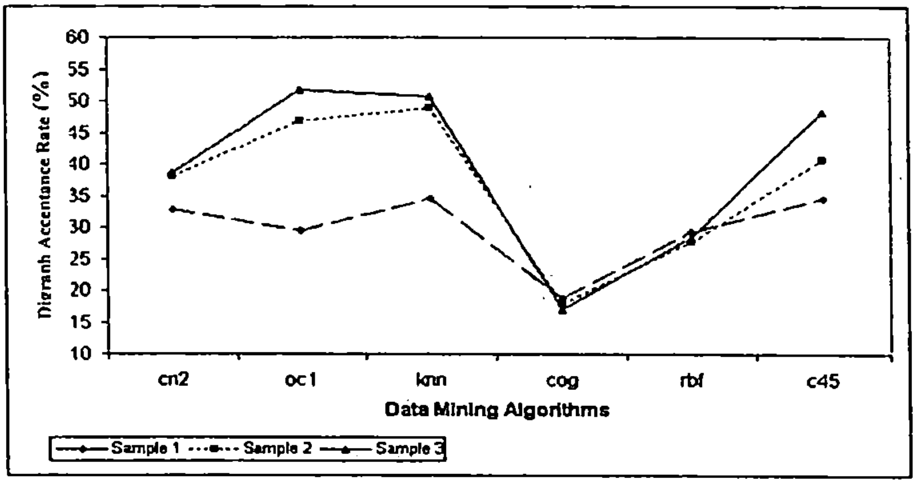


Figure 4. Varying sample sizes with fixed number of classes and attributes

4. Correlation Framework

The approaches investigated for user behaviour and process activity profiling could be used as the basis to provide user identification and authentication. Our initial results show the potential of developing and integrating the DM techniques investigated into a correlation framework, which could be integrated into an operating system user authentication scheme (Figure 5). The concepts behind the correlation framework are in some ways similar to the principles of inductive reasoning [16] where the goal is to arrive at a decision (i.e. if deviation is occurring) from a limited set of information (i.e. behaviour indicative data) available due to the inherent problem of gleaning specific information from audit trails. The key aspects of this design are defined in the sections that follow.

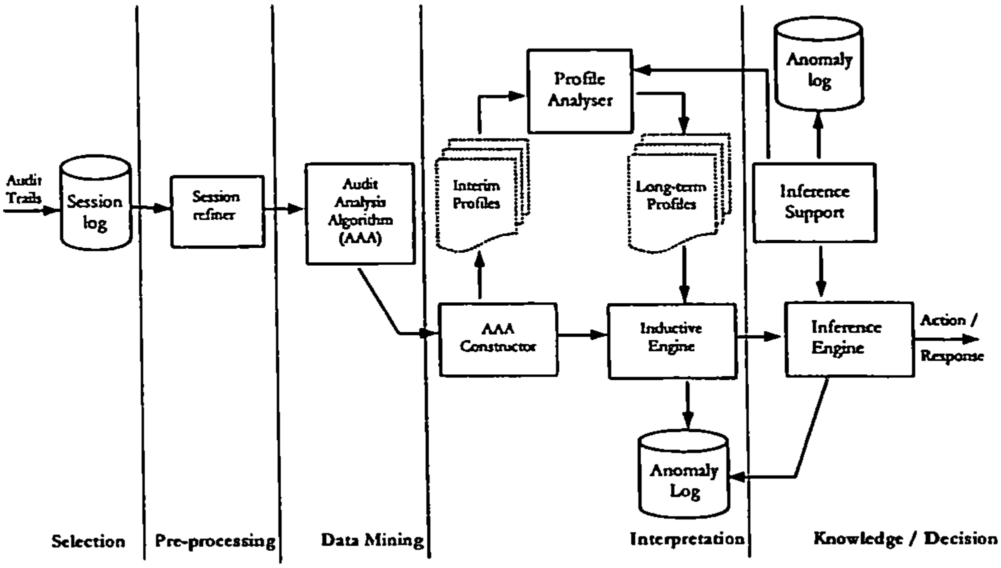


Figure 5. Correlation framework for continuous user authentication using Data Mining

4.1 Session Log

The Session Log would provide a temporary storage to the generated audit trails. The data stored would be restricted to only relevant data pertaining to the behavioural data (e.g. resource usage data, keystroke data, etc.), although it can potentially be a source for a permanent record to provide evidential support should the need transpire. This would enable a reduced access time for the Audit Processing module to select the relevant features for analysis in the later stage.

4.2 Session Refiner

The Session Refiner prepares the target data set prior to analysis. Typically this may involve converting the data into an acceptable format, annotating the beginning of each message (e.g. time stamp, absolute time, etc.) or may involve processes to quantify continuous features, e.g. to generate the audit trail semantic. These stages can be used to constrain the search space and make patterns or relationships in the audit trails more visible in the later stages of the correlation process.

4.3 Audit Analysis Algorithms

The Audit Analysis Algorithms (AAA) incorporate the DM algorithm(s) that will be used to identify system features, patterns and latent trends for classifying user behaviour. Data features such as correlation between adjacent or frequent sequential patterns of user behaviour will be analysed. The information gleaned from using these algorithms will be used to identify the temporal regularities of a user's behaviour, which will be reflected in the user's profiles in the latter stages.

4.4 AAA Constructor

The AAA Constructor would refine the inferred association rules or classification rules from the AAA engine. The various types of patterns exhibited in the data would be cleaned (i.e. removing redundant data), combined, and transformed into an understandable syntax. This will later be stored in the Interim Profile, which would provide a temporary repository.

4.5 Profiler Analyser

The concepts behind the Profiler Analyer are adapted from the IMS Profile Refiner [17] and would have a similar functionality in the proposed correlation framework. Similarly the audit trails generated would be optimised as input to the Inductive Engine to detect deviation from normal use and as a source for updating user profiles, which will inevitably change over time. Depending on how often similar patterns are exhibited by user(s), these changes will be reflected in the Long-term Profiles repository.

4.6 Inductive Engine

The Inductive Engine would enable the detection of any deviation occurring on the system. The Long-term Profiles of users would be compared against the generated audit trails to detect for anomalies. Anomalies detected would be stored in the Anomaly Log, which would provide the basis for detected deviations to be further analysed in order to reduce the probability of false positives prior to reporting the conclusion inferred through the Inductive Engine.

4.7 Inference Engine

The Inference Engine would be used to identify recurring valid behavioural patterns that are being flagged as anomalies. These would be filtered out in order to reduce the potential of high false positive errors by correlating previously known anomalies logged in the Anomaly Log to the current active anomaly detected and from known information input through the Inference Support component.

4.8 Inference Support

The Inference Support would be used to improve the inferred facts from other sources. System Administrators or Security Officers could input this information (e.g. public holidays, staff on sick leave, etc), which would otherwise take a longer time, through normal circumstances, to infer and thus detect anomalies occurring. Furthermore it would be used to provide input to the Profile Analyser where deemed necessary for instance, to disable profiling when a user is away as a countermeasure against the possibility of an unauthorised user introducing new temporal behaviour which would effect the legitimate user's profile. It would also enable any modifications (i.e. removing or adding anomalies) or maintenance required in the log files of the Anomaly Log.

5. Discussion and Conclusion

While there is a tendency to equate complex statistical analysis with correlation or the detection mechanism, this paper has presented the results to date from the comparative studies carried out using DM. The methodology used and the classification accuracy obtained in this initial investigative work suggests that DM techniques could be integrated into a correlation framework for continuous authentication. The high classification accuracy obtained and fast response time exhibited in classifying the user behaviour by some of the DM algorithms, when considering the vast amount of audit trail analysed, further demonstrates the potential of applying DM techniques within a real-time application. Whereas previous work in this area has been focussed on developing the DM algorithms for domain specific problems, no work to date has integrated these techniques into a correlation framework. The methodology developed in analysing the generated audit trails, which is advocated by the proposed correlation framework, has the potential to provide an important contribution to the development of a correlation framework for the purpose of continuous user authentication.

6. References

1. Computer Security Institue, "2003 CSI/FBI Computer Crime and Security Survey", http://www.gocsi.com/
2. The National Computing Centre, "The Business Information Security Survey (BISS 2002)", http://www.ncc.co.uk/ncc/biss2000.pdf
3. B. Betts, "Digital Forensic:crime scene", March 2000, http://www.infosecuritymag.com/articles/march00/cover.shtml
4. M. Ward, "Web Warning Centre in Net Attack", BBC News, 24th May 2001, http://news.bbc.co.uk/hi/english/sci/tech/newsid_1348000/1348820.stm
5. B. Miller; "Vital Signs of Identity", IEEE Spectrum, February, 1994
6. A. Seleznyov, O. Mazhelis and S. Puuronen, "Anomaly Intrusion Detection System Based on Online User Recognition", Proceedings of the Third International Network Conference (INC 2002)", Plymouth, UK, 16 – 18 July 2002
7. C. Warrender, S. Forrest and B. Pearlmutter, "Detecting Intrusion Using Calls: alternative data models", Symposium on Security and Privacy, 1999
8. W. Lee and S. Stolfo, "A Framework for Constructing Features and Models for Intrusion Detection Systems", ACM Transactions on Information and System Security, vol. 3, no. 4
9. D.L. Jobusch and A.E. Oldehoeft; "A Survey of Password Mechanisms: weaknesses and potential improvements. Part 1", Computers & Security, pp. 587-603, 1989
10. S. Bleha, C. Silvinsky and B. Hussein, "Computer-Access Security Systems Using Keystroke Dynamics", Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 12, 1990
11. R. Joyce and G. Gupta, "Identity Authentication Based on Keystroke Latencies", Communications of ACM, vol. 33, February 1990
12. T.F. Lunt, "IDES: an intelligent system for detecting intruders", Proceedings of the Computer Security, Threat and Countermeasures Symposium, Rome, Italy, November 1990
13. H. Singh, K.E. Burn-Thornton and P.D. Bull, "Classification of Network State Using Data Mining", Proceedings of the 4th IEEE MICC & ISCE '99, Malacca, Malaysia, vol. 1, pp. 183-187
14. D. Michie, D.J. Spiegelhalter and C.C. Taylor, "Machine Learning, Neural and Statistical Classification", ELLIS HORWOOD-1994, ISBN 0-13-106360-X, pp. 136-141
15. P.S. Dowland, H. Singh and S.M. Furnell, "A Preliminary Investigation of User Authentication Using Continuous Keystroke Analysis", Proceedings of the IFIP 8th Annual Working Conference on Information Security Management & Small Systems Security, Las Vegas, 27-28 September, 2001
16. J. Durkin, "Expert Systems Design and Development", PRENTICE HALL-1994, ISBN 0-02-330970-9, pp. 90-130
17. S.M. Furnell, "Data security in European Healthcare Information Systems" PhD Thesis, University of Plymouth, 1995

Advanced Authentication and Intrusion Detection Technologies

Paul Dowland, Dr Steven Furnell, George Magklaras, Maria Papadaki, Prof Paul Reynolds, Philip Rodwell, Harjit Singh
Network Research Group, Department of Communication and Electronic Engineering, University of Plymouth, UK

Abstract

Security is a vital consideration in the age of modern networks and Internet-based communications, and represents an essential underpinning of emerging applications such as electronic commerce. Within this domain, the ability to ensure the authorised and correct use of systems is an area of significant challenge. The research to be presented is centred around the Intrusion Monitoring System, a conceptual architecture for real-time user authentication and supervision, which has been defined by an earlier project within the Network Research Group. The current research encompasses advanced authentication technologies, based upon biometric techniques and user behaviour profiling. These approaches improve considerably upon the traditional user name and password combination, which has been proven to be weak and susceptible to compromise. While enhanced authentication will combat external impostors and internal masqueraders, further research addresses methods of identifying system misuse originating from authorised users, whom independent surveys have established account for around 80% of computer abuse incidents. Another important consideration relates to methods of responding to suspected intrusions in a manner that will trap genuine impostors and misfeasors, without unduly disrupting legitimate user activity in cases where a false classification has occurred. The research considers the application of these authentication and intrusion detection approaches within both traditional desktop environments and third generation mobile networks.

The Intrusion Monitoring System architecture

The Intrusion Monitoring System (IMS) is the focus of security research in the Network Research Group. IMS is an architecture for intrusion monitoring and activity supervision, based around the concept of a centralised host handling the monitoring of a number of networked client systems. Intrusion detection is the system is based upon the comparison of current user activity against both historical profiles of 'normal' behaviour for legitimate users and intrusion specifications of recognised attack patterns. The architecture is comprised of a number of functional modules, addressing data collection and response on the client side and data analysis and recording at the host (as illustrated in the figure below).

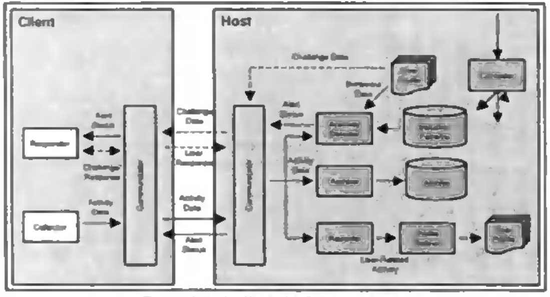


Figure 1: Intrusion Monitoring System architecture

Related Research Projects

The current IMS related research projects are listed below, a number of these involve collaboration and/or sponsorship from Orange Personal Communications Services.

1. User authentication and supervision in networked systems
This project concerns the investigation and evaluation of composite authentication techniques. The study recognises that a variety of authentication techniques are available which, when used in isolation, have known error rates in terms of false rejection and false acceptance. The research is focused upon the specification, implementation and evaluation of a composite authentication approach, in which a range of technologies are available and can be applied intelligently by the system as appropriate to the active user and their current task.

2. Behavioural profiling and intrusion detection systems using data mining
This project seeks to develop profiles of user behaviour by applying intelligent analysis techniques to system data collected in real-time during Windows NT sessions. The profile would represent a model of the legitimate users normal behaviour and could subsequently be utilised in a real-time supervision context to ensure that the activities of the user match those expected of the claimed identity. As such, the technique offers the potential to identify impostors and misusers within the system. This project contributes to the profiling aspect of the IMS architecture.

3. Generic approaches for intrusion specification and misuse detection
This project seeks to design and develop a generic intrusion specification language, which may then be used to specify intrusion characteristics in detection systems such as IMS. This work leads in to specific consideration of how to detect misfeasor attacks – that is, misuse of the system by a legitimate user. In terms of the IMS architecture, this project will contribute to the mechanisms utilised by the Anomaly Detector module.

4. Classifying and responding to network intrusions
This project seek to determine a comprehensive taxonomy of system-detectable intrusions and misuse, leading to the consideration of how the IMS system should respond to them. The work will involve design and practical evaluation of alternative response strategies, assessing factors such as the effectiveness against the nominated class(es) of intrusion and any negative effects that the response could have upon a legitimate user in a false rejection scenario.

5. Non-intrusive security for third generation mobile systems
This project highlights the need for improved methods of user authentication in future mobile systems such as the Universal Mobile Telecommunications System (UMTS). The work is focused upon an investigation of user-to-terminal and user-to-network forms of authentication for use in future mobile networks and devices.

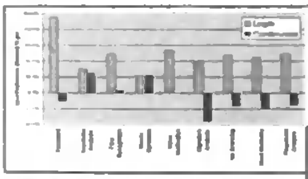


Figure 2: User Preference of Authentication and Supervision Methods (175 respondents)

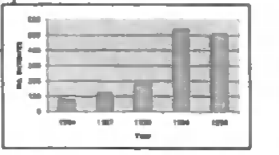


Figure 3: Reported Computer Crime Incidents (source: Audit Commission)

Authentication & Intrusion Detection Approaches

IDS techniques are based on the assumption that an intruder's behaviour will be different from that of a legitimate user. In order to detect this deviation from normal user behaviour, IDSs collect audit data such as system resource usage. Providing this continuous monitoring involves processing and analysing vast amount of audit data. Hence relying on human expertise is time consuming, knowledge intensive and infeasible past a certain volume of data. Therefore intelligent data analysis techniques are required to automate some of this process. The research is investigating techniques to automate some of the data analysis using Data Mining (DM) techniques and methodologies (figure 4 illustrates initial results).

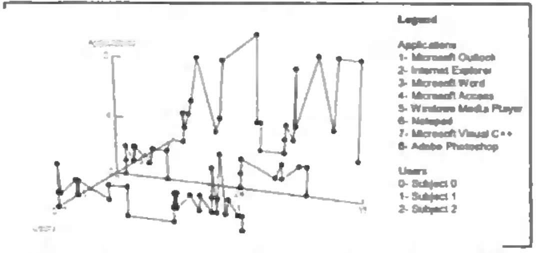


Figure 4: Graphical representation of applications run by users (IMS - Behavioural Profiling)

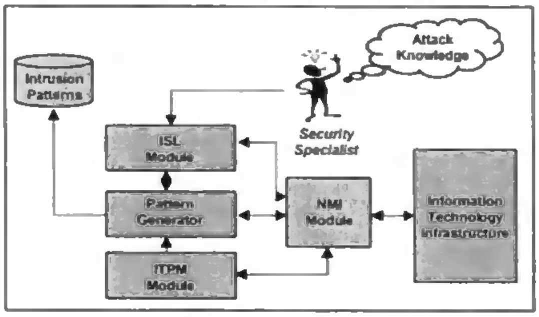


Figure 5: Functional Modules of an Intrusion Specification Language

ISL module The Intrusion Specification Language module, responsible for describing intrusions in a standardized, system independent manner. Used by the security specialist/architect.

Pattern generator module It converts the ISL descriptions into system-specific patterns. It also performs optimised pattern matching functions that are essential for real-time intrusion detection.

ITPM module The Insider Threat Prediction Model tries to sense the level of sophistication of a legitimate user. It can then estimate the probability that a particular user will misuse the infrastructure. This is an experimental/new approach of tackling the insider IT misuse problem.

NMI module The Network Management Integration module is responsible for utilising network management protocols in order to collect information and co-ordinate selected IDS responses from a variety of IT infrastructure components.

Information Technology Infrastructure The set of computer hardware, software and telecommunication components that perform a useful function inside an organisation.

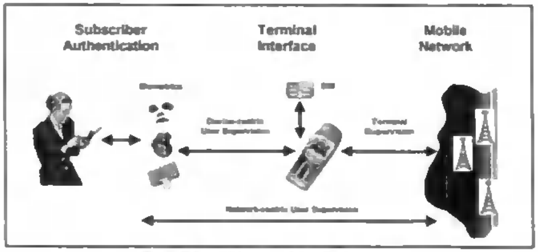


Figure 6: Supervision in a Mobile Environment

Summary

The techniques under investigation represent a considerable departure from traditional methods of authentication and access control, and aim to provide an added level of protection for IT systems. Modern society is increasingly dependent upon IT infrastructures. At the same time, surveys from bodies such as the Audit Commission and the FBI are reporting increased levels of computer crime and abuse (originating from both outside the organisation and from within). In view of these factors, the additional safeguards provided by advanced security techniques will become ever more necessary. As the research has identified, the techniques are relevant to both traditional networked PC environments, as well as other scenarios such as third generation mobile systems.

Dowland, P.S., Furnell, S.M., Illingworth, H.M. and Reynolds, P.L. 1999 "Computer Crime and Abuse: A Survey of Public Attitudes and Awareness", *Computers & Security*, vol. 18, no. 8, pp715-726.

Dowland, P. and Furnell, S 2000. "Enhancing Operating System Authentication Techniques", *Proceedings of the Second International Network Conference (INC 2000)*, Plymouth, UK, 3-6 July 2000, pp253-261

Furnell, S.M., Illingworth, H.M., Katsikas, S.K., Reynolds, P.L. and P.W.Sanders. 1997. "A comprehensive authentication and supervision architecture for networked multimedia systems", *Proceedings of IFIP CMS '97*, Athens, Greece, 22-23 September 1997, pp227-238

Furnell, S.M. and Dowland, P.S. 2000. "A conceptual architecture for real-time intrusion monitoring", *Information Management & Computer Security*, vol. 8, no. 2, pp65-74

Furnell, S.M., Dowland, P.S., Illingworth, H.M. and P.L.Reynolds. 2000. "Authentication and supervision: A survey of user attitudes", *Computers & Security*, vol. 19, no. 6, pp529-539

Papadaki, M. 2000. *A Taxonomy of I.T. System Intrusions*. M.Sc. Thesis, University of Plymouth, Plymouth, UK.

Rodwell, P.M., Furnell, S.M. and Reynolds, P.L. 2000. "Non-intrusive security requirements for third generation mobile systems", *Proceedings of PG Net 2000 – 1st Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting*, Liverpool, UK, 19-20 June 2000, pp7-12.

http://ted.see.plym.ac.uk/nrg