Towards an Insider Threat Prediction Specification Language

G.B.Magklaras[1], S.M.Furnell[1] and P.J.Brooke[2]

[1] Network Research Group, School of Computing, Communications & Electronics, University of Plymouth, Plymouth, United Kingdom
[2] School of Computing, University of Teesside, Middlesbrough, United Kingdom

nrg@plymouth.ac.uk

Contact details

Corresponding Author

Dr Steven M. Furnell

Network Research Group
School of Computing, Communications and Electronics
University of Plymouth
Plymouth
United Kingdom

Tel : +44 1752 233521
Fax : +44 1752 233520

e-mail : sfurnell@plymouth.ac.uk

Other authors

Mr George B. Magklaras

Network Research Group
Room A304, Portland Square Building
School of Computing, Communications and Electronics
University of Plymouth
Plymouth
United Kingdom

e-mail : georgios@ulrik.uio.no

Dr Phil J. Brooke

School of Computing
University of Teesside
Middlesbrough
Tees Valley
United Kingdom

e-mail : P.J.Brooke@tees.ac.uk

Towards an Insider Threat Prediction Specification Language

Purpose

This concept paper presents the process of constructing a language tailored to describing insider threat incidents, for the purposes of mitigating threats originating from legitimate users in an IT infrastructure.

Design/Methodology/Approach

Various information security surveys indicate that misuse by legitimate (insider) users has serious implications for the health of IT environments. A brief discussion of survey data and insider threat concepts is followed by an overview of existing research efforts to mitigate this particular problem. None of the existing insider threat mitigation frameworks provide facilities for systematically describing the elements of misuse incidents, and thus all threat mitigation frameworks could benefit from the existence of a domain specific language for describing legitimate user actions. The paper presents a language development methodology which centres upon ways to abstract the insider threat domain and approaches to encode the abstracted information into language semantics.

Research limitations/implications

Due to lack of suitable insider case repositories, and the fact that most insider misuse frameworks have not been extensively implemented in practice, the aforementioned language construction methodology is based upon observed information security survey

trends and the study of existing insider threat and intrusion specification frameworks. The development of a domain specific language goes through various stages of refinement that might eventually contradict these preliminary findings.

Practical implications

This paper summarizes the picture of the insider threat in IT infrastructures and provides a useful reference for insider threat modeling researchers by indicating ways to abstract insider threats. The problems of constructing insider threat signatures and utilizing them in insider threat models are also discussed.

Keywords: Intrusion Detection, Insider Threat, Insider Misuse, Domain Specific Language, Intrusion Specification

Introduction

The Information Security world often focuses on analyzing and counteracting threats of external origin. However, the problem of insider IT misuse is also an existing headache for the health of IT infrastructures. Surveys published by the British Department of Trade and Industry (DTI) and PriceWaterhouseCoopers (PWC, 2004) as well as the San Francisco-based Computer Security Institute (Richardson, 2003) are good sources for getting a qualitative and quantitative feeling of computer security incidents. Relevant information derived from these surveys is presented in the following section.

Amongst the various research and development issues related to the process of mitigating the problem of internal threats lies the ability to describe the actions that constitute the elements of the threat in a consistent manner. This goal can be achieved by constructing a suitable Insider Threat Prediction Specification Language (ITPSL), in order to facilitate ways of standardizing the description of Insider IT Misuse incidents and thus aid tools designed for detecting and preventing them.

After introducing the problem by quoting incident statistics, basic terminology and discussing some of the Insider IT misuse mitigation frameworks, this paper focuses on the methodology required to construct the language itself, by examining ways to classify insider incidents, as well as the suitability of pre-existing intrusion specification schemes for insider threat specification.

The Insider IT Misuse Threat

An ‘insider’ is a person that has been legitimately given the capability of accessing one or many components of the IT infrastructure, by interacting with one or more authentication mechanisms (plain text password, PKI, biometric or smart card token). The word ‘legitimately’ is a key term, as it emphasizes the main difference between an insider and an external cracker. An insider should always be able to have at least a point of entry in one or more computer systems. The implications of having such a point of entry is that an

insider does not usually need to consume as much time and effort to obtain additional privileges as an external cracker does, in order to exploit IT infrastructure vulnerabilities and mount an attack. It also means that an insider is less likely to get caught by implemented security measures because of the level of trust that she enjoys.

The other side of the insider IT misuse problem relates to what can be considered as misuse activity. Although the great majority of the people are familiar with the generic meaning of the word 'misuse', when we try to map it to an insider IT context, there is a need to clarify certain issues. Insider IT misuse can be a very subjective term. In fact, one of the most challenging tasks is to draw a clear line that separates an IT misuser from a person that uses the available resources in an acceptable way and for an approved purpose. The words 'acceptable' and 'approved' imply the presence of rules that qualify (or quantify) conditions of allowable usage for the resources concerned. These rules are often embodied within an IT usage policy. Part of this organisation-wide policy is the information security policy, defined as the '*set of laws, rules, practices, norms and fashions that regulate how an organisation manages, protects, and distributes the sensitive information and that regulates how an organisation protects system services*' (Caelli et al. 1991).

After defining the terms ‘insider’ and ‘misuse’, we also need to discuss the context of the term ‘threat’. Pfleeger et al. (2003) define the term threat in an IT infrastructure context as “a set of circumstances that has the potential to cause loss or harm”. As a result, in legitimate user context, these circumstances might involve intentional IT misuse

activities such as targeted information theft, introducing or accessing inappropriate material, and accidental misuse (e.g. unintentional information leak). In addition, there is also potential for flaws in the design and implementation of the computer system, which could render it susceptible to insider misuse. Such flaws may include improper filesystem permissions or relaxed information security policies, and in conventional information security parlance these are termed vulnerabilities.

The widespread manifestation of insider IT misuse incidents is an existing problem for the health of IT infrastructures. The DTI/PWC 2004 survey (PWC, 2004) mentions that that Insider Misuse has doubled since the year 2002, mainly driven by the increased adoption of World Wide Web and Internet related technologies. The same survey shows that the gap between insider and outsider incidents is smaller for respondents of medium and large scale organizations (Figure 1). The CSI/FBI survey (Richardson, 2003) also indicates closer margins between the occurrence of internal and external incidents (Figure 2).

Take in Figure 1

Take in Figure 2

Magklaras and Furnell (2004) discuss in greater detail the manifestation of the Insider IT Misuse based on the results of a small scale survey.. The survey queried 50 IT professionals of various specialties, including respondents in system administrator, IT

security and management roles. Despite the small number of respondents, their majority had a technical background on various aspects of the insider IT misuse, providing an insight on notable trends of the problem and thus establishing a profile of a misuser.

In particular, some important highlights of Magklaras and Furnell (2004) include the fact that the three most frequent types of IT misuse for the respondents of the survey were the downloading of pornographic material, the abuse of email resources, and the theft or malicious alteration of data. In direct comparison, the DTI/PWC 2004 survey (PWC, 2004) highlights the incidents of web browsing misuse, misuse of email, and unauthorized access to systems or data as the major system misuse categories (Figure 3).

In addition, all of the professionals surveyed by Magklaras and Furnell (2004) indicated some preference towards the existence of certain pre-employment security checks for prospective employees. The DTI/PWC survey (PWC, 2004) indicated that the majority (66%) of the respondents usually perform some sort of general employee background check during the recruiting stage, checking. This survey also comments that the absence of these security checks from company procedures is clearly a serious omission.

Finally, the profile of an insider threat was also refined by indicating that sophisticated users are more likely to misuse an IT infrastructure than less IT-literate users (Magklaras and Furnell, 2004).

The frequency of occurrence is not the only indicator of the impact of insider incidents. There are also substantial financial costs attributed to legitimate user actions. However, due to a combination of factors (such as the smaller percentage of respondents willing to state financial losses in information security surveys and the way some of the surveys associate stated losses to incident types) the process of safely estimating true insider costs is rendered infeasible.

Take in Figure 3

The various survey results combine to suggest that internal incidents are here to stay and their mitigation should be a priority issue for IT professionals.

Insider IT misuse mitigation frameworks

There are numerous research and development efforts that attempt to address the problem of legitimate user misuse. All of them focus on predicting or sensing insider threats. The process of predicting a particular set of events in order to prevent their occurrence and provide a better understanding of their underlying mechanisms does not represent a new methodology in the field of science. The utilisation of game theory in financial forecasting (Gibbons, 1992), in order to predict the value of shares in the stock exchange market and the processing of seismic data for oil discovery purposes (Helbig, 1993) are notable examples of methodologies that already serve our world and used on a daily basis by analysts, as value-added tools that help their activities.

The insider IT misuse mitigation framework suggested by Wood (2000) is one of the earliest examples of qualifying a set of metrics addressing the insider threat. The framework suggests that a malicious insider can be qualified in terms of distinct attributes such as the amount of access she has on some part or all parts of the IT infrastructure (physical and logical access in terms of privileges), her level of familiarity with the internal workings of the target systems, her motives as well as the skills, tactics and processes she uses to mount an attack.

A more recent research effort by Schultz (2002) presents a preliminary framework for understanding and predicting insider attacks by providing a combination of behavioural and system usage related metrics. The paper mentions the detection of system usage patterns that may act as “signatures” of a legitimate user or certain indicators of an attack preparation (“deliberate markers” and “preparatory behaviour”). Sequences of actions that might not be detected in individual systems, but which could certainly indicate misuse when considered against multiple systems are discussed. There is also a discussion of aspects of a legitimate user’s personality that could serve as threat indicators. In particular, on-line (e-mail, IRC or other forms of computerised human-to-human communication) verbal behaviour with signs of aggression, dominance towards particular people might serve as a good prognosis factor of certain attacks (“verbal behaviour”).

Magklaras and Furnell (2002) discuss an alternative framework for insider IT misuse mitigation, focusing on insider threat metrics that could be collected at system level. The Insider Threat Prediction Model (ITPM) is a three-level hierarchy of mathematical functions evaluated in a bottom-up approach. Each of the threat component functions models particular aspects of insider attributes and behavior. The end result is an integer, the Evaluated Potential Threat (EPT), which classifies the level of potential threat a particular user represents for the IT infrastructure and could be used as an indicator of whether the user poses a threat or not.

While each of the aforementioned frameworks has is its own theoretical advantages and disadvantages, they also have something important in common. Whether one places emphasis on verbal behaviour, gauging of user knowledge, or the observation of user action sequences in order to sense or predict insider threat, all of these metrics could benefit from a standardized way of describing them efficiently, in order to make insider IT misuse threat signatures. This is the subject of the following sections.

Insider Threat Prediction Specification Language: The need and its construction methodology

Information security surveys and mass media might report accurately the outline of an insider misuse case. They do not provide a complete picture about the exact conditions under which the incident occurs, nor they always reveal in detail the sequence of user

actions. As an example, one should consider the high-profile case of Robert Hanssen (CNN, 2002). A 56-year old trusted FBI veteran, Hanssen abused his trusted access to the Automated Case Support System that contained classified information about ongoing investigations and handed critical information to Russian agencies. In return, he was receiving large sums of money, inflicting a great deal of damage upon the prestigious image of the Federal Bureau of Investigation and the national security of his country.

The motives and the outline of Hanssen's methodology were covered by the mass media. Some of the details related to his data hiding techniques were also mentioned in computing sites (Slashdot, 2001). However, even the latter details are not enough for someone to re-construct the case in a laboratory for the purposes of experimenting and developing insider threat prediction techniques. If one is not a member of the forensic specialist team that handles an insider misuse case, he will be able to only speculate about the actions and the attack path a malicious insider had followed. This creates a lack of suitable case repositories noted by (NSTISSAM, 1999), and is one of the limiting factors in the field of insider IT misuse mitigation research.

The establishment of a world-wide insider case repository would be of great aid to researchers. However, apart from the coordination, the building of such a repository would require a way to unambiguously describe the insider misuse actions in a standard manner. This paves the way for the shaping of a Domain Specific Language (DSL), a semantic mechanism tailored specifically for describing the details of a particular task.

The main goal is the usage of appropriate semantics to reduce the effort required to reference and manipulate elements of that particular domain.

DSL schemata have been employed successfully in a number of different areas. Consel (2004) discusses the range of applications that have employed a DSL which includes device driver construction, active networking and operating system process scheduling. This list is by no means exhaustive and it really concerns all domains that consist of software entities that have enough common elements to be considered as a whole. A DSL is really a framework that offers the ability of building specific and concise notations to express a problem domain, as well as safe (as predictable) code due to semantic restrictions. Both of these properties are very desirable in the process of developing insider threat specifications.

Thus, a methodology for deriving a Domain Specific Language includes three important steps:

- the abstraction of the domain, which involves the removal of all the unnecessary details of the environment;
- the systematic categorisation of the necessary (abstracted) details into language semantics;
- the process of engineering the developed semantics into software.

The derivation of the necessary abstractions is achieved partly by the establishment of the aforementioned insider threat mitigation frameworks. These frameworks rely on two important elements that achieve the abstraction. The first is a careful classification of the Insider IT threat elements. The classification schemes (or taxonomies) enhance the ability to examine the problem in a more systematic way, could certainly form the core of a specification language and are a common occurrence in the Information Security literature. The second element is a model that combines all the threat elements and provides an estimate for the magnitude of the threat.

Whilst the models are discussed in detail by various researchers (Wood, 2000; Schultz, 2002; and Magklaras and Furnell, 2002), the section that follows will focus on the taxonomy issue. The formation of a structured way to identify insider IT misuse threat elements forms a key component of a threat specification language. The language semantics and the process of engineering them into software will also be discussed in latter sections.

A taxonomy for Insider Misuse Threat Prediction

An overview of intrusion specification taxonomies is provided by Furnell et al. (2001). Amongst these taxonomies, one that specifically addresses insider IT misuse incidents is given by Tuglular (2000). This taxonomy integrates an established security policy to the process of classifying computer misuse incidents in three dimensions: incident, response and consequences. These dimensions can be divided into additional sub-dimensions that

further classify a particular misfeasor. Tuglular's paper is one of the first to suggest a 'target-type of threat' association as a way to prevent insider misuse. The target is an 'asset' and the rule is called a 'strategy' in the terminology he proposes. The suggestion is mentioned in a single sentence and forms the basis for a methodology to predict insider misuse threats. If one can associate successfully certain actions to threats then it establishes the first step towards systematizing insider IT threat prediction.

Most research efforts in the field of intrusion taxonomy classification are still at an early stage. The Tuglular taxonomy, and others mentioned in (Furnell et al. 2001), are useful for the systematic study of intrusions, but they offer little help to a process designed to automatically detect intrusive activities. This is because the classification criteria employed by these taxonomies cannot be qualified or quantified very easily by an Intrusion Detection System with the level of information they exhibit. Moreover, none of these taxonomies is tailored for the process of estimating the likelihood of Insider Threat.

The best way of enhancing the expressiveness of an intrusion taxonomy scheme for insider misuse activities is to focus on the human actions and how their consequences impact the elements of the IT infrastructure that are being targeted. The idea is that it is easier to detect which particular element is affected by a potentially intrusive action, rather than focusing on the task of sensing the motives for initializing an attack.

Another important property of a suitable Insider IT misuse prediction taxonomy is the freedom of the security architect to choose what can be considered as an Insider IT

misuse threat indicator. Most taxonomies enforce a rigid framework for classifying phenomena with clear borders of distinction that offer little space for subjective or varying interpretation of facts. This schema does not fit the case of Insider IT misuse prediction. The previous section of this paper argued that there are different views for what is considered as legitimate user misuse amongst the various organizations. Consequently, there are also different views for what is perceived as an insider threat prediction indicator and a taxonomy tailored for the needs of a threat prediction process should be flexible enough to accommodate this fact.

Take in Figure 4

As a result, one can construct a suitable threat prediction taxonomy based around consequences detected at system level. Figure 4 above displays the top level of the taxonomy structure indicating the three primary, non-mutually exclusive levels that address these consequences.

The Operating System (O/S) based consequences are branched down to two sublevels of file-system and memory manipulation, illustrated by Figures 5 and 6 respectively. A justification for this is that a large number of security faults (Aslam et al. 1996) involve filesystem and memory management issues, and indeed the core modules of UNIX (Bach, 1986) and Windows-based (Richter, 1997) operating systems provide (amongst others) specific support for the related functions. Hence, it is safe to assume that these

two kernel functional attributes can be used as a strong criterion for further classifying legitimate user activities.

Take in Figure 5

At File/Directory level, a misuser may attempt to read or alter (write/create) certain files. These files might contain sensitive or unauthorised information (information theft or fraudulent modification of vital information). A knowledgeable insider might also attempt to read or modify file information that is not directly related to its content. Bach (Bach, 1986) and Richter (Richter, 1997) emphasize that most Operating Systems allow a file to contain additional information such as access/creation/modification times as well as information that relates the file to its owner and permits access to it under certain conditions. Although the mechanisms that implement these file attributes are different amongst Operating Systems, they are collectively known as file metadata and they are vital mechanisms to secure the privacy, availability and integrity of the file contents. Consequently, they are good candidates for exploitation by a legitimate user who is about to perform a deliberate or accidental misuse act.

The points mentioned in the previous paragraph are also valid for ‘filesystem’ related data. Every Operating System organizes its files and directories by means of a specific set of rules that define how a file (contents and metadata) are about to be stored on the physical medium. The Operating System sub-modules that handle these issues are known as **filesystems**. Attempts to read or alter the physical medium’s Master Boot Record

(MBR), intentional or accidental modification of partition table data are some of the most notable auditable actions that could point to legitimate user misuse acts. Robert Hanssen's case is a classic reminder of this kind of activity. His specially modified 40-track floppy disk was created by a set of filesystem modification actions, in order to create a hidden area to store the sensitive information (Slashdot, 2001).

In addition to filesystem content and metadata modification, a survey of insider misuse conducted by the authors (Magklaras and Furnell, 2004) showed that excessive disk space consumption is perceived as a problem for many of the respondents. Under certain conditions that depend on the configuration of the IT infrastructure, a legitimate user might produce a deliberate or accidental Denial of Service attack (DoS).

Take in Figure 6

While the filesystem provides useful insights about the actions that could indicate a potential for IT misuse acts, an equally interesting picture of insider threats could be drawn from observing the Random Access Memory (RAM) of the system. The reason is simple. Every time an application is executed, a substantial part of its contents (program instructions together with user supplied runtime data) are transferred to RAM, where the execution of that application takes place. The 'Memory Manipulation' sub-category examines how actions related to potential misuse acts could be categorised in terms of observable system memory events (Figure 6).

Memory inspection is the best way to see if a legitimate user attempts to run or even install a suspicious program. Indeed, it is one of the core techniques used in the detection of overtly malicious code, such as viruses and Trojan horse programs. However, software threats do not end here, and a problem originating from end-user actions was highlighted by the authors' aforementioned survey. The majority of the respondents (76%) claimed that an attempt to install one or more unauthorized applications is also classified as a misuse act for their organizations**.** Hence, this could be used as a strong criterion for the purposes of sensing insider threats in an IT environment. The execution or installation of these programs could be intercepted by either recognizing a program's footprint in memory or by intercepting a well-known series of system calls produced by various suspicious programs. For example, the fact that a non-advanced user is trying to compile an advanced vulnerability scanning tool is an event that should be noticed, and serves as a good indicator of potential misuse activities that are about to follow.

In addition, attempts to consume large memory portions of an operational system that are related to a legitimate user account can serve as good indicators of (intentional or accidental) insider misuse at Operating System level. One might argue that the 'irregular memory usage' sub-categories should really belong under the 'Program execution' hierarchy of events. However, it is possible that someone will produce a quick and easy Denial of Service attack on a running system by forcing the host to commit large portions of system memory to a process, as demonstrated in various case studies described in (Moore et al. 2001). Moreover, a large category of security faults can be achieved by means of accessing normally restricted memory areas, creating what is commonly known

as a “buffer overflow” attack (Frykholm, 2000). As a result of these issues, it was felt that a separate sub-category hierarchy should exist to describe these events.

The filesystem and memory manipulation consequences conclude the O/S consequence category of the proposed taxonomy. The next category, “network consequences”, represents another distinct set of factors that could be taken into consideration in order to classify insider misuse threat indicators. Figure 7 illustrates the network-related consequences of acts that could be used as legitimate user threat indicators.

The authors’ insider misuse survey indicated that 26% of the surveyed IT professionals consider the content of web pages that a legitimate user visits to be an important Threat Indication factor. Hence, it is reasonable to assume that URLs that contain a ‘promising’ link to sexually explicit content or to illegal software downloads should be noted as distinct ways of indicating potential to misuse the system (suspicious URLs).

Take in Figure 7

Network packets that are associated with certain legitimate users and indicate the usage of a variety of network protocols and applications that might introduce certain vulnerabilities are also distinct ways of accidental or intentional IT misuse. For example, it could be said that a user that employs the TELNET (Postel and Reynolds, 1983) protocol to login to a multi-user system is more likely to have her account compromised than a user who logins via the Secure Shell (SSH) application (Ylonen, 1995) due to the

fact that the earlier application transmits the user password in clear-text form across the network, whereas the latter one encrypts it.

Someone might also like to differentiate between TCP and UDP based applications/protocols. From a potential threat point of view, UDP services are less secure than TCP based ones. For example, Ziegler (2002) discusses in detail how UDP's lack of flow control and state mechanisms can create various data security problems. Consequently, the distinction between the usage of UDP and TCP services can serve as a potential insider misuse threat indicator, on the basis that UDP services are more likely to be accidentally (or intentionally) abused by a legitimate user.

The participants in the authors' survey indicated that resource over-utilization is an existing issue in IT infrastructures. Although the 'Filesystem Manipulation' subcategory of the taxonomy indicates ways with which disk storage capacity can be misused, the results of over-utilisation can also affect network capacity. For instance, a legitimate user could start downloading massive quantities of data, exceeding the network bandwidth cost budget of a business (Downloading over X Mbytes of data in a period Y). The X and Y number limits can be selected by the network administrator according to the company budget requirements.

In addition, a legitimate user might also cause network congestion by exceeding the data network's 'burst' or throughput capacity or exhausting the number of available network endpoints, as described by Sharda (1999). Bandwidth hungry applications, such as video

streaming players, and multiple data transfers can cause congestion that can severely impact the performance of a data network or affect the Quality of Service (QoS) of certain applications that require sustained data network throughput.

Finally, incoming or outgoing SMTP headers or attachments might indicate activity related to e-mail misuse that can certainly be traced in network or host level. Outgoing e-mails that contain a set of particular files as attachments (e.g. password database files, other sensitive material) and have unusual destination addresses (e.g. unknown Hotmail accounts, a large number of recipients) should serve not necessarily as intrusion indicators but as insider threat estimators. The plethora of malicious code efforts and phishing techniques may have an external origin, but the threat is realized by the actions of unsuspecting legitimate users. In addition, proprietary information theft could also be realized by means of emailing sensitive material to non-authorized external entities.

The last system consequences category (“hardware”) plays an important role in preventing a number of computer system threats. Insiders can often access the physical hardware of the machine very easily. Thus, removal or addition of hardware components, as well as modifications of their default configuration are some of the events that may act as important indicators of insider misuse prediction in a computer system.

From a taxonomy to encoding and language semantics

After identifying and characterizing the insider IT misuse threat factors, the next issue concerns the development of the encoding schemes and semantics of the desired language. Earlier sections made reference to the concept of Domain Specific Languages (Consel, 2004) and the first steps for devising a suitable threat specification language have been made. The Common Intrusion Specification Language (CISL) (Feiertag et al. 1999) consists of a semantic framework to unambiguously describe intrusive activities together with proposed data structures that store the event information and can form standardized messages exchanged by various Intrusion Detection System (IDS) components.

The CISL framework could be re-used for producing a suitable Insider Threat Prediction Specification Language. However, the framework would require substantial re-engineering to achieve this goal. The existing CISL framework and the latest related research are summarized in the paragraphs that follow. The discussion then proceeds to present the CISL major flaws from an insider threat specification perspective, and suggests an approach to overcome these problems.

In CISL, the semantic representation of intrusive activities is achieved by the formation of an S-Expression. This is a recursive grouping of tags and data, delimited by parentheses. The tags provide semantic clues to the interpretation of the S-Expression and

the data might represent system entities or attributes. For this reason, the tags are also called Semantic Identifiers (SIDs).

The best of way of illustrating how CISL works is by considering an example. The statement (Hostname ‘frigg.uio.no’) is a simple S-Expression. It groups two terms, without semantically binding them. One can guess that it refers to a computer system with the FQDN name ‘frigg.uio.no’, but the true meaning of the statement is still vague. In fact, the full semantic meaning of S-Expressions becomes apparent when one forms more complex S-Expressions, by means of combining several SIDs into a sentence.

Figure 8 illustrates a suitably crafted CISL intrusion specification which could be translated in the following plain English translation:

“*On the 24th of February 2005, three actions took place in sequence in the host ‘frigg.uio.no’. First, someone logged into the account named 'tom' (real name ‘Tom Attacker’) from a host with FQDN 'outside.firewall.com'. Then, about a half-minute later, this same person deleted the file '/etc/passwd' of the host. Finally, about four-and-a-half minutes later, a user attempted but failed to log in to the account 'ksimpson' at 'frigg.uio.no'. The attempted login was initiated by a user at 'hostb.uib.no'.*”

The particular CISL sentence describes a malicious attack that erases an important system file of a UNIX system and consists of three multi-SID S-Expressions. In general, a sentence can be formed by one or more S-Expressions nested at different levels.

However, there are strict rules that allow the nesting of S-Expressions. The rules are defined by the nature of the SIDs, as there are several different types of them.

Take in Figure 8

Verb SID's are joined together in a sentence by conjunction SIDs. In the previous example of Figure 8, 'And' is the conjunction SID that holds together the three SIDs that form the sentence. In addition, a CISL sentence might employ role, adverb, attribute, referent and atom SID types. Role SIDs indicate what part an entity plays in a sentence (such as 'Initiator'). Adverb SIDs provide the space and time context of a verb SID. Attribute SIDs indicate special properties or relations amongst the sentence entities, whereas atom SIDs specialise in defining values that are bound to certain event instances (for instance 'Username'). Lastly, referent SIDs allow the linking of two or more parts of a sentence ('Refer to' and 'Refer as'). There are additional SID types, but the aforementioned ones are the most commonly employed.

One can clearly observe a structural hierarchy for forming complex sentences that also contributes to the semantic meaning. This semantic structure is inspired by the syntax of natural languages. A verb is always at the heart of every sentence and is followed by a sequence of one or more qualifiers that describe the various entities that play parts in the sentence, or qualify the verb itself. In addition, a similar hierarchy is also reflected in the formation of the previously described insider misuse taxonomy.

CISL (Feiertag et al. 1999) is not only about semantic rules. Its authors were concerned with the encapsulation of the structured semantic information into the 'Generalised Intrusion Detection Object' (GIDO), data structures that hold the encoded event information. The purpose of encoding the information in a standard way is to make the process of exchanging the information amongst various IDS components easy.

The Common Intrusion Detection Framework (CIDF) that embodies CISL (Feiertag et al. 1999) considers an IDS as a group of discrete functional components that exchange messages.. Some of the components intercept an intrusion event (E-boxes) or organise them into searchable collections (D-boxes), whereas others analyze it (A-boxes) to determine whether the event is worth looking and event take some sort of action (R-boxes). One of the major objectives behind this conceptual IDS view was to enable seamless integration that accommodates for inevitable differences in IDS implementations. This is another important issue that concerns the formation of an ITPSL.

Unfortunately, despite the well-conceived interoperability target, the CISL GIDO encoding process introduced many problems. Doyle (1999) has criticized many of the aspects of the CISL GIDO structure. Although the purpose of the document was to evaluate the fitness of CISL for use in the DARPA Cyber Command and Control (CC2) initiative, the paper identifies serious inadequacies that concern the CISL time resolution data representation facilities, as well as data throughput limitations caused by the fixed size of the GIDO data structure. Finally, Doyle comments on the lack of support for the

next generation Internet Protocol (Version 6). Whilst these points are fair, they could easily be corrected by making the necessary changes to the relevant data types and overcome the perceived obstacles. In fact, section 7 of the CISL standard (Feiertag et al. 1999) contains specific guidelines that explain how to add information to a GIDO, to clarify or correct its contents. This suggests that the encoding principles are certainly extensible.

A more serious aspect of Doyle's critique in (Doyle, 1999) refers to the semantic structure of the CISL language. In particular, his criticism that CISL has "no facilities for representing trends or other complex behavioral patterns; ill-specified, inexpressive, and essentially meaningless facilities for representing decision-theoretic information about probabilities and utilities" indicates that the language would be a bad choice for describing information about a threat prediction model. The basic reasoning behind this critique is that CISL is too report-orientated and threat mitigation requires a different level of information, not just mere report structures of what is happening on one or more systems. These indeed represent more serious limitations that would require a more radical re-design of the CISL.

In response to the CISL encoding limitations, the IETF Intrusion Detection Exchange Format working group (see www.ietf.org/idwg) took over the scope of the CIDF work. It addressed most of the GIDO encoding issues by introducing a new Object Oriented format for encoding and transmitting Intrusion Detection related information. The Intrusion Detection Message Exchange Format (IDMEF) (Curry et al. 2004) enriched the

type of standardized information that IDS sensors may represent, as well as the process of standardizing the exchange of messages using protocols such as IDXP (Feinstein et al. 2004) and data exchange languages such as XML (W3C, 2006). For example, the IDMEF 'Confidence' and 'Impact' classes can now be used to represent decision theoretic information (Curry et al. 2004). The earlier can assign a confidence and thus a probability to an observed event, whereas the latter relates privilege escalation consequences to three broad severity levels. This functionality can serve as the basis for encoding probabilistic information, in order to use it in an ITPSL concept.

These standardization features were lacking from the previous CIDF platform and they constitute a very important step towards establishing better interoperability amongst different IDS products. However, at the time of writing, the working group has not managed to expand on the semantic scope the CISL language and address the various expressiveness issues that Doyle mentioned. The IDMEF draft standard (Curry et al. 2004) proposes more extensive encoding and data structures, but it does not suggest semantic guidelines like the ones proposed by the CIDF framework. For IDMEF, the term 'language' refers to the data types and encoding principles for IDS data and not to the syntactical guidelines of an Intrusion Specification Language.

Hence, the establishment of an Intrusion Specification Language tailored to Insider Threat Prediction could be achieved by adopting the basic syntactic guidelines of the CISL framework and address the syntactic inadequacies indicated by Doyle (1999). After the semantic refinement step, an effort to match the suggested event expression

statements to the IDMEF data structures should take place. This will ensure that the ITPSL scheme would be fully compliant with the relevant IETF standards of the research field.

Figure 9 below illustrates the process of turning an ITSPL-based text description into a multi-level threat signature. A high-level text description of the threat is parsed by a suitably crafted compiler and turned into network, file and memory-level (multi-level) statements that detect the different threat components at system level. The produced signature could then populate a database of signatures, such as the one Magklaras (Magklaras, 2005) proposes for the Insider Threat Prediction Model (ITPM). This process could also facilitate the building of a world-wide case repository of insider threat cases, such as the one mentioned in (NSTISSAM, 1999). This would benefit computer security analysts and forensic specialists as well as IDS vendors.

Take in Figure 9

The process of refining the original CISL semantic schema would enrich the original language by adding new atom and adverb SID types that represent decision-theoretic and probabilistic information. For example, user privileges related to authorized or not network, file and memory level operations can be represented by the IDMEF 'Impact' class. In addition, there are plenty of IDMEF data structures that can represent information related to the file, network and command execution ITPSL expression components. The 'FileList' and 'FileAccess' classes contain adequate attributes to

represent the file attributes. The ‘Address’ class can represent network related data, and lastly, the ‘Process’ class could accommodate most of the requirements of the command execution data of the ITPSL schema.

Such a language would help one to establish more easily insider threat signatures that could be used in various IDS implementations and computing architectures. Figure 10 illustrates how the language interacts with the ITPM model (Magklaras and Furnell, 2002). The ITPSL encoded threat signature is fed into a module that translates its contents to Operating System specific Application Programming Interface (API) directives. Each OS/Computing platform implements different mechanisms to facilitate the monitoring of filesystem, network and memory related events. The translation of the ITPSL encoded statements to platform specific instructions achieves the desired platform independence feature. The monitoring modules feed the ITPM model with the necessary data, in order to establish whether a user constitutes a threat with respect to the signature contents. Whilst the ITPM is shown here to interact with ITPSL, the scheme could also prove useful to other threat modeling efforts. The produced positive or negative result could then be used by an IDS or IPS system, in order to further increase (or reduce) the intensity of monitoring various operational aspects of a system or react to prevent/block intrusive activity respectively.

Take in Figure 10

For instance, let us consider the hypothetical case of a malicious insider stealing proprietary information and forwarding it to a rival company. Assuming that the misuser gets caught, a security specialist normally gathers forensic evidence from the computing infrastructure. He might look at the media used to transfer information, the information access patterns, the contents of emails and personal storage media. He could then establish the ITPSL text level description of the incident on a repository database. A researcher or IDS vendor product engineer could then acquire the posted signature, recreate the misuse threat and see how he could improve the detection of the threat at different stages and computer architecture levels (network, file and memory level). He could also refine the signature, in order to include undiscovered variations of the incident, as the language framework should provide a good way to structure insider threat information. The produced signatures could then be re-used in future systems to model and warn about eminent threats of similar nature.

ITPSL in comparison to currently available security tools

Earlier sections of the paper discussed the lack of facilities for systematically describing the elements of misuse incidents in current threat mitigation frameworks. Nevertheless, there is currently a variety of tools that help IT practitioners monitor and respond to insider activities. The variety of commercial and open source solutions is too large to include an exhaustive discussion of all available tools here. However, a comparison of the proposed language features and the currently employed IT security tools will indicate where this research effort fits in the overall field of practice.

Internet firewalls (Zwicky et al. 2000) are commonly employed tools looking closely at data passing through today's networked IT infrastructures. There are many types of firewalling mechanisms, ranging from stateless and stateful packet filtering to more sophisticated application-aware network filters. Irrespective of the mechanism employed, the basic idea is that network traffic is inspected at protocol and possible payload level in search of patterns or trends that indicate malicious traffic. Although firewalls were traditionally employed to protect an IT infrastructure from attacks of external origin, they are currently utilized to block traffic from the inside to the external world and in that respect they can act as mechanisms to mitigate insider threats. In fact, most of the networking consequences of the proposed taxonomy (Figure 7) could be mapped to firewall toolkit rules.

Intrusion Detection and Prevention Systems (IDS/IPS) are some of the latest tools that provision more refined mechanisms to detect and prevent an information security breach (Endorf et al. 2003). IPS devices exercise access control mechanisms to protect computer systems from malicious acts. They were originally developed in an attempt to increase the accuracy of passive network monitoring techniques and provided large improvements over the aforementioned firewall mechanisms. IPS devices could be viewed as extensions of IDS mechanisms. IDS are devices designed not only to prevent and (where possible) respond to a plethora of computer security incidents, but also to integrate the operation of other security components (anti-virus, firewall and cryptographic applications) into one

overall system. The implications of this integrated approach are that IDS/IPS approaches examine both host and network based data to mitigate threats.

The File-system manipulation O/S consequences (Figure 5) as well as the Memory Manipulation O/S consequences (Figure 6) of the proposed taxonomy are typical examples of concepts that are today directly applicable to IDS/IPS solutions targeting insider attack vectors. Thus, one might wonder about the necessity of the proposed language. If most of the proposed detection criteria of the taxonomy that abstracts the problem are already employed in available solutions today, what is the need for yet another language? The answer to this question lies in how the signatures are encoded and how easy it is for a security administrator to encode a scenario using current security tools targeting insider incidents. Firewalls, IDS/IPS, antivirus and anti-spyware solutions have rule writing conventions that could to some extent be viewed as mini DSL constructs. Examples of these rule writing conventions are the IPTABLES firewalling rules (Ziegler, 2002) and Sourcefire's SNORT rule parser engines (Beale et al. 2003), which are widely employed to encode intrusion signatures for their IPS/IDS product series. Similar examples can be found on other firewall and IPS/IDS product offerings, as well as antivirus solutions. In fact, anti-virus vendors construct the signatures and offer them as part of their product, with their customer not engaging at all in any stage of the virus signature construction.

The common traits of today's available solutions indicate proprietary coding schemes or schemes that require a substantial amount of system-specific level of knowledge, in order

to encode a threat signature, with evident cross-vendor boundaries. A threat signature from product vendor A will generally not be usable with the product of vendor B, and when it is, the effort and the compatibility nightmares will always make the task of porting signatures an undesirable overhead. This is exactly where ITPSL fits into the picture. It can provide not only the means for constructing a structured repository of insider misuse cases but also act as a complement of IDS/IPS and other frameworks or tools (Figure 10), providing a high level 'glue' to describe insider threat components. This component could be used by commercial vendors not only as an information repository but also as a tool that eases the porting of signatures and scenarios to their product platform.

Conclusions

Insider Threat is a problem that affects the well being of IT infrastructures. Various frameworks for mitigating insider misuse exist following different philosophies of approaching the problem. However, all frameworks lack a way of describing precisely acts of legitimate user misuse, an important ability for every researcher in the field. A domain specific language tailored around insider misuse incidents can facilitate this need and enhance the capabilities of these frameworks.

Although the paper has presented the concept of the language, the development of the proposed approach is currently a work in progress. As such, it would be premature to

attempt to convey more specific details at this stage. Indeed, constant refinement of the semantics and language interface mechanisms is expected, especially during the early stages of its development. An important first step in the process of constructing such a language is the abstraction of the problem domain by means of classifying insider misuse incidents. Insider taxonomies are frequently encountered in the research literature. However, the building of an insider misuse language requires a threat taxonomy based on consequences detected at system level. This design approach would allow the language to fit easily around events that can be captured in an automated fashion and not on parameters that need to be deduced such as motive, for example.

The proposed taxonomy could then pave the way for encoding threat signatures. The CISL (Feiertag et al. 1999) and the IDMEF (Curry et al. 2004) frameworks are examples of previous research attempts to provide standardized semantics for specifying intrusions, as well as ways to encode intrusion specific information. By adapting their semantics and data structures to the field of insider misuse, one could produce a mechanism to encode insider threat specific information and make use of it in insider threat modeling frameworks.

References

Aslam, T., Krsul, I. and Spafford E. (1996), Use of a Taxonomy of Security Faults, Technical Report TR-96-051, COAST Laboratory, Department of Computer Sciences, Purdue University, IN.

Beale, J., Foster, J.C. and Posluns, J. (2003), Snort 2.0 Intrusion Detection, Syngress Publishing, ISBN: 1931836744

Bach, M. (1986), The design of the UNIX Operating System, Prentice Hall International Editions, NJ.

Caelli, W., Longley, D. and Shain, M. (1991), Information Security Handbook, Stockton Press.

CNN.com (2002), "The case against Robert Hanssen", In-Depth Special series. Available http://edition.cnn.com/SPECIALS/2001/hanssen/

Consel, C. (2004), "From A Program Family To A Domain-Specific Language", in Lengauer, C.; Batory, D.; Consel, C.; Odersky, M. (Eds.), Domain-Specific Program Generation, Lecture Notes in Computer Science 3016, Springer-Verlag, pp. 19-29.

Curry, D., Debar, H. and Feinstein, B. (2004), The Intrusion Detection Message Exchange Format, Internet Draft, Intrusion Detection Exchange Format working group, Internet Engineering Task Force, 8 July 2004.

Doyle, J. (1999), Some representational limitations of the Common Intrusion Specification Language, Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139, November 1999 Revision

Endorf, C., Schultz, E. and Mellander, J. (2003), Intrusion Detection and Prevention: The Authorative Guide to Detecting Malicious Activity, McGraw Hill, ISBN: 0072229543.

Feiertag, R., Kahn, C., Porras, P., Schnackenberg, D., Staniford-Chen, S. and Tung, B. (1999), A Common Intrusion Specification Language (CISL), 11 June 1999, Available http://www.isi.edu/~brian/cidf/drafts/language.txt

Feinstein, B., Matthews, G. and White, J. (2002), The Intrusion Detection Exchange Protocol (IDXP), Internet Draft, Intrusion Detection Exchange Format working group, Internet Engineering Task Force, 22 April 2003, Available http://www.ietf.org/internet-drafts/draft-ietf-idwg-beep-idxp-07.txt.

Frykholm, N. (2000), Countermeasures against Buffer Overflow Attacks, White Paper, RSA Laboratories.

Furnell, S., Magklaras, G., Papadaki, M. and Dowland, P. (2001), “A Generic Taxonomy for Intrusion Specification and Response”, Proceedings of Euromedia 2001, Valencia, Spain, 18-20 April 2001, pp. 125-131.

Gibbons, R. (1992), A Primer in Game Theory, Prentice Hall International.

Helbig K. (1993), Modelling the Earth for Oil Exploration: Final Report of the CEC's Geoscience I Program 1990-1993, Pergamon Publishing.

Magklaras, G. (2005), An Architecture for Insider Misuse Threat Prediction in IT Systems, MPhil Thesis, School of Computing, Communications and Electronics, University of Plymouth, UK.

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

Magklaras, G. and Furnell, S. (2004), "The Insider Misuse Threat Survey: Investigating IT misuse from legitimate users", *Proceedings of the 5th Australian Information Warfare & Security Conference*, Perth Western Australia, 25-26 November 2004, pp. 42-51.

Moore, D., Voelker, G. and Savage S. (2001), "Inferring Internet Denial of Service Activity", Proceedings of the 10th USENIX Security Symposium, Washington D.C, August 2001, pp. 9-22.

NSTISSAM. (1999), The Insider Threat To US Government Information Systems, NSTISSAM INFOSEC /1-99, U.S. National Security Telecommunications And

Information Systems Security Committee, Available http://www.cnss.gov/Assets/pdf/nstissam_infosec_1-99.pdf.

Pfleeger, C. and Pfleeger, S. (2003), Security in Computing, Third Edition, Prentice Hall.

Postel, J. and Reynolds, J. (1983), TELNET Protocol Specification, Request For Comments (RFC) 854, IETF Network Working Group, May 1983.

PWC. (2004), Information Security Breaches Survey 2004 – Technical Report, Available http://www.pwc.com/images/gx/eng/about/svcs/grms/2004Technical_Report.pdf

Richardson, R. (2003), 2003 CSI/FBI Computer Crime and Security Survey, Computer Security Institute. Spring 2003.

Richter, J. (1997), Advanced Windows, Microsoft Press, Redmond, Washington.

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

Sharda, N. (1999), Multimedia Information Networking, Prentice Hall Inc., Chapter 12.

Slashdot (2001), “Spying and Technology: Robert Philip Hanssen”, posting on Slashdot.org, 22 February 2001, Available http://slashdot.org/articles/01/02/22/0622249.shtml

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

W3C. (2006), Extensible Markup Language, Architecture Domain, World Wide Web Consortium (W3C), Available http://www.w3.org/XML/

Wood, B. (2000), An insider threat Model for Adversary Simulation, SRI International, Research on Mitigating the Insider Threat to Information Systems - #2: Proceedings of a Workshop Held by RAND, August 2000.

Ylonen, T. (1995), The SSH (Secure Shell) Remote Login Protocol, Internet Draft, IETF Network Working Group, 15 November 1995, Available http://www.free.lp.se/fish/rfc.txt

Ziegler, R. (2002), LINUX Firewalls, Second Edition, New Riders Publishing, Chapter 2, pp. 48-50.

Zwicky, E.D., Cooper, S. and Chapman, D.B. (2000), Building Internet Firewalls, Second Edition, O’Reilly & Associates, ISBN: 1565928717

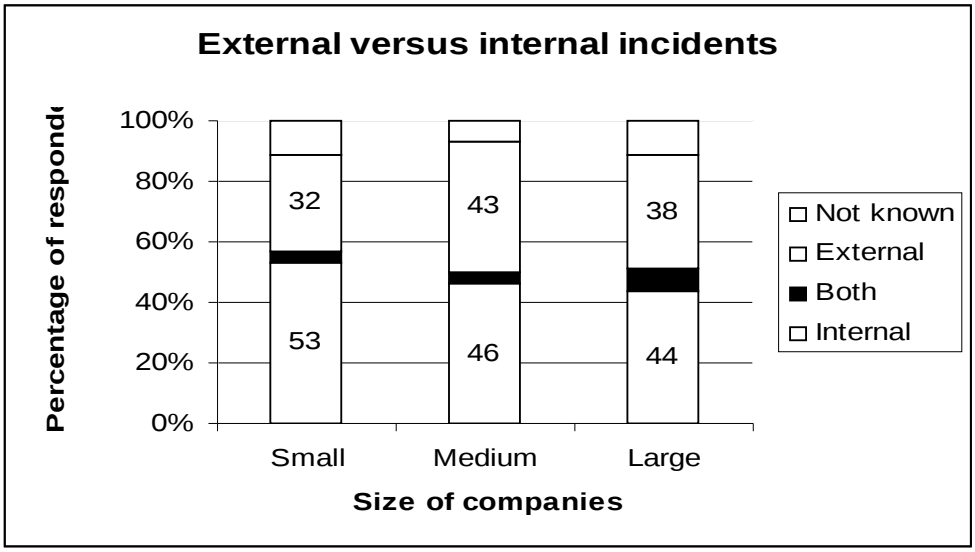

Figure 1: External versus internal incidents in terms of report frequency (PWC, 2004)

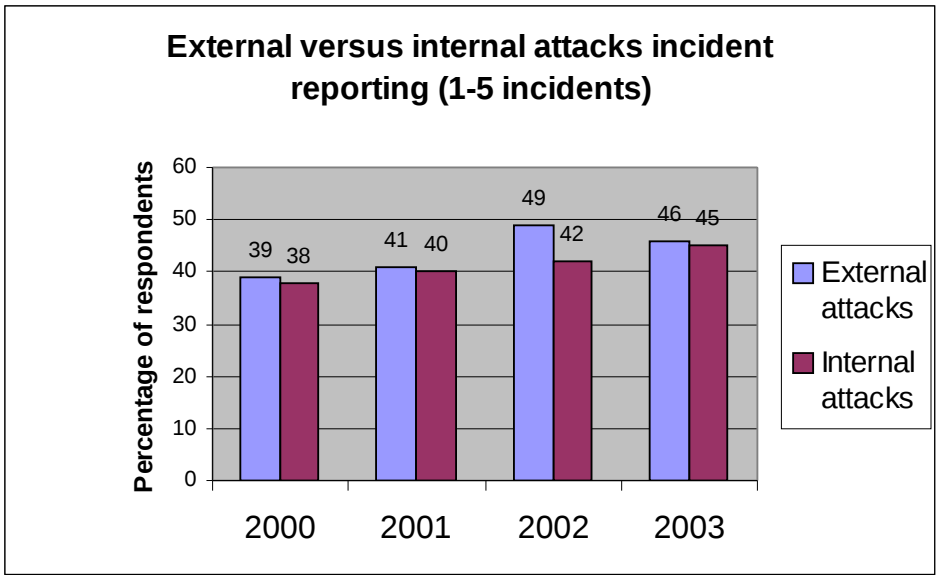

Figure 2: External versus internal attack incident frequency (Richardson, 2003)

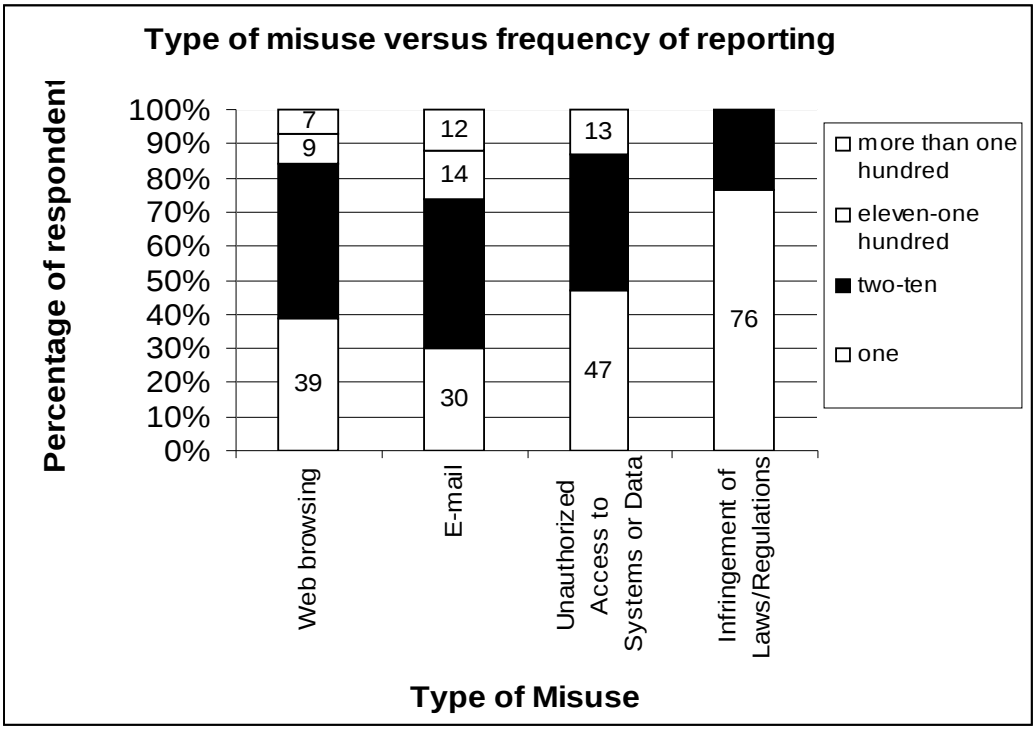

Figure 3: Types of misuse reported by UK businesses (PWC, 2004)

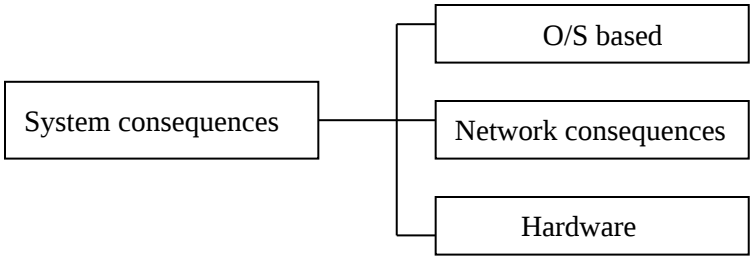

Figure 4: Top level of an insider threat prediction taxonomy

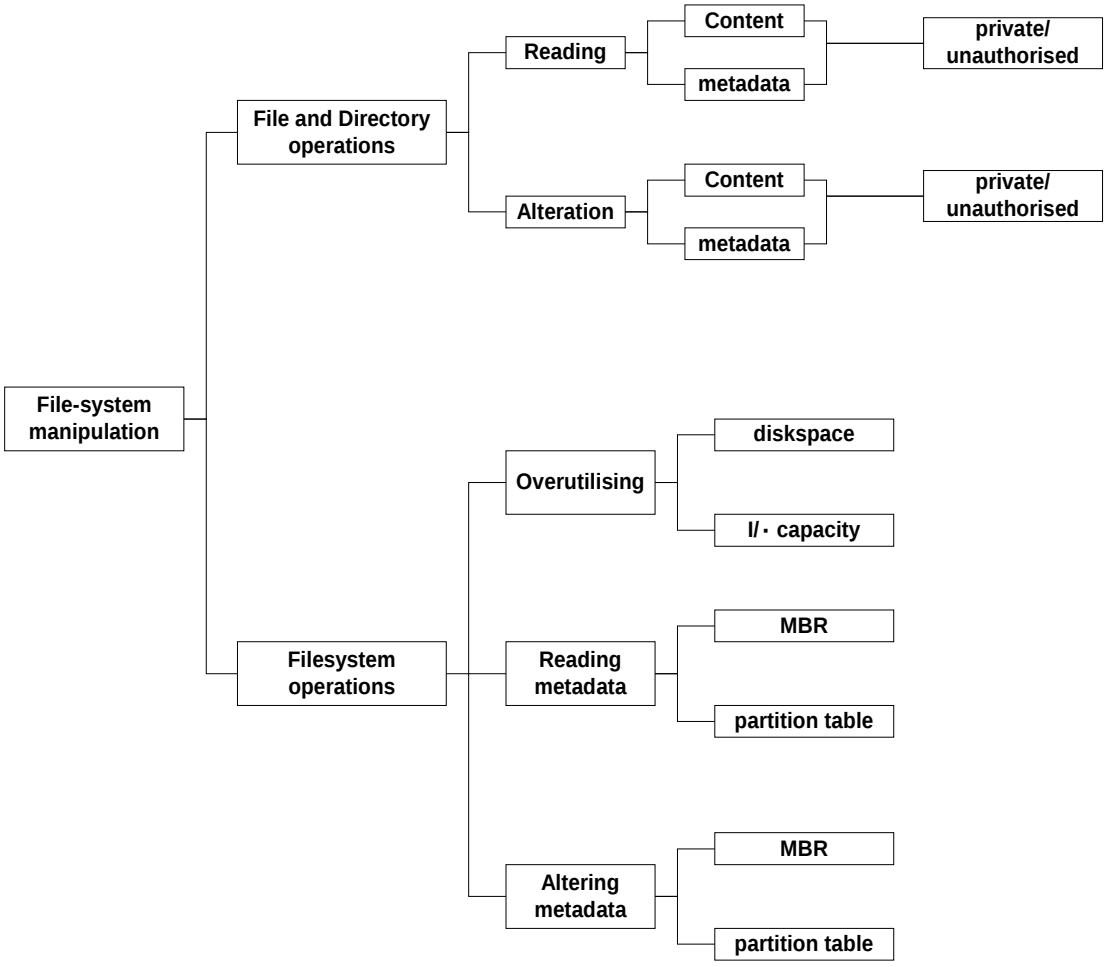

Figure 5: File-system manipulation O/S consequences

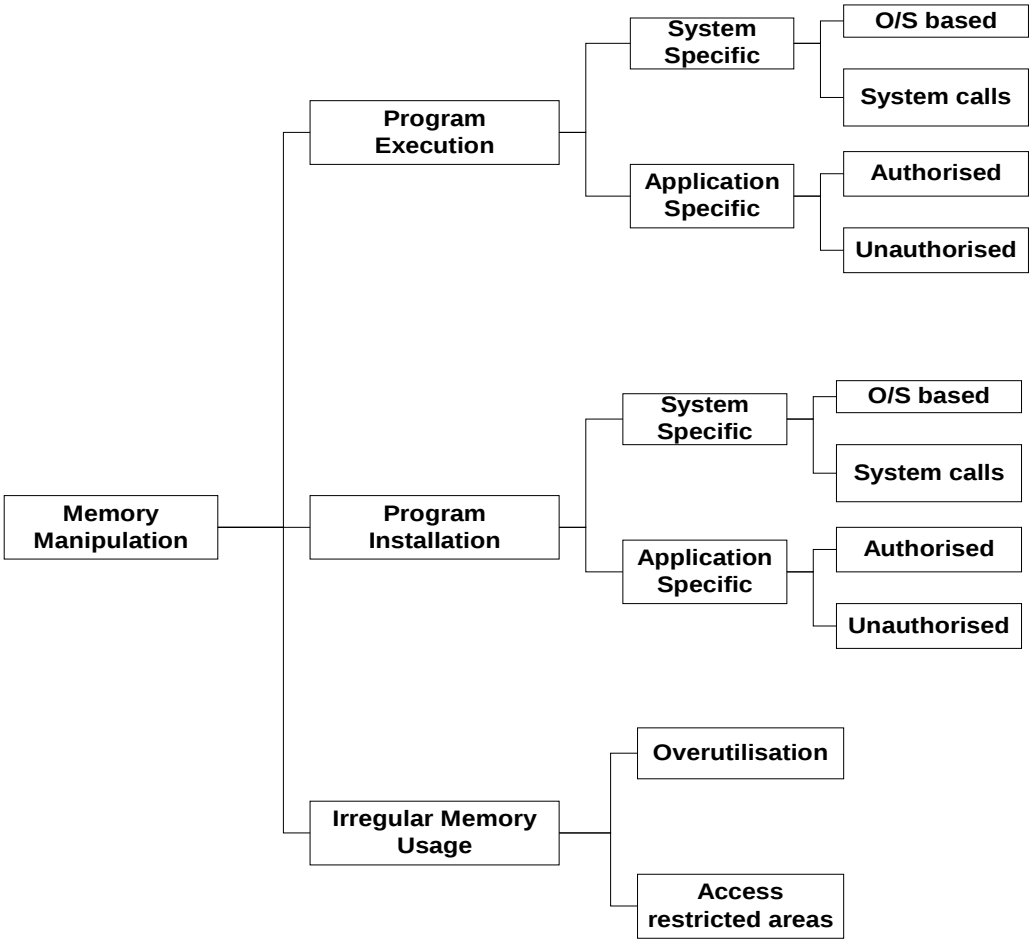

Figure 6: Memory Manipulation O/S Consequences

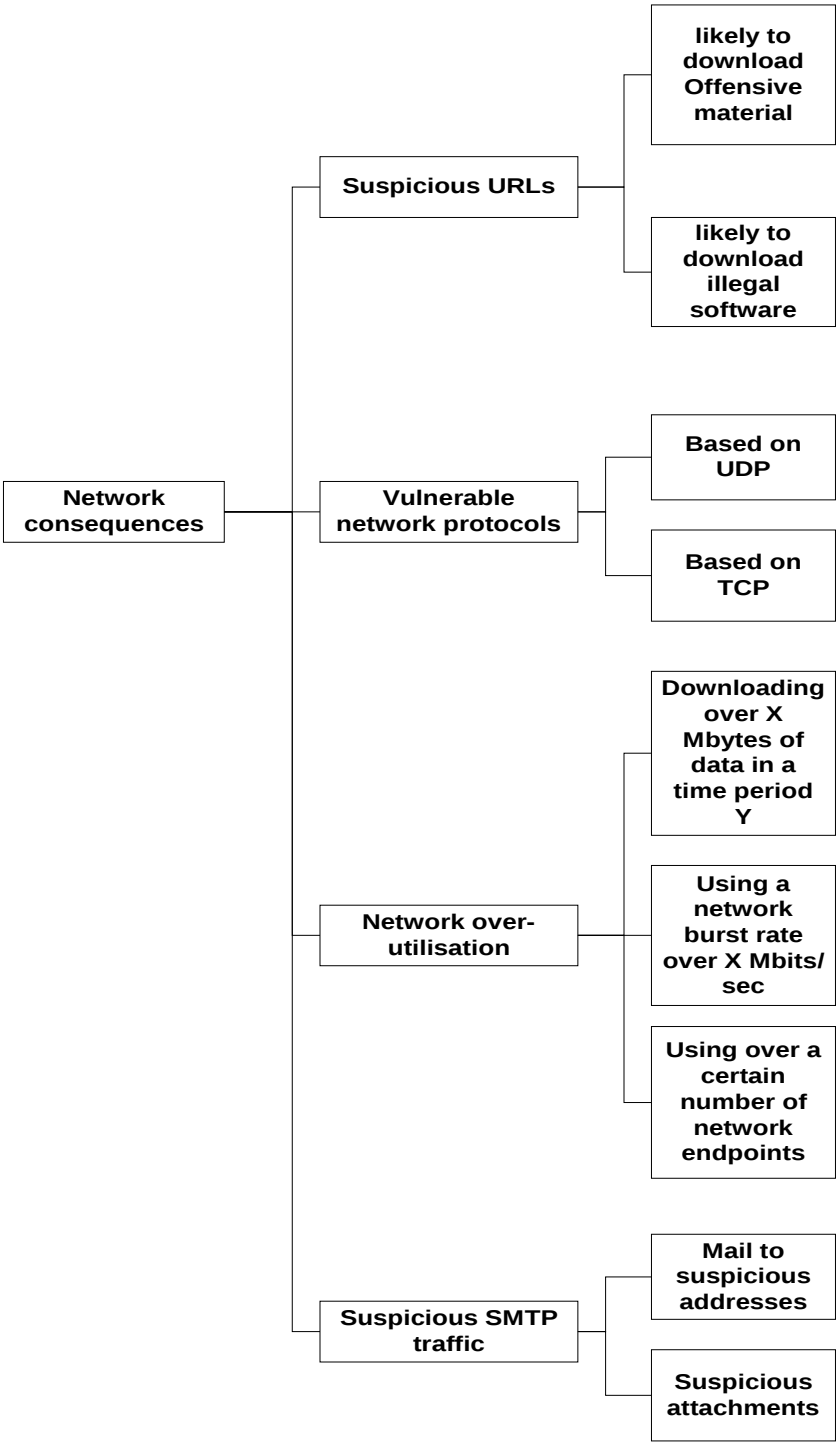

Figure 7: Network consequences of the insider IT misuse prediction taxonomy

```
(And
        (OpenApplicationSession
            (When
                (Time 14:57:36 24 Feb 2005)
            )
            (Initiator
                (HostName 'outside.firewall.com')
            )
            (Account
                (UserName 'tom')
                (RealName 'Tom Attacker')
                (HostName 'frigg.uio.no')
                (ReferAs 0x12345678)
            )
            (Receiver
                (StandardTCPPort 22)
            )
        )
        (Delete
            (World Unix)
            (When
                (Time 14:58:12 24 Feb 2005)
            )
            (Initiator
                (ReferTo 0x12345678)
            )
            (FileSource
                (HostName 'frigg.uio.no')
                (FullFileName '/etc/passwd')
            )
        )
        (OpenApplicationSession
            (World Unix)
            (Outcome
                (CIDFReturnCode failed)
                (Comment '/etc/passwd missing')
            )
            (When
                (Time 15:02:48 24 Feb 2005)
            )
            (Initiator
                (HostName 'hostb.uib.no')
            )
            (Account
                (UserName 'ksimpson')
                (RealName 'Karen Simpson')
                (HostName 'frigg.uio.no')
            )
            (Receiver
                (StandardTCPPort 22)
            )
        )
    )
```

Figure 8: CISL sentence syntax example

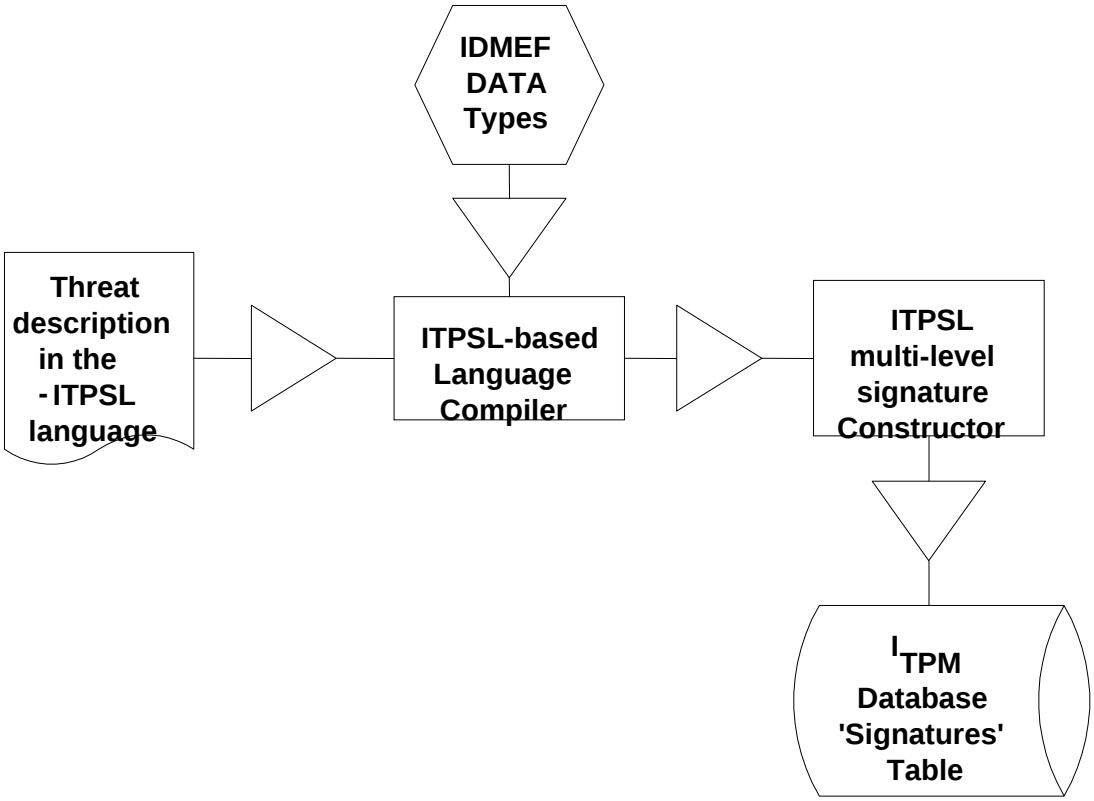

Figure 9: From ITPSL text description to a threat signature

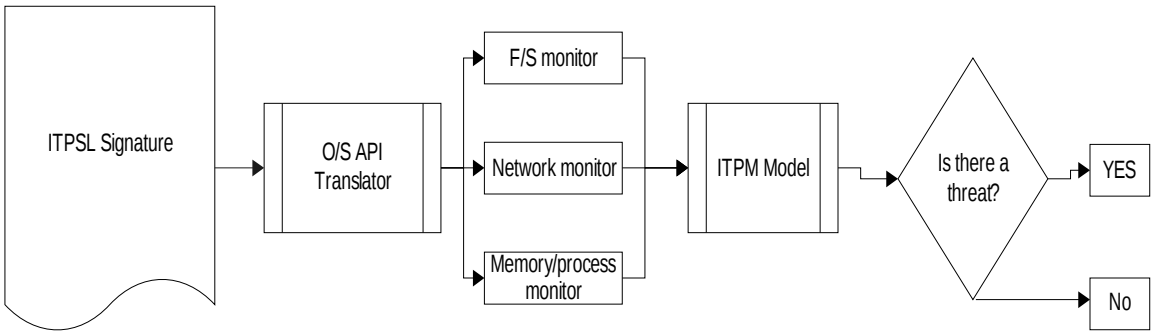

Figure 10: ITPSL/ITPM relationship