On the Design and Execution of Cyber-Security User Studies: Methodology, Challenges, and Lessons Learned

Malek Ben Salem and Salvatore J. Stolfo
Computer Science Department
Columbia University
New York, New York, USA
{malek,sal}@cs.columbia.edu

***Abstract*—Real-world data collection poses an important challenge in the security field. Insider and masquerader attack data collection poses even a greater challenge. Very few organizations acknowledge such breaches because of liability concerns and potential implications on their market value. This caused the scarcity of real-world data sets that could be used to study insider and masquerader attacks. Moreover, user studies conducted to collect such data lack rigor in their design and execution. In this paper, we present the methodology followed to conduct a user study and build a data set for evaluating masquerade attack detection techniques. We discuss the design, technical, and procedural challenges encountered during our own masquerade data gathering project, and share some of the lessons learned from this several-year project.**

I. Introduction

Lack of large-scale, real-world data has hindered the development of effective intrusion detection systems for the detection of insider attacks. Most organizations that undergo such types of attacks prefer not to announce them publicly out of liability and confidentiality concerns. According to the 2010 cyber crime watch survey, which was conducted by the Computer Emergency Response Team (CERT), and which surveyed 523 security executives and law enforcement officials, 72% of the insider incidents that occurred at the surveyed isntituions were handled internally without legal action or the involvement of law enforcement [6]. Another 13% of the insider incidents are handled internally with some legal action. Announcing such attacks may also have market share implications. For the same reasons, they are even less likely to share real-world data that could be used to study such attacks with the research community.

The study of masquerade attacks, a class of insider attacks in which a user of a system illegitimately poses as, or assumes the identity of another legitimate user, suffers similarly from the scarcity of real-world data, despite their significance. According to the 2010 cyber crime watch survey [6], 35% of the surveyed executives and law enforcement officials experienced unauthorized access and use of their information, systems, and networks. This type of intrusion, known as *a masquerade attack*, was second in the top five list of electronic crimes perpetrated by outsiders after virus, worms and other malicious code attacks.

In the absence of a real-world data set for the study of masquerade attacks, we had to launch our own data collection project. In this paper we present the procedure followed to gather our own data set for the evaluation of masquerade attack detection techniques. We describe the methodology for conducting a user study where we collected masquerader data and tested a hypothesis that malicious masqueraders exhibit a different search behavior than that of normal legitimate users. We discuss the challenges encountered during the collection and analysis of the data set.

The rest of this paper is organized as follows. In Section II, we describe the objectives of the masquerade attack data collection project. Section III covers the methodology for designing a user study. In Section IV, we describe the procedure followed during the execution of the user study or capture-the-flag exercise. In Section V, we present the lessons learned throughout the masquerade data gathering project. Finally, Section VI concludes the paper by summarizing the main points of the paper.

II. Project Objectives

In the case of masquerade attack detection, most detection approaches used machine learning techniques to profile normal user behavior, and detect abnormal behavior that could be indicative of a masquerade attack. The vast majority of these techniques were evaluated using the the Schonlau dataset [11], gathered by Mathias Schonlau [2]. This dataset suffers from several shortcomings. The first shortcoming is the absence of any command arguments. Only simple commands have been collected. Another weakness is the lack of timestamps that indicate when the user commands were issued. No indication is available as to what time period is covered by the 15,000 commands collected from each user. It could take one user a few days to issue this number of commands, when it takes another a few months to do the same. This indicates another shortcoming of this dataset, namely the heterogeneity of the users. Not all users have the same expertise with Unix commands, nor do they have the same job functions. The wide variety of backgrounds amongst the users causes differences in their behaviors, such as the variety of commands that they use.

While all of the above shortcomings of the Schonlau dataset are important, perhaps the most significant weakness of this dataset is the **lack of** real **masquerader data**. All of the user command sequences gathered in this dataset were issued by normal users performing their regular day-to-day jobs. No command sequences were issued by attackers. Masquerade attacks were simulated by randomly inserting excerpts of command sequence from one user into the command sequences issued by another user. This practically turned the masquerade attack detection exercise into an author identification exercise. The fact that the command sequences belong to users with widely varying Unix proficiency and different job description further weakens the accuracy results achieved by the proposed "masquerade attack" detection classifiers, which have been only evaluated in an "author identification" exercise.

In order to overcome these weaknesses, we launched an initiative to collect our own dataset for masquerade attack detection and to make it available for the broader research community. We call this dataset the RUU (Are You You?) dataset [1]. The dataset exceeds 10 GBytes in size. The data collection tasks consisted of gathering computer usage data belonging to a large homogeneous set of *normal* users, and simulating masquerader attacks in a capture-the-flag exercise. The threat model considered assumes that the masquerader attackers are not familiar with the file system under attack. In the next sections, we describe the steps taken to design and execute the capture-the-flag exercise. This exercise served two goals: (1) collecting simulated masquerader data, and (2) testing a hypothesis related to masqueraders' search behavior. In the following section, we describe the design aspects of the user study in order to test our hypothesis, while collecting quality data.

III. User Study Methodology

The first step in designing a user study is to state the hypothesis that is to be tested empirically. This also requires identifying the null hypothesis which is to be rejected by the experiment. The goal of our first user study is to show that the intent of a masquerader can be manifested in their file system search behavior. Our *experimental* hypothesis states that if the intent of the masquerader is *malicious*, then they will engage in a *significant* search activity on the victim's system. Our *null hypothesis* states that the manipulation of the masquerader's intent does not have any significant effect on the the masquerader's search behavior. In other words, the observed *significant* effect on search activity that gets observed during the experiment can be attributed to the manipulation of the masquerader's intent, and cannot be the result of pure chance.

A. Experimental Variables

Stating the experimental hypotheses also requires identifying the experimental variables: the independent variable, the dependent variable, and any confounding variables. The independent variable is the one variable that gets manipulated by the researcher, while all other are kept constant. The dependent variable is directly and tightly dependent on the independent variable. It is the observed behavioral feature to be measured by the researcher during the experiment.

We hypothesize that user search behavior is a behavioral feature that is impacted by the user's intent. If a masquerader is looking to steal information, their intent will be manifested in their search behavior through the volume of the search activities performed by the masquerader. Our goal was to confirm this conjecture and to show that the attacker's search behavior is different from a normal user's search behavior, and that monitoring search behavior could be used for the detection of a masquerader's attack.

So the masquerader's *intent* constitutes the independent variable in our experiment. The dependent variable in our study is the search behavior of the masquerader, and their search volume in particular. Confounding variables are random variables that could affect the observed behavioral feature, namely *search*, such as problems with the experimental equipment or skill level. These variables are to be minimized if not eliminated. We address the approach that we have taken to limit them in subsections III-C and IV-C .

The question then is how can we manipulate an attacker's intent in this user study? This is by no means a simple task. However, it can be achieved by crafting different and detailed scenario narratives that are handed to the participants in the experiment.

B. Scenario Narratives and Control Groups

When dealing with human experimental design for cybersecurity studies, the scenario narratives should give the experiment participants detailed background information about the attack and the attacker's motives. This enables the participants to play the role of the attacker described in the experiment, and assume their intent.

We developed a very specific attack scenario that described the masquerader's motives and the surrounding conditions of the attack. The masquerade attack scenario had to be:

- Representative of masquerade attacks, *i.e.* generalizable: When conducting a cybersecurity-related user study, it is very expensive to test all attack variants both in effort and time. Testing each attack variant requires recruiting an additional number of human subjects to participate in the experiment. Therefore, it is very important that the scenario narrative used in the study is descriptive and representative of the attack under study.
- Conforming to our threat model: Gathering quality data which can be effectively used for empirically testing the experimental hypothesis requires that the scenario narrative used in the user study accurately reflects the threat model.
- Easily executable in a user study: This means, for instance, that the execution of the masquerader scenario had to be time-limited. Not specifying a time limit for the attack adds a lot of uncontrolled variability to

the experiments. Furthermore, it makes the experiments costly both in participants and researcher time.

- Detailed: The scenario narrative should be as detailed as possible. Answers to anticipated questions that could be posed by the participants should be included. Giving the answers to these questions to the participants in advance reduces the needs for asking such questions, and therefore limits the verbal communication between the researcher and the participant. Furthermore, it ensures that all participants receive the same instructions, therefore minimizing the participant bias.

In our attack scenario, the masquerader gets an opportunity to access a coworker's computer during a 15-minute lunch break, while the coworker leaves the office and stays logged in to their computer. We described the financial difficulties that the masquerader was going through, and the personal conflict that they had with the coworker. The attacker's objective was to find any information that could be used for financial gain. We strove to ensure that the task of the user study participants was goal-oriented, thus revealing the malicious intent of the attacker.

One may argue that simulating a masquerade attack is not appropriate, and that it is hard for an innocent student to act as a masquerader. We argue that, if the scenario of the experiment is well-written, and with very clear instructions, the participants in the experiment will follow the instructions. To this extent, we refer the reader to the very well-known Milgram experiment, which showed how subjects obey an authority figure and blindly follow instructions, even when they contradict their own values and ethics [9].

Besides the 'malicious attacker' scenario described above, we developed two other scenarios: a 'benign masquerader' scenario, and a 'neutral' scenario. In the benign scenario, the participants experienced a hard drive failure and could access a coworker's computer while their coworker left their computer exposed for 15 minutes, in order to finish working on an urgent project. In the neutral scenario, the participants in this scenario had no compelling reason to access the co-worker's computer. They were left to freely choose whether they wanted to access their coworker's desktop when the coworker left during a lunch break.

These scenarios are our means to manipulate the intent of the attacker. Therefore, we strove to keep all variables constant including the duration of the experiment, the type of relationship between the attacker and the victim, etc. We used each scenario to collect data for a control group against which we compare the results achieved using the 'malicious attacker' scenario. Table I compares the experimental variables that we controlled across all three scenarios.

TABLE I
COMPARISON BETWEEN EXPERIMENTAL VARIABLES IN ALL USER STUDY SCENARIOS

Experimental Variable	**Value**	**Same/ Different**
Scope	Local File System of Colleague's Computer	Same
Environmental Constraints	IDS Lab Computer	Same
Desktop Configuration	Same Recent Documents and Applications	Same
Time Constraints	15 minutes	Same
Intent	**Malicious, Benign, or Neutral**	**Different**

C. Sampling Procedures for Higher Experiment Sensitivity

In order to increase the sensitivity of our experiment, we had to reduce uncontrolled variability. This in turn requires controlling user bias which makes up the largest source of error variance in user study experiments [8]. In behavioral sciences, there are three different techniques or sampling procedures used to reduce subject variability and user bias. The first and preferred technique is the use of the same subject in all 'treatment conditions' of the experiment, that is in all three scenarios. This procedure could not be used in our experiment as it undermined the assumption that masqueraders were not familiar with the file system under attack. Using the same subjects in different treatment conditions of the experiment means that they will be exposed to the file system more than once. This implies that, in the second and third treatment condition or scenario, the subjects have prior knowledge about the file system, which violates the assumptions made in our threat model. Recall that our threat model assumes that the masquerade attacker is not familiar with the victim's file system.

The second approach and probably the most obvious approach is to select a homogeneous group of subjects, *i.e.* subjects with similar characteristics that are relevant to the experiment, such as their familiarity with the use of computers, their ability to search for information in a file system, and their acuity or sense of cyber-security. Finally, the third approach for reducing subject variability is the use of several small subject sets with characteristics that are highly homogeneous within one set, but widely varying between sets.

We have chosen the second approach, and selected subjects who were all students at the Computer Science department of Columbia University, so that they have comparable skills. This should minimize the variability between subject with respect to their familiarity with computer usage, and how to search a desktop in order to steal information, or how to perform a data theft attack without being detected. This should reduce confounds and bias in the results of this user study.

D. Power Analysis and Sample Size Estimation

Power analysis is an important step in designing a user study, which usually gets neglected by many researchers working on cyber-security user studies. An experiment's power is an indication of how statistically significant its results may be, and it varies normally between 0.5 and 0.9. The higher the power, the more statistically significant the results are. The researcher has to determine the desired power of the experiment, in order to calculate the required number of samples, or human subjects, needed for each experimental condition of the user study. Obviously, reaching a higher power value requires a higher number of samples.

The adequate sample size for the experiment depends on several parameters:

- Form of the experiment: The number of independent variables manipulated in the experiment and the number experimental conditions drive the number of subjects needed for the user study. The more treatment condition analyzed, the higher the number of participants needed in the experiment.
- Hypothesis to be tested and the null hypothesis: This requires identifying the desired effect size w^2 that the researcher wishes to detect. The effect-size measure is a measure of the size of an effect in the overall population, regardless of the specific details of the user study.
- Desired power: Achieving a higher power value for the experiment results requires a higher number of samples. A power of about 0.8 seems to be reasonable for human behavioral experiments [8].

The sampling size n needs to be large enough in order for the experiment to produce a reasonable accuracy, *i.e.* to limit the sampling error. Using a larger sample size may only add to the recruiting costs without adding more accuracy.

1) Calculating the Effect Size: One way to define the effect w is through Cohen's *d*, which is frequently used in estimating sample sizes [7]. A lower Cohen's *d* indicates the need for larger sample sizes, and vice versa.

The measure of the effect size is the the standardized difference between two means. For example, one of the dependent variables that we measured in our user study is the number of file touches or accesses by the user. This is a search-dependent feature that, we conjectured, is dependent on the attacker's intent, as can be seen in Figure 1(a).

We measure the population effect size w of this feature by calculating the standardized difference between means of file accesses within the malicious scenario population and the benign scenario population. The standardized difference takes into account the variability of of file touches from one user to another.

The effect size is measured as follows:

$$w^2 = \frac{\mu_1 - \mu_2}{\sigma_{12}} \tag{1}$$

where μ_1 is the mean of file accesses by masqueraders in the malicious scenario, μ_2 is the mean of decoy file accesses by users in the benign scenario, and σ_{12} is the standard deviation based on both user populations of the two scenarios. Considering the sample sizes s_1 and s_2 of the two populations or groups, then σ_{12} can be defined:

$$\sigma_{12} = \sqrt{\frac{SS_1 + SS_2}{df_1 + df_2}} \tag{2}$$

where df_1 and df_2 are the degrees of freedom in both populations 1 and 2 respectively, i.e, $df_i = s_i - 1$ where $i \in 1, 2$ and SS_i is defined as

$$SS_i = \sum_{j=1}^{s_i} y_{i,j}^2 \tag{3}$$

where $y_{i,j}$ is the number of file touches by user j in population i.

2) Estimating the Sample Size: Once the effect size has been estimated and the desired power value determined, the required sample size n can be calculated as follows:

$$n = \phi^2 \frac{1 - w^2}{w^2} \tag{4}$$

where ϕ is known as the non-centrality parameter.

The non-centrality parameter ϕ indicates to which extent the user study provides evidence for differences among the two population means. It can be extracted from the power function charts developed by Pearson and Hartley based on the desired power [10]. Using the right sample size is important for reaching the desired power of the experiment between 0.5 and 0.9). However, increasing the sample size to reach very high power values beyond 0.9 may be very costly, as it becomes harder to reach power values beyond that value.

IV. User Study Execution

During the execution of a user study, special care has to be taken to ensure the validity of the results and the compliance with institutional policies. Several procedural and technical challenges could be encountered throughout this process, the first of which is obtaining institutional review board approval to perform the experiments.

A. Obtaining Institutional Review Board Approval

The major compliance procedural challenge encountered during the concept phase of our data collection project was the IRB process. Obtaining IRB approval to conduct user studies is a costly process, both in time and effort. During the IRB process, the research plan and objectives including the detailed description of the planned experiments are reviewed in advance in order to protect the privacy rights of the human subjects involved in the research. This process is required in all institutions that receive research funding from the US federal government. It is a lengthy process and may be iterative in some cases, as some clarifications may be requested by the IRB. For example, we had to submit the exact text of the *call for participation* to students in our user studies. We had to specify in advance what pieces of data we would collect, for how many users, and for how long. We did not necessarily know the answers to all of these questions when we initiated the IRB review process. In order to continue working on the project, we had to extend or re-initiate the review processes on several occasions.

B. Sensor Development and Deployment

To collect masquerader data during the user studies, as well as normal user data, we developed a first sensor that gathers user command data, including command arguments and timestamps. Recall that our objective is to collect a dataset of Unix and Linux user commands from a homogeneous set of users and overcomes the weaknesses of the Schonlau dataset.

Several technical challenges were encountered during the sensor development and deployment phases of the project.

We built a first sensor for the Linux operating system. The sensor uses a kernel hook to audit all events on the host. It collects all process IDs, process names, and process command arguments in real time. The hooking mechanism used is the *auditd* daemon included in most modern Linux distributions. When we deployed the sensor, we could not get enough adopters. Most students on campus did not run the Linux operating system on their personal computers. Therefore, we had to develop a second sensor that runs on Windows systems. This delayed the project by several months.

We developed a second sensor for the Windows XP platform. The Windows sensor monitors all registry-based activity, process creation and destruction, window GUI access, and DLL libraries activity. The data gathered consists of the process name and ID, the process path, the parent of the process, the type of process action (e.g., type of registry access, process creation, process destruction, etc.), the process command arguments, action flags (success/failure), and registry activity results. A timestamp is also recorded for each action. The Windows sensor uses a low-level system driver, DLL registration mechanisms, and a system table hook to monitor user activity. It relies on hooks placed in the Windows ServiceTable, which is a typical approach used by malicious rootkits.

In the first data collection round, we had about 15 volunteers who agreed to install the sensor on their computers and to share the data collected about their normal activities on the computer. All of these students were taking the Intrusion Detection Systems (IDS) class at Columbia University. This sample was not large enough to conduct experiments and achieve results with high statistical significance. Therefore, we had to collect more data when the IDS class was offered the following year. Meanwhile we had prepared a second sensor for Windows Vista. Unfortunately, we realized that most students have upgraded their operating systems from Windows XP to Windows 7 directly.

Developing a sensor for Windows 7 required rewriting the core parts of the sensor, since Windows 7 no longer allowed placing hooks in the Windows ServiceTable to intercept system calls. Moreover, the sensor could not run on 64-bit versions of the Windows operating systems. Even certain updates to the operating system, such as the Windows XP Service Pack 3 (SP3), which includes security, performance, and stability updates to Windows XP, caused the sensor to crash in some instances. This caused our server to lose contact with some user sensors. The data collected for some users covered only intermittent periods of time. In some cases, users had to re-install the sensor. In other cases, users decided to run the sensor on virtual machines, where the guest operating system is Windows XP. This posed a data quality issue. User data collected from virtual machines is only a subset of the user's interaction with their personal computer. Therefore, it may not fully reflect the user's typical behavior. All these technical issues posed data sanitization challenges, which we present in the following section.

C. Reducing Confounds and Bias

Besides reducing subject variability, we strove to reduce the experimental treatment variability by presenting each user study participant with the same experiment conditions. In particular, we used the same desktop and file system in all experiments. We also ensured that the desktop accessed by the subjects looked the same to each participant. In particular, we cleaned up the list of recently accessed documents, and opened MS Office documents before the start of each experiment, and automated the data collection and uploading to a file server so that the data collected does not reside on the desktop used in the experiment and does not bias the results of the experiment. Finally, we strove to limit the number for unanalyzed control factors. For example, we ensured that all the experiments were run by the same research assistant.

D. Data Sanitization Challenges

When deploying the sensors on users' personal computers, each installed sensor was given a unique sensor ID. Data collected from one sensor was uploaded to a central server and stored under its own directory. Many users had to to re-install the sensor due to some incompatibilities with their operating system. Therefore, data belonging to one user was stored under different directories. Linking or combining this data was not straightforward in the absence of any user or system identification mechanism, other than the sensor ID. Extensive data analysis was required to find clues that could be used to link the data collected from one user.

Our sensors provided mechanisms for the users to protect their private data and sanitize it if they wished to. Unfortunately, many users did not take advantage of these mechanisms, either due to laziness, or due to lack of awareness of the consequences of revealing their identities to the research and broader communities by sharing their data. It is also possible that users did not care about the consequences of revealing their identity when their data was shared. Whatever the reason was, we had a moral obligation of sanitizing the data and ensuring that the identities of the students were anonymized. This proved to be a major challenge because we did not know the names and user IDs of our users. In the absence of the list of user names and IDs, we had to manually review all records of data collected and anonymize these user names and IDs wherever they showed up, such as in file or directory names. Inspecting 20 million records was a very time-consuming process, and the results are possibly less than fully-satisfactory, as we may have missed a username in a record here or there.

E. Post-Experiment Questionnaires

Having study participants fill in post-experiment questionnaires is one of the best practices when conducting user studies. These questionnaires can be used to gather more information about the users including demographics and skill levels that could be used for statistical analysis. The researcher can also use these questionnaires to determine whether the participants could identify the hypotheses tested in the experiment. Knowing the true experimental hypothesis may cause some user bias, and therefore impact the results of the experiment. Some insights into the strategies used by the attackers could also be learned from the participants' answers to these questionnaires.

F. *Final Data Set Characteristics*

Eighteen computer science students installed the Windows host sensor described above on their personal computers. The host sensor collected and uploaded it to a server, after the students had the chance to review the data and their upload. The students signed an agreement for sharing their data with the research community. This dataset reached more than 10 GBytes in size. The data collected for each student spanned 4 days on average. An average of more than 500,000 records per user were collected over this time. Tables II and III show two sample RUU records.

The dataset also contains data collected from forty students who acted as masqueraders in our user study. The normal user dataset and the simulated masquerader dataset are both available for download after signing a usage license agreement[1].

TABLE II
SAMPLE RUU RECORD: REGISTRY ACCESS

Column[1]	Value
Syshash	0cc7ebd580b39bb037627c2a71c979
Auditaction	QueryValue
Processname	explorer.exe
Path	HKCR\CLSID\871C5380-42A0-1069-A2EA-08002B30309D\ShellFolder\Attributes
Stringreturn	SUCCESS
PID	408
PPID	-1
Timestamp	2009-12-09 21:05:46

TABLE III
SAMPLE RUU RECORD: FILE ACCESS

Column[1]	Value
Syshash	68cad4c71ba63fb140a3bb8a4e7d0d
Auditaction	USERTOUCH_1
Processname	AcroRd32.exe
Path	C:\Program Files\Adobe\Reader 9.0\Reader \AcroRd32.exe
Stringreturn	shopping_list_20091211.pdf - Adobe Reader
PID	1676
PPID	-1
Misc1	1676 ¡= 1008 - 0 1553 3
Timestamp	2009-12-11 19:18:15
Decoy Alert	1
Size	1400

G. *Summary of Experimental Results*

The user studies and data collection exercise described above were conducted in order to test a conjecture, that modeling user search behavior can be used to detect masquerade attacks. Another objective of these user studies was to investigate whether decoys can be used to dedtct such attacks effectively as well.

We conducted one experiment where the objective was to provide evidence for our conjecture that the masquerader's intent has a significant effect on their search behavior. We extracted three features from the data collected in the user study after experimenting with several features such as the frequencies of the different types of user actions and application events: (1) the number of files touched during an epoch of two minutes, (2) the number of automated search-related actions initiated by the masquerader, and (3) the percentage of manual search actions during the same epoch. Automated search actions are search actions launched using a desktop search tool such as *Google Desktop Search*. Manual search actions are file system navigation or exploration systems. The distributions of these features for the malicious, benign and neutral scenario participants of the user study are displayed in Figure 1. The experiments demonstrated that the manipulation of the user intent has demonstrated a significant effect on user search behavior. Further details about these experiments can be found in our upcoming paper [5].

To evaluate the role of decoys in detecting masqueraders, we placed 30 decoys in the local file system of the lab computer used in our user study. We conducted several experiments to evaluate the effective placement of these decoys and the importance of various decoys properties in effective masquerade attack detection. Figure 2 shows the distribution of decoy touches by user study scenario. One can clearly see that the number of decoy touches is very low in the benign and neutral scenarios when compared to the malicious scenario.

Our study shows that, among 40 masqueraders, 17 attackers were detected during the first minute of their masquerade activity, while another ten were detected during the second minute after accessing the victim's computer, as can be seen in Figure 3. All masqueraders were detected within ten minutes of their accessing the system under attack. Details about the decoy experiments can be found in our decoy placement paper [4].

Finally, we have also used the dataset and user studies to show that combining trap-based detection with user behavior monitoring achieves better accuracy results than user profiling alone when detecting masquerade attacks as can be seen in Figure 4 [3].

V. LESSONS LEARNED

During the different phases of the user study, we have learned some lessons, which we group into compliance-related, scientific, and practical lessons learned, and share below:

A. *Compliance-related Lessons Learned*

- **Initiate the IRB review process early:** Anticipate future data and experiment needs of your research project, and start the IRB review process as early as possible. This is a lengthy and iterative process, and one may not have detailed answers to all questions presented in the IRB

[1] More information about the data collected in this user study can be found at: http://sneakers.cs.columbia.edu/ids/RUU/study.html

protocol in advance. So it is wise to initiate the process early in case clarifications are requested by the review board.

- **List a larger number of user study subjects in the IRB protocol than you may need:** This number is required per the IRB protocol and it is hard to judge. It may be wise to ask for a larger sample of human subjects, so that the IRB can be left 'open' in case one needs to solicit additional subjects.
- **Have user study participants sign waivers:** Ensure that all participants in a user study sign waivers indicating that they are willing to share their data 'as-is' with the research community: Data sanitization is a major issue and it depends upon the guarantees of anonymity that are provided. Subjects in our user study were told about the data being publicly available, and were provided with mechanisms to sanitize their own data. However many have chosen not to do so for various reasons, such as lack of awareness or laziness.

B. *Scientific Lessons Learned*

- **Identify the independent variable:** A single independent variable should be identified, and the variables that depend on it should be observed and measured. The independent variable is the factor that is controlled by the researcher in the experiment. Variability in all other confounding factors should be eliminated or reduced to a minimum.
- **List all the assumptions made about the participants in a user study:** Ensure the assumptions are described clearly in the user study scenario. For example, does the participant know whether they are being monitored? Do they know whether the system is baited? This clarifies the task for the participants, and limits verbal communication between the participant and the researcher, thus reducing user-induced variability. Detailed scenario narratives are critical in limiting user variability and bias.
- **Identify ways for reducing variability and baselining users:** When designing a user study, it is critical to reduce all source of variability, and especially user bias. Baselining users is important. For example, when asking the participants to run a sensor that collects their data, one may ask them to do so without discussing the purpose of the sensor. This provides a way to record their user behavior without any bias at all.
- **Perform a power analysis:** Conducting a power analysis early on is important for identifying the desired power of the experiment, and therefore estimating the required minimum number of human subjects needed to achieve that power. An under-powered experiment may lead to statistically insignificant results.

C. *Practical Lessons Learned*

- **Anticipate the technology market trends:** Technology changes very quickly and users adopt new technologies fast. Data collection tools or sensors used in a user study have to follow these trends if the data gathering project is to span many years.
- **Pilot experiment:** If possible, pilot the experiment before conducting the real user study. Besides discovering potential technical problems, you could receive questions from the participants that should be answered in the scenario narratives when conducting a pilot experiment. Again, identifying these questions and providing answers in the scenario narratives given to the participants provides them with the same instructions, and reduces user bias.
- **Have participants fill in post-experiment questionnaires:** Although this step is not required for the success of a user study, the qualitative and quantitative analysis of these questionnaires may lead to some additional insights about the experiment results.

VI. Conclusion

Human behavior is complex and is not easily described through models. Conducting human behavioral experiments is therefore very complicated and requires special care and planning to reduce user bias, increase the sensitivity of the experiment, and improve data quality. In this paper, we presented a methodology for conducting cyber-security user studies. We also described several challenges encountered throughout all phases of our data collection project for the evaluation of masquerade attack techniques, including data sanitization challenges. We highlighted some lessons learned throughout the project that could serve as guidelines for researchers doing user studies in the same field. We believe that user studies, require extensive planning. Many stakeholders are involved, including the IRB. Many sources of variability should be controlled or eliminated, and special efforts are required to reduce user bias and protect user privacy.

Acknowledgment

This work was partially supported by the National Science Foundation, CCF Grant 0950373, 'Measuring the Security Posture of Large Financial Enterprises: An EAGER Proposal to NSF CCF'. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the National Science Foundation.

References

[1] Ben-Salem, M. RUU dataset: http://www1.cs.columbia.edu/ids/RUU/data/.

[2] Ben-Salem, M., Hershkop, S., and Stolfo, S. J. A survey of insider attack detection research. In *Insider Attack and Cyber Security: Beyond the Hacker* (Heidelberg, 2008), Springer.

[3] Ben-Salem, M., Hershkop, S., and Stolfo, S. J. Combining a baiting and a user search profiling techniques for masquerade detection. In *Columbia University Computer Science Department, Technical Report # cucs-018-11* (2011).

[4] Ben-Salem, M., and Stolfo, S. J. Decoy document deployment for effective masquerade attack detection. In *DIMVA'11: Proceedings of the Eighth Conference on Detection of Intrusions and Malware & Vulnerability Assessment* (Heidelberg, July 2011), Springer, pp. 35 – 54.

[5] Ben-Salem, M., and Stolfo, S. J. Modeling user search-behavior for masquerade detection. In *To Appear in the Proceedings of the 14th International Symposium on Recent Advances in Intrusion Detection* (Heidelberg, September 2011), Springer.

[6] CERT. 2010 e-crimes watch survey, 2010.

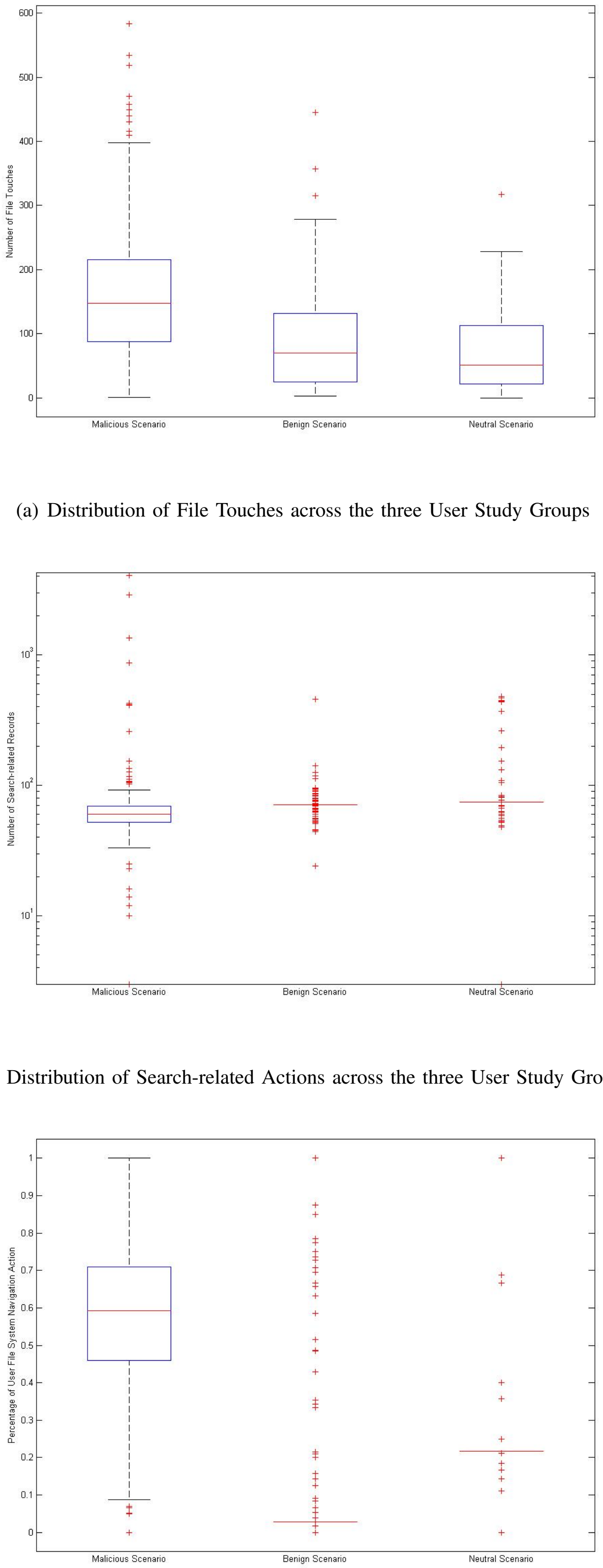
[7] Cohen, J. A power primer. *Psychological Bulletin 112*, 1 (July 1992), 155–159.

[8] Keppel, G. *Design and analysis : a researcher's handbook*. Pearson Prentice Hall, 2004.

[9] Milgram, S. *Obedience to Authority: An Experimental View*. Harper-collins, New York, January 1974.

[10] Pearson, E. S., and Hartley, H. O. Charts of the power function for analysis of variance tests, derived from the non-central F-distribution. *Biometrika 38*, 1 (July 1951), 112–130.

[11] Schonlau, M. Schonlau dataset: http://www.schonlau.net, 2001.



(a) Distribution of File Touches across the three User Study Groups

(b) Distribution of Search-related Actions across the three User Study Groups

(c) Distribution of the Percentage of File System Navigation User Actions across the three User Study Groups

Fig. 1. Distribution of Search-related Features across the three User Study Groups

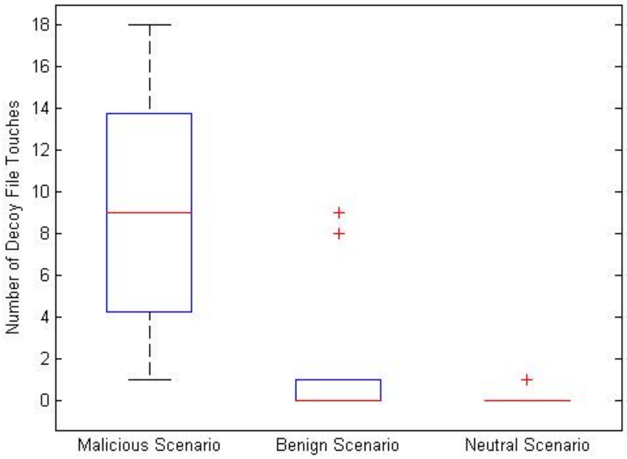


Fig. 2. Distribution of the Number of Decoy Document Accesses by Scenario

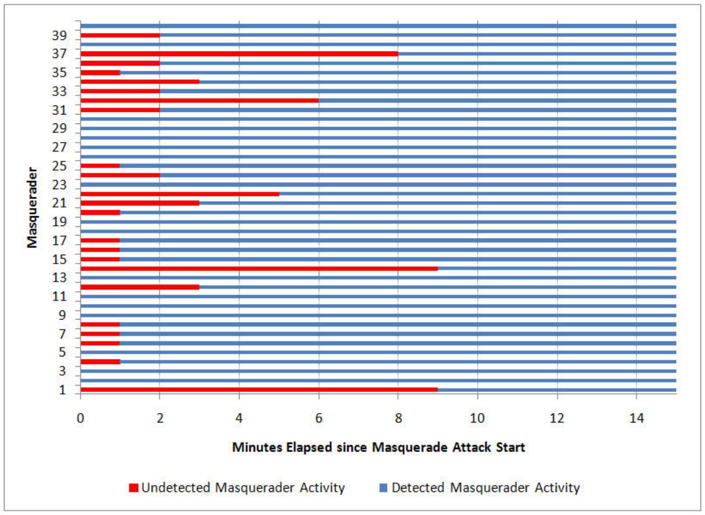


Fig. 3. Detection Time by User

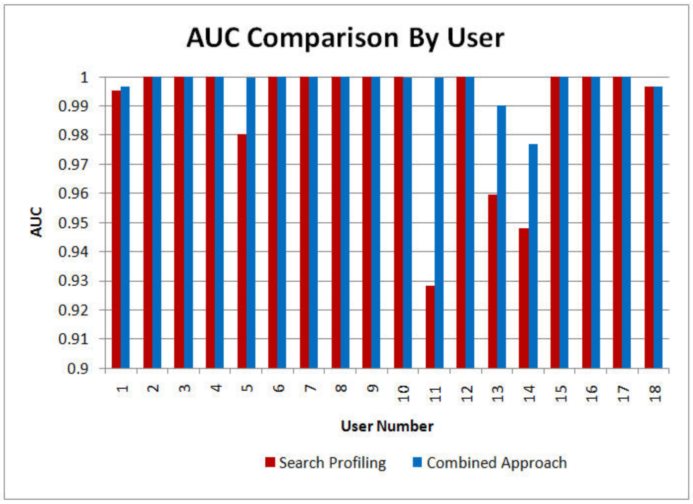


Fig. 4. AUC Comparison By User Model for the Search Profiling and Intgrated Detection Approaches