A Preliminary Model of End User Sophistication for Insider Threat Prediction in IT Systems

G.B.Magklaras and S.M.Furnell

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

nrg@plymouth.ac.uk

Abstract: - The dangers that originate from acts of IT system misuse by legitimate users constitute a separate category of threats with well documented consequences for the integrity, privacy and availability of computer systems and networks. Amongst the various properties of malicious legitimate users one of the most notable ones is the level of his/her sophistication. Various studies indicate that user sophistication and the potential to misuse IT systems are properties that are strongly related. This paper presents a methodology that automates the process of gauging end-user sophistication. The establishment of suitable metrics to characterize End-User Sophistication is discussed followed by an experimental verification of the metrics on a sample of 60 legitimate users, using the UNIX Operating System. The results indicate that a combination of application execution audits and computational resource utilization metrics could be used to characterize the level of IT sophistication of an end-user. Although additional testing in a greater variety of computational environments is required in order to validate the derived preliminary scheme, it is considered that the derived methodology could serve as a component of experimental Insider Threat Prediction processes, or any other model that requires a procedure to measure the level of IT knowledge of a legitimate user base.

Keywords: insider misuse, insider threat, user sophistication modeling, insider profiling, user capability measurement

Introduction

Information Security professionals often emphasize the increasing dependency of our modern society on Information Technology systems. Air traffic, telecommunications, defense, energy and water distribution systems are all typical examples of mission critical infrastructures that are controlled by IT systems. They also believe that amongst the various attacks that may potentially target these systems, those originating from 'insiders' may have serious consequences for the proper functioning of computer systems and networks.

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. The word 'legitimately' 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 within 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. It also means that an insider is less likely to get caught by implemented security measures because of the level of trust that he/she enjoys. These aspects make the problem of tackling insider IT misuse a composite and difficult one.

This paper considers the sophistication of an end user as a potential factor that influences their capability to comment insider misuse. The next section presents an overview of the problem posed by insiders, contrasting the impacts of their activities against those of external attackers, and briefly examining how technical abilities played a role in some reported incidents. The discussion then proceeds to more formally consider end user sophistication as an insider misuse threat factor, before proposing a means of modeling it in practice. The proposed model is then evaluated in the context of an experimental study, testing the concepts on a small user population and a pre-defined number of computer applications. The paper concludes with consideration of the implications and limitations of the findings.

The Insider IT Misuse problem

The British Department of Trade and Industry (DTI) in association with PriceWaterhouseCoopers (PWC) published the ‘information security breaches survey 2004’ [1]. The survey mentions that Insider Misuse has doubled since the year 2002, mainly driven by the increased adoption of World Wide Web and Internet related technologies. Approximately a third of the DTI/PWC 2004 respondents claimed that their worst security incident was internal. This verifies the existence of internal security threats.

Figure 3.1 displays the distribution of the DTI/PWC 2004 worst security incidents for small, medium and large organizations. Whilst the smaller IT infrastructures appear to face more incidents of external origin, the gap between insider and outsider incidents is smaller for respondents of medium and large scale organizations. This indicates that the likelihood of IT misuse from legitimate users is a very probable scenario.

Figure 3.2 depicts the types of misuse reported by UK businesses [1], showing that the misuse of World Wide Web and email facilities are the most frequent activities. Excessive usage of these facilities for personal purposes, as well as for viewing and disseminating inappropriate material, were considered by the DTI/PWC survey as misuse incidents for web and email facilities. The category of ‘Unauthorized Access to Systems or Data’ included legitimate users’ attempts to obtain another user’s password, and finally the ‘Infringement of Laws and Regulations’ included violations of the Data Protection or Computer Misuse Acts.

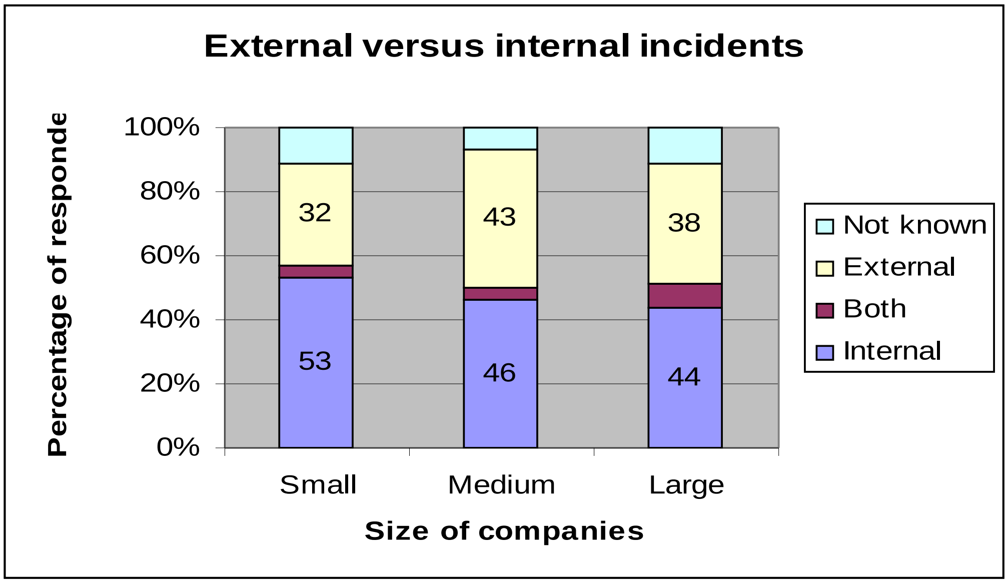

Figure 1: "Cause of the worst security incident", source [1]

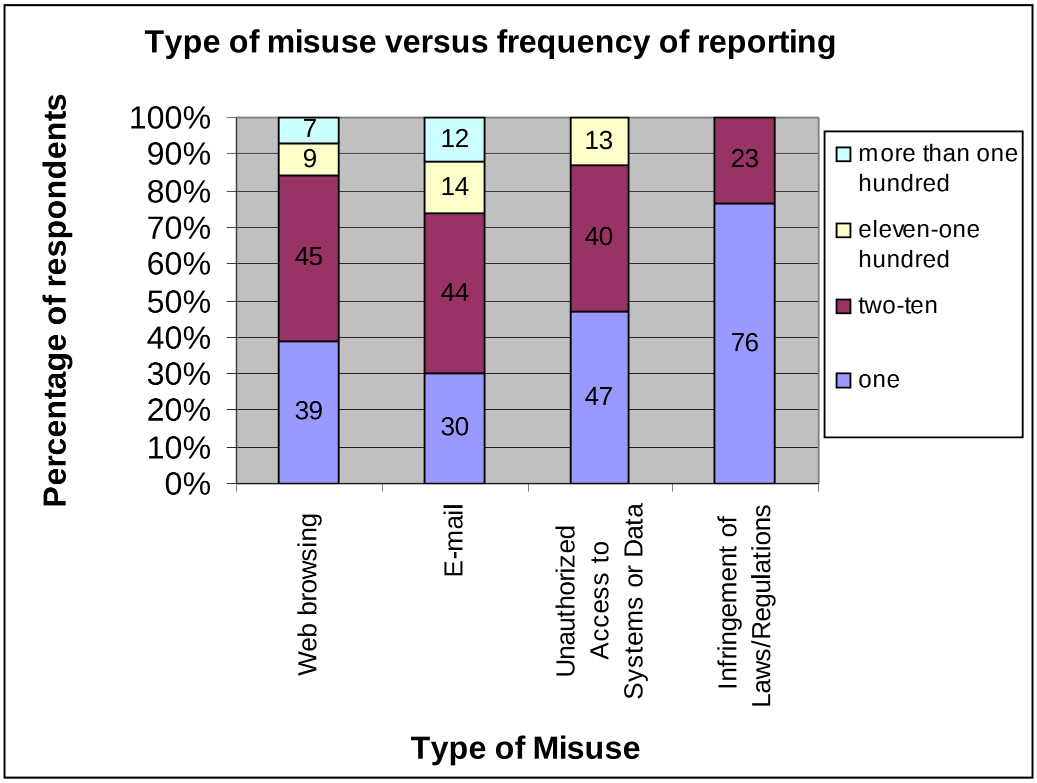

Figure 2: Types of misuse reported by UK businesses [1]

The severity of insider incidents should not only be judged by their reported frequency of occurrence. The financial cost of a security breach should also be taken into consideration. The 2003 survey from the US-based Computer Security Institute (CSI) [2] is another source that emphasizes the presence of insider threats. The notable thing about the survey is the table with title ‘The cost of Computer Crime’, where one can view aggregate costs sampled over a 48 month period (2000-2003). The table quotes incident categories such as “System Penetration by Outsider”, “Insider abuse of Net access” and “Unauthorised insider access”. These categories relate clearly the cost of security incidents to external or internal origins. In contrast, the rest of the incident categories could be attributed to both internal and external origins. This fact combined with the small percentage of the survey respondents that were able to quantify their losses (just 47% for 2003) makes the comparison between internal and external incidents unfeasible.

Computer crime surveys are not the only indicators of the presence of insider threats. There are also widely cited case studies that reveal the seriousness of insider IT misuse cases. For example, CIO magazine [3] describes the case of John Michael Sullivan, a former employee of Lance Incorporation. Sullivan was demoted and eventually resigned from the company. Several months after his resignation, a server designed to exchange data with the handheld devices of Lance’s sales team had suddenly lost a great amount of valuable data and became inoperable. An investigation by Computer Crime Specialists revealed that the server outage was due to a logic bomb that Sullivan had skillfully implanted into the hand-held device data distribution server. The logic bomb was set to execute on the anniversary of Sullivan’s hiring date and cost his employer an estimated loss of 1 million US dollars, due to lost sales and equipment recovery costs.

Whilst the previous case reveals the potential consequences of insider misuse in terms of lost revenue, there are also cases that relate insider IT misuse to National Security issues. One of the highest profile cases that support this relation is that of Robert Hanssen [4]. Hanssen, a highly regarded FBI agent turned out to be a mole, selling confidential information to Russian Intelligent Services. He was not only highly regarded amongst his FBI colleagues but he was also very knowledgeable with regards to Information Technology. Hanssen was an expert in the process of hiding the information he unlawfully acquired, by employing specially formatted floppy disks [5]. At a first glance, the floppy disks appeared to be empty or containing legitimate information [5]. However a closer examination by FBI Computer Forensics Specialists revealed they contained the illegitimate material that Hanssen had hidden in the very last track of 40-track formatted storage medium. This track was not employed to store filesystem data by convention and hence the technique was a sophisticated way to hide information.

Although the previously mentioned cases refer to different sides of the insider IT misuse spectrum, they have two common properties. The first is concerned with the level of trust that the malicious insiders enjoyed. This level of trust provides the necessary privileges to ease the task of misusing IT infrastructures from an internal point. However, trust and its resulting privileges are not a panacea for the completion of a successful insider attack. It is reasonable to assume that Sullivan was not the only IT engineer in Lance Incorporation. If he was not skilled in the technique of implanting code in a live production system used and administered by other IT engineers, his attack might have been unsuccessful. The same conclusions could be drawn for the case of Robert Hanssen. Consequently, one preliminary assumption is that the level of IT sophistication of a legitimate user can be considered as an important indicator of the misuse threat.

The following section will further emphasize the relation of End User Sophistication and insider threat, and will also discuss research and development efforts that address the mitigation of the insider misuse problem.

End User Sophistication as an Insider Misuse Threat factor

Measuring End User Sophistication is part of a broader research initiative aimed to model insider threats. An insider threat model is a new idea that aims to mitigate effectively insider threats by aiding the process of detecting, and if possible predicting, a particular range of threats from legitimate users.

Parker [6] has established a general model of computer crime attacks based on factors such as the IT skills and knowledge of the attacker, as well as other parameters such as the resources, authority and motives for performing the attack. However, Parker's approach is quite generic as it addresses both external and internal incidents. Wood [7] offers a more insider-specific approach for qualifying a set of metrics to mitigate insider threat. Amongst a range of malicious insider qualification criteria, Wood suggests that a malicious insider can be qualified in terms of two attributes:

- **Knowledge:** The legitimate user is familiar with some or all the internal workings of the target systems, or has the ability to obtain that knowledge without arousing suspicion.
- **Skills:** The knowledgeable insider will always have the skills to mount an attack that is usually limited to systems that he/she is very familiar with. The model assumes that a given adversary is unlikely to attack unfamiliar targets.

The role of user sophistication as an important insider threat factor is also emphasized by Neumann [8]. The paper mentions how the level of technical knowledge and sophistication of legitimate users can act as a deterrent for the process of detecting and mitigating insider threats, even when compartmentalized Operating Systems are employed in an IT infrastructure.

Similar conclusions are drawn by Magklaras and Furnell [9], based upon a number of real world case studies of malicious insiders that abused their level of trust and knowledge skills and misused IT infrastructures. The paper presents a preliminary Insider Threat Prediction Model (ITPM), the core mechanism of which consists of a hierarchy of Insider Threat Misuse estimation functions. The goal of these functions is to gauge certain end user properties such as:

- Legitimate User Attributes: The documented professional role and the IT infrastructure access privileges of the user are examined and related to potential levels of threat. The User Attributes function awards more points to critical roles and legitimate users with great levels of IT infrastructure access (both in terms of file contents, application execution as well as physical access to the equipment).
- Legitimate User Behavior: The operations on file and application content, the level of legitimate user sophistication in terms of his/her IT knowledge and the network I/O operations of the user are evaluated using suitable metrics.

$$\boldsymbol{EPT} = \sum \boldsymbol{F}_{threat\ components} \Rightarrow$$

$$\boldsymbol{EPT} = \boldsymbol{F}_{accessrights} + \boldsymbol{F}_{behavior} \Rightarrow$$

$$\Rightarrow \quad \boldsymbol{EPT} = \boldsymbol{C}_{role} + \boldsymbol{C}_{criticalfiles} + \boldsymbol{C}_{hardware} + \boldsymbol{C}_{utilities} + \boldsymbol{C}_{sysadm} + \boldsymbol{F}_{behavior} \quad (1)$$

$$\boldsymbol{EPT} = \boldsymbol{C}_{role} + \boldsymbol{C}_{criticalfiles} + \boldsymbol{C}_{hardware} + \boldsymbol{C}_{utilities} + \boldsymbol{C}_{sysadm} + \boldsymbol{F}_{sophistication} + \boldsymbol{F}_{fileops} + \boldsymbol{F}_{execops} + \boldsymbol{F}_{network}$$

Equation (1) provides a revised overview of the proposed ITPM model described in [9]. Whilst it is outside the scope of this paper to discuss the details of the fundamental design philosophy of the framework, the displayed metrics examine a range of issues associated to the users role and level of access ($F_{accessrights}$), as well as a number of metrics that focus on their on-line actions ($F_{behavior}$). The legitimate user's documented role inside an organisation ($\boldsymbol{C}_{role}$), whether he has access to system administration utilities ($\boldsymbol{C}_{sysadm}$), commercially sensitive files ($\boldsymbol{C}_{criticalfiles}$) , critical application utilities ($\boldsymbol{C}_{utilities}$) and physical access to central IT infrastructure components ($\boldsymbol{C}_{hardware}$) are the concerns of the $F_{accessrights}$ function.

In contrast, $F_{behavior}$ examines user parameters related to behavioral characteristics. $F_{sophistication}$ tries to gauge the level of IT knowledge of the legitimate user. $F_{fileops}$ is looking at file access patterns, whereas $F_{execops}$ represents a way to examine the sequence of user actions, in order to decide whether the way a legitimate user executes applications resembles any known ways of misusing a computer system. Lastly, $F_{network}$ searches for insider threat signs by examining the traffic patterns the user generates at the network level.

The derived Evaluated Potential Threat (EPT) is an arithmetic value proportional to the probability of a legitimate user misusing the IT infrastructure. As a result, EPT will range from 0-100 points, representing a misuse probability of 0 to 1. These functions contain a series of constants and further subfunctions as shown by (1).

Each EPT component has a certain weight which defines its arithmetic contribution to the final EPT value. One could then form a weight matrix such as the one shown below:

$$(6,6,6,6,6,12,18,18,20)=(C_{role},C_{data},C_{hardware},C_{sysadm},C_{utilities},F_{sophistication},F_{fileops},F_{execops},F_{netops})$$

A system administrator/security specialist can re-define the weight matric, in order to reward a particular metric that he trusts more than the others. Consequently, $F_{sophistication}$ is a sub-function of $F_{behavior}$, contributing an overall arithmetic weight to EPT. However, all the experiments described in this paper use the aforementioned Weight Matrix values. Thus, $F_{sophistication}$ contributes a maximum of 12 points to the EPT value.

An alternative framework for insider threat prediction has also been proposed in [10], and identifies factors such as the personality traits and verbal behavior of the insiders as being amongst the potential indicators that could be used to identify attacks. However, although this is conceptual feasible, it is difficult to see how such factors could be reliably measured and assessed in practice.

It requires means, opportunity and motive to maliciously misuse a system. However, the framework by Magklaras and Furnell [9] concentrates on the 'means' and 'opportunity' factors. A legitimate user's motive must never be underestimated as a factor of a threat prediction process, but in practice, due to its human-centric nature it introduces certain difficulties. For example, if a manager has reasons to believe that an employee is disgruntled or depressed, he could used [9] to assess the risk that the insider might pose to the system, if their feelings became a motive for misuse. This approach might work well for small scale organizations, where managers might easily be aware of peculiar employee personality trends. On the other hand, the method might be problematic when applied to larger corporations where direct contact amongst employee groups is limited.

As such, the 'motive' factor has been excluded from [9] and the factors contributing to the EPT value above are considered to offer more potential as tangible metrics, which could be measured within a live system.

The remainder of this paper will focus upon the discussion of a suitable way to measure the end-user sophistication, which is a necessary part of an insider threat modeling process. The next section outlines the steps of a procedure to measure end user sophistication, and thus realize the $F_{sophistication}$ function as part of the Inside Threat Prediction Model.

The proposed End User Sophistication Model

The idea of modeling end user sophistication is not a new one. Evans and Simkin [11] have produced early studies on measuring sophistication, amongst Computing Professionals and Computer Science students. Their study tried to identify how competence in Computer Programming can be correlated to factors such as age, gender and a range of other individual differences. However, their effort focused only upon computer professionals. A generic End-User Sophistication model should address a much broader user base, not only professionals and students of the IT field. Nevertheless, Evans and Simkin were one of the first to consider technical aptitude (in this case computer programming ability) as an End-User Sophistication parameter.

Huff et al [12] have attempted to address the restriction of the Evans and Simkin and deduced a more generic model of end user sophistication. Their paper discusses how end user sophistication could be evaluated for the purposes of increasing the efficiency of human resource management inside an organization. Huff’s research team conducted interviews of 31 employees from eight different organizations. The interviews had a semi-structured nature, asking the subjects to complete short questionnaires and talk about their experience of particular IT issues. The results were collected and analyzed by the authors and an additional panel of Computer Science Academics.

The result of this analysis was the formulation of an ‘End User Computing (EUC)’ sophistication model that classified users in terms of three important attributes:

- **Breadth of knowledge**: Their findings indicate that advanced users were able to employ a greater variety of IT tools than intermediate or novice ones.
- **Depth of knowledge**: The level of mastery of a particular IT sub-domain or application (gained either by extensive training or hands-on experience) is proportional to the level of user sophistication.
- **Finesse**: The ability of a user to solve particular IT problems in efficient and innovative ways, given a certain level of breadth and depth capability is also an end-user sophistication classification metric.

The authors do not provide a structured methodology of how exactly they measured the ‘finesse’ attributes of users. Although the way (tools and their combination) of solving a series of problems is a reasonable metric of the end user abilities, it would be difficult to devise standardized tests for an automated algorithm on a live system. Consequently, someone may focus on the breadth and depth dimensions of EUC sophistication.

In order to devise a metric for measuring the breadth of knowledge, if n represents the number of unique applications executed by a particular user per session, and c the number of sampled user sessions, then:

$$avdiffapps = \Sigma n_i / c \ , (i=1 \text{->} i=c) \qquad (2)$$

The target is to find the variety of the application/command vocabulary of the user. What therefore counts for our purposes is how many different applications are used in each session. For example, if a user has three instances of Netscape running, only one application is counted. That number is then averaged over the number of sessions recorded (in our case 20 for each user) in order to enable a general profile of the user’s application set to be established. Provided that enough sessions are sampled, a good picture of the typical applications accessed can still be obtained. As a result, a basic requirement for (2) dictates that a session to exceed a certain minimum duration (i.e. to make sure that the profiling is not inadvertently based upon 20 sessions in which the user simply logged in briefly, checked mail, and left again).

This scheme will reward more points to users that execute on average a greater variety of tools. In order to dimension the *avdiffapps* values to fit in the proposed scales of the ITPM scoring scheme, it is necessary to consider the average values of *avdiffapps* for each user category. We divide the users in three levels with regards to their End User Sophistication:

- Advanced: Users that clearly exhibit a high level of sophistication, that indicates mastery of applications or system internals.
- Ordinary: Users that have an intermediate level of knowledge of certain applications.
- Novice: Users that clearly know very little about the IT infrastructure (software and hardware).

If μ represents the arithmetic average of *avdiffaps* for every user category, and c a pre-defined scoring constant associated to a particular user category, then:

$$\begin{gathered}
F_{breadth} = c_{high},\ if\ \mu_{ordinary} < x \\
F_{breadth} = c_{medium},\ if\ \mu_{novice} < x \le \mu_{ordinary} \\
F_{breadth} = c_{low},\ if\ 0 < x \le \mu_{novice}, \\
\textbf{where } c_{high} > c_{medium} > c_{low}
\end{gathered} \tag{3}$$

Huff et al claim that “depth capability has much to do with mastery of the features and functions of different types of application systems, practices, techniques etc” [12]. In order to inspect these parameters on a working system, one has to devise mechanisms for checking:

I) The type of applications utilized on average and rate them in terms of the level of system knowledge they require in order to be used.

II) The way each of these applications is called and used, by considering issues such as means of execution (scripted versus manual) and system resource impact.

In order to realize the requirements of mechanism I, one has to define a one-to-one association between an application program and a score that indicates the level of system knowledge required to use this particular application. The greater the knowledge required, the greater the score. Thus, applications could be classified in three broad categories: Applications requiring advanced knowledge of the system (system masters) scoring a total of 3 points, applications that indicate advanced knowledge of the system that worth 1.5 points and finally applications that require the absolute minimum level of sophistication for 0.75 points. Then, the arithmetic average of all the sampled application scores of a particular user could serve as a suitable quantification mechanism.

The reasoning behind the scoring method is that there should be a sufficient value gap amongst applications that are perceived to match the needs of system masters versus those of advanced users and those of ordinary users (e.g. you cannot assume that someone who frequently uses compilers or assemblers has the same amount of knowledge as someone that uses word processors). So, the value of a system master would be 4 times the value of ordinary users and twice that of advanced users, doubling as we go towards higher user categories.

Hence, if $F_{appscore}$ indicates a function designed to gauge the level of sophistication for a particular user in terms of the type of applications he invokes, then:

$$\boldsymbol{F_{appscore}=Score_{app1}+Score_{app2}+Score_{app3}+\ldots+Score_{appn} / n} \quad (4)$$

where n=number of recorded used applications for a user

In order to address the requirements of mechanism II, the function $F_{resutil}$ was devised. It represents the arithmetic sum of three computational resource consumption indicators.

$$\boldsymbol{F_{resutil}=S_{CPU}+S_{RAM}+S_{SIMAPPS}} \quad (5)$$

S_{CPU}, S_{RAM} and $S_{SIMAPPS}$ represent the scores allocated for the measured CPU, RAM and simultaneous applications metrics. The first two of the metrics represent the average percentage of consumption of CPU and RAM by the user, whereas the third one records how many applications the user employed at the same time (assuming that, on average, sophisticated users would utilise more applications at the same time than the less sophisticated users).

The End User Sophistication model can now be summarised by the following formula:

$$\boldsymbol{F_{sophistication} = F_{breadth} + F_{appscore} + F_{resutil}} \quad (6)$$

Experimental verification of the proposed model

In an attempt to verify the proposed modeling scheme, this section presents the numeric results of an experiment that monitored 60 UNIX users in the Norwegian National EMBnet Node, a scientific establishment located at the University of Oslo in Norway, over a period of four months. The sample contained three categories of users that were pre-classified in terms of their documented professional role and experience. Hence, in full accordance with the proposed user categories of the previous section, the sample included:

- Advanced users: Includes system administrators and scientific personnel with substantial programming knowledge (software engineers, computer science and bioinformatics academic personnel) that have been users of the system for more than two years.
- Ordinary users: Scientists that had been using the server facilities for a minimum of 12 and a maximum of 24 months.
- Novices: Students who have recently attended an introductory course for using the UNIX system, or users that have been using the system for less than twelve months.

The participants employed a series of generic applications, such as email and word processing programs, as well as specialized bioinformatics utilities such as the EMBOSS application suite [13], BLAST [14] and a variety of programming language interpreters and compilers. A total of 20 ‘sessions’ per user were employed to collect the amount of data. The term ‘session’ refers to all the commands and system resource impact indicators collected from the moment a user logs in until the time he/she logs out. This includes data from multiple user shell sessions. There were an equal number of participants from all categories.

All the results were collected by examining Operating System shell-level commands on a single UNIX server, as well as system resource utilization metrics obtained by a series of PERL [15] scripts. Amongst other utilities, these scripts employed the UNIX ps command-line utility [16], in order to collect the average CPU and RAM utilization metrics and a suitable command logging system based on the execve system call [17]. The latter is necessary in order to reliably record all the applications executed by a particular user, together with their associated arguments.

At a first stage, the efficiency of each of the individual computational resource utilization indicators ($F_{resutil}$) was gauged. The first important conclusion was that the level of user sophistication was proportional to the CPU and RAM utilisation. Advanced users on average consumed approximately three times more CPU and RAM than ordinary users. Advanced users also appeared to consume on average approximately ten times more of these resources than the novice users. Figures 3 and 4 illustrate the distribution of values for these two metrics for all user categories.

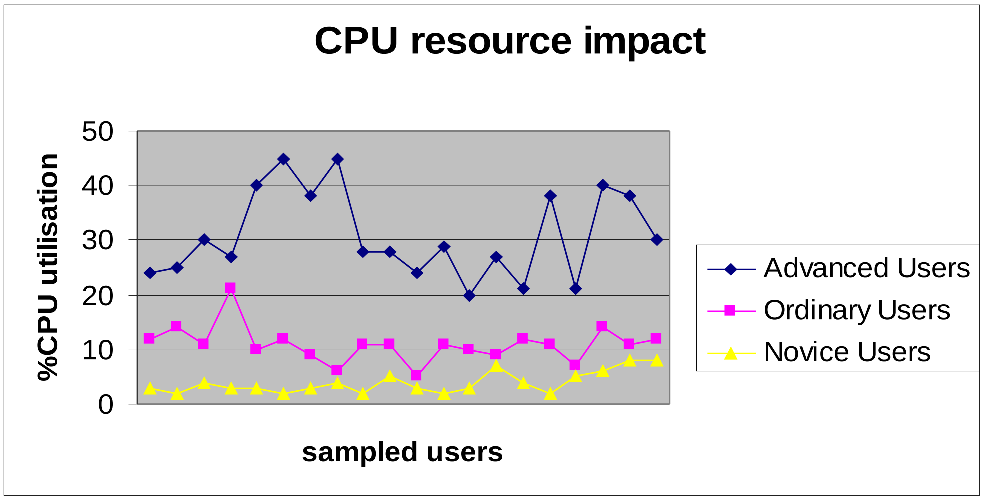

Figure 3: S_{CPU} User Sophistication Classification Results

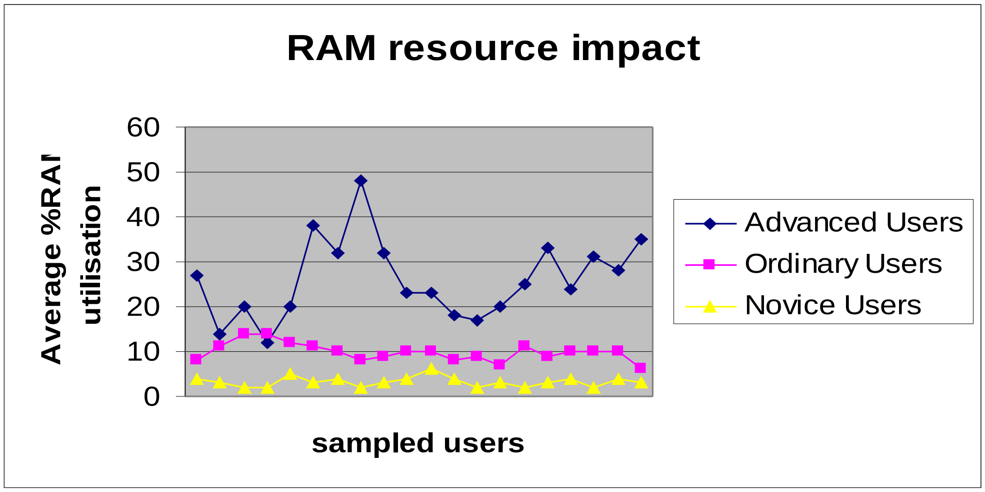

Figure 4: S_{RAM} User Sophistication Classification results

The same conclusions could be deducted by looking into the number of applications used simultaneously (per user session) for the three user categories. In particular, the most sophisticated users employed on average twice as many simultaneous applications as the ordinary users and four times the average amount of simultaneous applications of novice users. Figure 5 summarizes these findings.

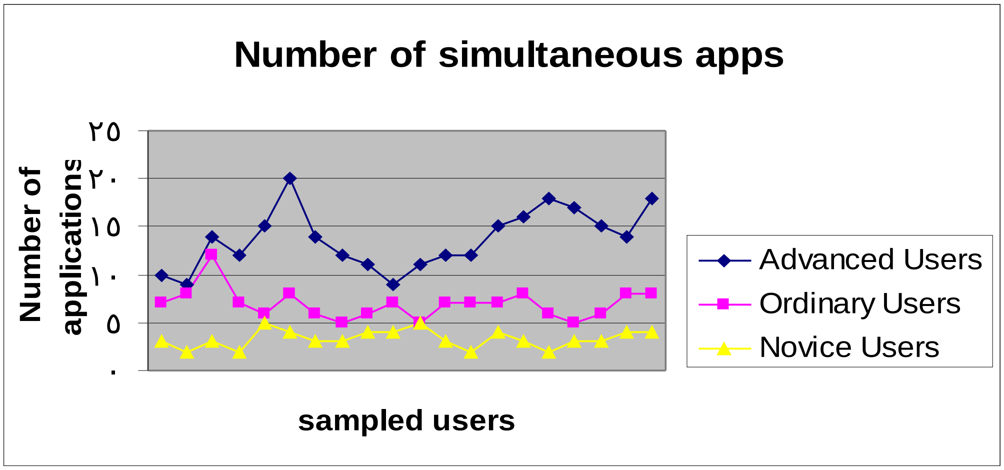

Figure 5: $S_{SIMAPPS}$ User Sophistication Classification Results

$$S_{CPU} = 1, \text{ if } (\mu_{ordinary} < x \le \mu_{advanced}) \text{ OR } (\mu_{ordinary} < x \text{ AND } x \ge \mu_{advanced})$$

$$S_{CPU} =0.5, \text{ if } \mu_{novice} < x \le \mu_{ordinary}$$

$$S_{CPU} = 0.1, \text{ if } 0 < x \le \mu_{novice}$$

$$S_{RAM} = 1, \text{ if } (\mu_{ordinary} < x \le \mu_{advanced}) \text{ OR } (\mu_{ordinary} < x \text{ AND } x \ge \mu_{advanced})$$

$$S_{RAM} = 0.5, \text{ if } \mu_{novice} < x \le \mu_{ordinary} \qquad (7)$$

$$S_{RAM} = 0.1, \text{ if } 0 < x \le \mu_{novice}$$

$$S_{SIMAPPS} = 1, \text{ if } (\mu_{ordinary} < x \le \mu_{advanced}) \text{ OR } (\mu_{ordinary} < x \text{ AND } x \ge \mu_{advanced})$$

$$S_{SIMAPPS} = 0.5, \text{ if } \mu_{novice} < x \le \mu_{ordinary}$$

$$S_{SIMAPPS} = 0.1, \text{ if } 0 < x \le \mu_{novice}$$

However, the $F_{resutil}$ metrics indicate clearly an undesirable overlap amongst the different user categories. For instance, the RAM resource impact graph indicates substantial overlap between the Advanced and the Ordinary user category. Despite indicating a general trend that relates the level of user sophistication to the value of computational resource consumption, the S_{CPU}, S_{RAM} and $S_{SIMAPPS}$ metrics could easily misclassify users. This overlap represents the weak point of each one of these metrics. In order to overcome this problem, all metrics of the sophistication model are combined as described in formula (6). In addition, a refinement of the $F_{resutil}$ indicators is also necessary. For each of these score variables, if μ represents the arithmetic average of each metric for every user category, and x the recorded value of a metric per user, as shown in equation set (7).

In addition, with regards to the $F_{breadth}$ equation (3), the arithmetic values of 6, 3 and 1 have been chosen respectively for the constants c_{high}, c_{medium}, and c_{low}. This value scheme represents a c_{high} to c_{low} ratio of 6 to 1 and a c_{medium} to c_{low} ratio of 3 to 1. The purpose behind the selection of this ratio scheme was the reduction of the overlap amongst the user categories. The selected ratios also emphasize the importance of the $F_{breadth}$ metric, by rewarding more sophistication points to end users that use a wider range of applications.

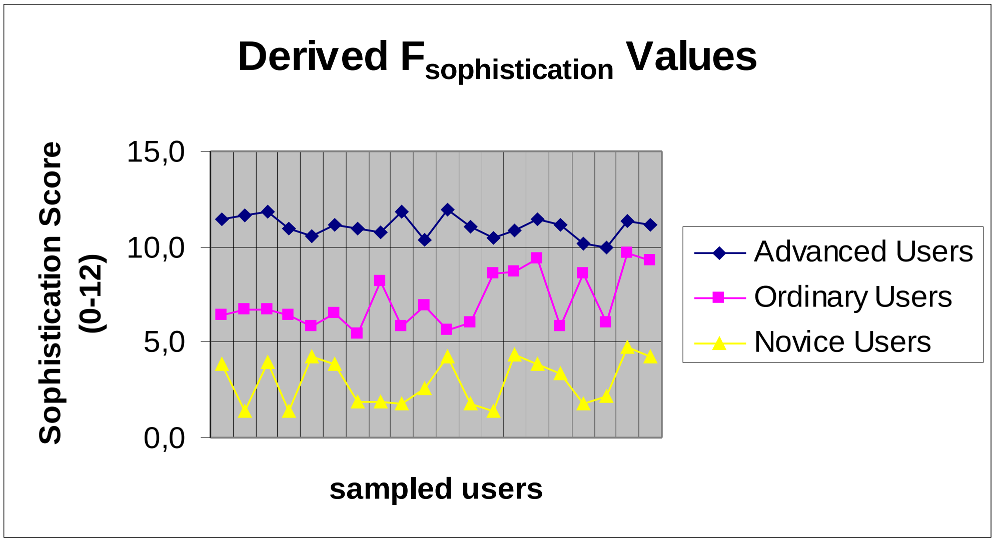

Figure 6: End-User Sophistication Model Experimental results

The end result of these adjustments is shown in the graph of Figure 6. One can now observe more clearly the borders of distinction amongst the different user categories. Hence, for a particular computational environment (Operating System plus a number of applications), user sophistication thresholds values could be established, in order to classify the technical sophistication of end-users. In this particular case, the experimental results indicated that advanced users achieve an $F_{sophistication}$ score that ranges from 10 to 11.9 units, ordinary users are placed in the range of 5.4 to 9.7 and novice users scores were measured in the range of 1.4 to 4.7 points. Consequently, the following rule could be applied to classify users:

If $F_{sophistication}$ >=10 => Advanced User

If $F_{sophistication}$<=9.7 AND $F_{sophistication}$ >= 5.4 => Ordinary User

If $F_{sophistication}$<=4.7 => Novice User

This leaves the question of how to handle the border-line cases. For example, if a particular user measures 9.8, the previous rules cannot indicate whether he should be classed as an ordinary or an advanced user. The suggested approach to resolve these cases is to calculate the half of the difference between the minimum value of the arithmetically higher category and the maximum value of the lower category. That value should then be added to the maximum value of the lower category to calculate a borderline threshold value. If the measured sophistication value for the user is greater than the borderline threshold value, the user should belong to the higher category. If it is lower, then the user is classed in the lower user group. In the exceptional case where the user's $F_{sophistication}$ value is equal to the borderline threshold value, the user is classed as a borderline case and the calculations have to be repeated in order to get safely classify the user. So, for example, the aforementioned value of 9.8 would place the user between the advanced and the ordinary user category. $F_{advancedMIN}$=10 and $F_{ordinaryMAX}$=9.7. Thus, the calculated borderline threshold value is 9.85. 9.8 is below 9.85, hence in this case the user is classified as an ordinary one.

The results from the experimental study suggest that the proposed technique has the potential to classify end-user sophistication in an effective manner. Although the findings at this stage are only based upon a relatively small end-user population (and thus on one level have the potential to be perceived as situational), they are encouraging from the perspective that the use of IT and the software within the target environment was not unusual. As such, there is no obvious reason to believe that the techniques could not be applied within other environments, with the own range of systems.

Conclusions

This paper argued that there is a certain relation between insider misuse cases and the level of the end-user sophistication. It has also described a set of metrics and their combination into a set of formulae, in order to devise a suitable end-user sophistication model. Consequently, a methodology could be derived that would allow an automated process to classify users in terms of their level of sophistication. The first step of this methodology can be achieved by selecting a user sample which contains an equal number of users from each sophistication level. The next step involves the process of training the model by measuring repeatedly the metrics for people of the same category, in order to establish minimum and maximum $F_{sophistication}$ values for each user sophistication level. These values can then be used for subsequent measurements of new users, in order to gauge their level of sophistication.

This model can then be used as a component of a wider Insider Threat Prediction Model. The classification result could be used to assign a probabilistic threat weight to the behavior of an end user. This weight is indicative of the probability of the user misusing the system, and proportional to the calculated $F_{sophistication}$ value. The weights value could be scaled up or down, depending upon how much emphasis the model designer wishes to place upon the use of End-User Sophistication as an Insider Threat Prediction metric.

In addition, the model's resource utilization metrics (RAM, CPU and number of instances of a particular application), as well as the user application execution data (user commands and their associated data), could be provided and reused by a variety of other system-level tools. This should minimize the amount of software development required for realizing the proposed approach and attributes a modular character to the model, so that it can be integrated easily to other information security tools.

It should be emphasized that this model requires a successful selection of a user sample for training. If the initial sample user categorization according to the user's documented role and experience is false, the model will yield inaccurate results. Therefore, the entire procedure requires intervention from experts for the purposes of validating the training user sample. Moreover, the model refers to specific computational environments with a certain set of Operating System commands and user-space applications. If new applications are installed on the target system, this model would require re-sampling of the training values, in order to function correctly. These two limitations make the model inflexible for today's evolving IT infrastructures and they certainly need to be addressed by future work in the field.

Nevertheless, the model represents a novel experimental approach that could not only provide a metric for Insider Threat Prediction process, but which could also be useful for people concerned with the automatic customization of Human Computer Interaction (HCI) interfaces, or people that would like to estimate the productivity potential of their computing users.

References

[1] PriceWaterhouseCoopers Internet portal (2004), "Information Security Breaches Survey 2004 – Technical Report" http://www.pwc.com/images/gx/eng/about/svcs/grms/2004Technical_Report.pdf

[2] Richardson, R. 2003. *2003 CSI/FBI Computer Crime and Security Survey*. Computer Security Institute. Spring 2003.

[3] Scarlet S.D. "Dr. Crime's Terminal of Doom and Other Tales of Betrayal, Sabotage and Skullduggery", *CIO magazine*, June 2002. http://www.cio.com/archive/060102/doom.html

[4] CNN.com, "The case against Robert Hanssen", In-Depth Special series. http://edition.cnn.com/SPECIALS/2001/hanssen/

[5] Slashdot.org, "Spying and Technology: Robert Philip Hanssen". http://slashdot.org/articles/01/02/22/0622249.shtml

[6] Parker D.B. 1998. *Fighting Computer Crime: A new framework for protecting information*, John Wiley and Sons, New York.

[7] 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.

[8] Neumann P. 1999. "The Challenges of Insider Misuse", SRI Computer Science Laboratory, Paper prepared for the Workshop on Preventing, Detecting, and Responding to Malicious Insider Misuse, 16-18 August 1999.

[9] Magklaras G., Furnell S. 2002. "Insider Threat Prediction Tool: Evaluating the probability of IT misuse", *Computers and Security*, vol. 21, no. 1, pp. 62-73.

[10] Schultz, E.E. 2002. "A framework for understanding and predicting insider attacks", *Computers & Security*, vol. 21, no. 6, pp. 526-531.

[11] Evans, G., Simkin, M. 1989. "What Best predicts Computer Proficiency?" *Communications of the ACM* (32:11), November 1989, pages 1322-1327.

[12] Huff S., Munro M., Marcolin B. 1992. "Modeling and measuring End User Sophistication", University of Western Ontario, Canada, Paper Published on the 1992 ACM Proceedings, ACM 089791-501-1/92/0004/0001

[13] EMBOSS.org portal, The European Molecular Biology Open Software Suite. http:// www.emboss.org

[14] The Basic Local Alignment Search Tool (BLAST). http://www.ncbi.nlm.nih.gov/Education/BLASTinfo/references.html

[15] The Practical Extraction and Report Language (PERL). http://www.perldoc.com/perl5.6/pod/perl.html

[16] Nemeth E., Snyder G., Sybass S., Hein T. 2000. *Unix System Administration Handbook*, Prentice Hall, ISBN 0130206016.

[17] Johnson M., Troan E. 1999. *Linux Application Development*, Addison Wesley, ISBN 0201308215 (pages 113-115 describe the execve system call).