CERIAS Tech Report 2006-15

DETECTING SOCIAL ENGINEERING

by Michael Hoeschele

Center for Education and Research in
Information Assurance and Security,
Purdue University, West Lafayette, IN 47907-2086

DETECTING SOCIAL ENGINEERING

A Thesis

Submitted to the Faculty

of

Purdue University

by

Michael David Hoeschele

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

May 2006

Purdue University

West Lafayette, Indiana

ACKNOWLEDGEMENTS

I want to thank my advisor, Marc Rogers, for his feedback, genuine interest, and direction in my research. I would also like to thank my thesis committee members Melissa Dark and Victor Raskin for their comments and time given to help me refine my research. Finally I would like to thank my parents for encouraging me to pursue the field of computer security and to make the trip out to Purdue University.

TABLE OF CONTENTS

Page
LIST OF TABLES v
LIST OF EQUATIONS vi
ABSTRACT vii
INTRODUCTION 1
STATEMENT OF THE PROBLEM 2
SIGNIFICANCE OF THE PROBLEM 3
PURPOSE OF THE STUDY 3
DEFINITIONS 4
DELIMITATIONS 5
LIMITATIONS 5
REVIEW OF THE LITERATURE 6
Social Engineering 6
Methods 6
Motivation 9
Targets 9
Psychological Foundations to Social Engineering 10
Social Engineering Defense Architecture (SEDA) 12
State of the Field 13
Policy 14
Inoculation Training 14
Success of Current Solutions 15
Effects of Surveillance in the Workplace 16

Page

METHODS 20
Data Creation 20
Constructs 21
Instruments 23
Measurement 23
Attack Detection Testing 24
Data Parsing Testing 27
Data Analysis 27

RESULTS 29
Attack Detection Testing 29
Test Cases of Interest 29
Data Parsing Testing 31
Test Cases of Interest 32

DISCUSSION 35

CONCLUSION 39
Future Research 40

REFERENCES 42

APPENDICES
Appendix A 46
Appendix B 72
Appendix C 89
Appendix D 92
Appendix E 101
Appendix F 103
Appendix G 105
Appendix H 106
Appendix I 107
Appendix J 108

LIST OF TABLES

Table Page
1: Attack Signature Breakdown 22
2: Data based attacks 25
3: Signature based attacks 25
4: Results 29

LIST OF EQUATIONS

Equation Page
1: Accuracy formula 27
2: False positive and negative calculation 28

ABSTRACT

Hoeschele, Michael David. M.S., Purdue University, May, 2006, Detecting Social Engineering. Major Professor: Marcus K. Rogers.

This study consisted of creating and evaluating a proof of concept model of the Social Engineering Defense Architecture (SEDA) as theoretically proposed by Hoeschele and Rogers (2005). The SEDA is a potential solution to the problem of Social Engineering (SE) attacks perpetrated over the phone lines. The proof of concept model implemented some simple attack detection processes and the database to store all gathered information. The model was tested by generating benign telephone conversations in addition to conversations that include Social Engineering (SE) attacks. The conversations were then processed by the model to determine its accuracy to detect attacks. The model was able to detect all attacks and to store all of the correct data in the database, resulting in 100% accuracy.

INTRODUCTION

Social Engineering (SE) presents an interesting problem for cyber forensics that has not been researched sufficiently; this is evident by the lack of formal published research on the topic. Research focused specifically on SE primarily discusses definitions of the topic, and the techniques used to breach security. No research has been preformed on forensic analysis of these attacks. Currently the only method presented to detect or trace SE attacks is the education of personnel on SE techniques. An example of this is found in the Information Security Management Handbook. The chapter entitled Social Engineering: The Forgotten Risk by Rogers and Berti (2002) discusses how to prevent SE attacks. Most of their suggestions are security policies and training of personnel to be aware of what an attack may look like, both of which are still dependant upon the human element. Dolan (2004) makes similar suggestions for detecting SE attacks. He claims that successful social engineering attacks rely on the employees of an organization, "to contain such an attack, employees must be well trained and familiar with common SE techniques" (p. 6).

In 1997 the Client Server Computing magazine published the article Liar, Liar by Julie Bort where she states, "Of the 384 respondents who confessed to being attacked over the last year, social engineering was to blame in 15 percent of the cases--the second largest cause" (p. 40). While this is somewhat outdated, it clearly shows that SE is

neither a new, nor a small problem. No other direct measures of the prevalence of SE attacks could be found. Other more current publications, like the FBI/CSI annual Computer Crime and Security reports, from 2002, 2003, and 2004, showed vague statistics on telecommunications fraud that could include SE (Gordon, Lobe, Lucyshyn, Richardson, 2004; Power, 2002; Richardson, 2003). More details on these reports are provided in the financial findings section of the literature review.

This topic is of interest because of Kevin Mitnick's book, The Art of Deception (2003). The book provides many interesting stories on the ease of perpetrating SE attacks, but fails to articulate reasonable or useful methods of preventing such attacks. Also absent from the book is a discussion of how to perform a forensics investigation after a SE attack occurred.

Bort (1997) states, "There is no hardware or software that can defend your information systems against a human being telling a convincing lie" (p. 40). This was true in 1997, but Raskin, Hempelmann, and Triezenberg, (2004) are currently researching the problem of lie detection in Natural Language Processing and are making progress. They have proven it to be a tractable problem and created a working model. Currently the model can correctly parse parts of the English language, but cannot handle it in its entirety.

STATEMENT OF THE PROBLEM

Hoeschele and Rogers (2005) proposed the Social Engineering Defense Architecture (SEDA) as a potential system to log and prevent Social Engineering (SE) attacks. Prior to this research, none of the systems involved in the SEDA have been implemented, which prevented a determination as to how well they work. This research

specifically targeted the viability of parsing pertinent information from conversations into a database and simple attack detection. The two types of attack detection implemented are data and signature based. Data based attack detection relies on detecting conflicting information from a caller, like a change in name from one call to the next. Whereas signature based attack detection depends on certain phrases being used in the conversation.

SIGNIFICANCE OF THE PROBLEM

The significance of the problem was that the Social Engineering Defense Architecture (Hoeschele & Rogers, 2005), which may be able to prevent Social Engineering (SE) attacks, had not been tested before this research to determine its viability. The only way to determine this is through empirical testing. More specifically, testing of the architecture not the component parts is of the greatest significance. This is because if the architecture is flawed, the component parts are useless, but if it can be shown that it is a viable architecture then it is worthwhile to continue research. Then testing of the component parts can begin without wondering how they will fit into the architecture.

PURPOSE OF THE STUDY

The purpose of this study was to develop and test a proof of concept model for the Social Engineering Defense Architecture (SEDA) (Hoeschele & Rogers, 2005). While parts of the SEDA were tested, the primary focus of the research is not on how well it performs, as it will not be a comprehensive proof of concept model. Instead the focus of this research was on determining the feasibility of the architecture of the SEDA in its current design. The specific parts that were tested are parsing information out of a

conversation, storing that information, detecting conflicts in the information indicating a social engineering attack, and detecting key phrases used in social engineering attacks. A database back end was also implemented to support the SE attack detection and parsed data logging.

DEFINITIONS

The following is a definition of terms and concepts that will be used in this study.

Computer Forensics: The definition is the same of as that of Digital Forensic Sciences as defined by the Digital Forensics Research Workshop (2001):

> The use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations. (p. 16)

Social Engineering: The malicious intent of cyber attackers attempting to illegally compromise an organization's assets by "Using relationships with people" (Dolan, 2004, p. 2).

False Positive: Input sentences for the SEDA proof of concept model that are benign in nature, but are flagged as attacks or incorrectly parsed data.

False Negative: Input sentences for the SEDA proof of concept model that are attacks, but are not flagged as such or data that is not parsed but should have been.

Proof of Concept Model: An implementation of part of the SEDA design.

Outsider: The caller in a phone conversation.

Insider: The receiver of a call in a phone conversation.

DELIMITATIONS

This research tested a proof of concept model of the Social Engineering Defense Architecture (SEDA). Components, like a voice to text converter, text independent voice signature generator, and Natural Language Processing (NLP) deception detection are not yet in a state where they can be implemented into the SEDA. As such only a proof of concept model was built. This modular approach also allows for easy updates to the SEDA when new technologies are released. One of the major delimitations in this study was the structure of the conversations designed to simplify data parsing. Further discussion of this can be found in the data creation section. Another was the simplified security policy used in testing, which limits the applicability of the results to the real world because of its austerity.

LIMITATIONS

The major limitation of this research was the immature nature of technologies the SEDA depends on. As mentioned in the delimitations section, not all of the components are available for implementation; this research was limited to a proof of concept model only. Also, due to the lack of existing research in this area, there was a large amount of trial and error to determine what was feasible.

REVIEW OF THE LITERATURE

While there is little scientific research on social engineering in general, and detecting and preventing it specifically, there is pertinent research regarding the principals that Social Engineering (SE) depends on. It is also valuable to discuss the current solution to the problem of SE attacks including inoculation training to better understand other possible solution. Finally, the Social Engineering Defense Architecture (SEDA) will be discussed followed by research regarding people's potential reaction to the deployment of such a system, which is important in aiding its acceptance.

Social Engineering

The use of Social Engineering (SE) is a common occurrence in society. Using relationships between people to obtain a goal is an every day occurrence and does not have to be nefarious in purpose. However, the SE that is referenced in this research is of the unscrupulous sort. The following is a discussion of the who, why, and how of SE attacks.

Methods. Social engineers, akin to computer hackers, take advantage of weaknesses in the system in place to keep enemies out, and assist insiders. As Dolan (2004) stated:

> Social engineers use tactics to leverage trust, helpfulness, easily attainable information, knowledge of internal processes, authority, technology and any

> combination there of. They often use several small attacks to put them in the position to reach their final goal. Social engineering is all about taking advantage of others to gather information and infiltrate an attack. The information gained in a phone book may lead to a phone call. The information gained in the phone call may lead to another phone call. A social engineer builds on each tidbit of information he or she gains to eventually stage a final, deadly attack. A successful social engineering attempt could result in great financial loss for the target company. A motivated attacker will be willing to gain information in any way possible. (p. 3)

One important reason why SE is easily achieved is that people in general have the desire to help others, and gain satisfaction from it (Dolan, 2004). One of the fundamental skills of a social engineer is the ability to establish trust. This can be accomplished by masquerading as someone an employee should trust. Typically most of the information necessary to impersonate someone's identity for a SE attack is publicly available. Reverse phone look-up directories are available on the web free of charge, (e.g., http://www.reversephonedirectory.com/). Once a phone number or address is obtained, the other directly follows. Many organizations' web pages hold vast quantities of information such as organizational charts, which provides social engineers with a target from which to steal an identity or claim association. Another common technique is for a social engineer to ask to be transferred to an employee from one of the main telephone operators. The receiver of the transfer call does not have a phone number that is posted publicly, so anyone who calls them is already considered an insider to the receiver of the

call, as they were able to find the number, assumedly from the company phonebook. This can prove to be a strong enough credential to allow more internal numbers to leak out, increasing the social engineer's cache of information.

Further examples of SE attacks that Wendy Arthurs (2001) discussed are as follows:

> IT Support – Somebody, claiming to be from the company's IT support group, phones a user and explains that he is fault finding on the network. He has limited the fault to within the user's department but he needs a user ID and password from that department to finish tracing the problem. Unless the user has been properly educated in security practices, he will very likely give the "trouble-shooter" his information.
>
> Manager – The social engineer, using a perceived position of authority, phones the help desk demanding to know why he can't log on with his password. He then intimidates the help desk into giving him a new password by telling them he only has a limited time to retrieve some information for a report to the company vice president. He may also threaten to report the help desk employee to his supervisor.
>
> Trusted Third Party – The social engineer phones the help desk, claiming to be Susan Sly, the vice-president's executive assistant, stating the vice-president has authorized her to collect the information. If the help desk employee balks, she threatens job loss or a report to the employee's supervisor. (p. 2)

It can be seen from these examples that a large portion of the attacks are committed over the phone and rely on the fact that the receiver of the call has to take the caller's word that they are who they say they are. There is typically no form of authentication other than verbal quizzes regarding information only an employee should know.

Motivation. The motivation for SE attacks varies between a sense of curiosity and challenge seeking to directed attacks with the intent to compromise an organization's assets. The Hackers Manifesto gives us a brief insight into why hackers desire to break into secure areas. The claim is that of curiosity and a quest for knowledge. While harm may not be the intent, it is clear that substantial harm can be caused by such activities. Some social engineers also seek a challenge, something to test their skills on (Mentor, 1986).

The more troublesome type of attack is that of a social engineer with a purpose. The motivation could come from a recently fired employee seeking vengeance or a seasoned social engineer locating information for money. This type of social engineer has a better chance of succeeding and causing considerable damage in the process as they have much greater motivation to succeed and potentially more resources at their disposal (Dolan, 2004).

Targets. The types of targets for any SE attack range from product designs to personal employee information. However, it is important to note that these are only the end targets of the attack, and many small pieces of information must be obtained before the final target can be reached. The types of targets used to reach a final goal can include company policy or protocol, company phone books, organization trees, or server names. As this data is easily obtainable, the real challenge is protecting such seemingly trivial

information it in such a way as not to interfere too greatly with day-to-day workflow (Rogers & Berti, 2002).

Psychological Foundations to Social Engineering

It is evident from the above discussion of SE that a high level of deception occurs during SA attacks. Thus it is relevant to discuss research on people's proficiency at deception detection and how well they can be trained to improve this skill. It is important to understand the general psychological foundations to Social Engineering (SE) and deception to provide a background and context with which to discuss the topic.

It was "found that [untrained students] are able to differentiate correctly between honest and dishonest statements about 60% of the time, not terribly impressive in light of chance being 50% (as cited in Kraut, 1980)" (Forrest & Feldman, 2000). However, later Ekman and O'Sullivan (1991) found that members of the U.S. Secret Service showed relatively greater success than untrained students at detecting deception. This shows that it is possible to train people to better detect deception. However, improving the ability to detect deception is difficult (Ekman and O'Sullivan, 1991). Ekman and O'Sullivan showed this by testing police detectives, US Customs Service agents, and Central Intelligence Agency agents, all of whom are highly trained in deception detection by their agencies. However, their scores were no better than the untrained students. Considering the objectives of these professionals, they are highly motivated to correctly detect deception. They should also have far more experience at attempting to detect deception than the students.

According to Forrest and Feldman, studies by DePaulo, Stone, and Lassiter (1985a); Ekman and Sullivan (1991); and Zuckerman and Driver (1985) "suggest that the

most accurate judges in detecting deception are those who pay attention to the nonverbal behaviors of communicators because nonverbal behaviors are more predictive of deception than are other types of behavior" (2000, p. 118). Examples of nonverbal behaviors are speech hesitation, pitch, and speech errors. With regards to strictly verbal communications, DePaulo et al. (1985a), claim that the verbal components of a message are relatively poor indicators of deception.

Forrest and Feldman (2002) have researched a theoretical foundation for nonverbal versus verbal messages and deception detection. They found that subjects who were highly motivated to determine if someone was lying to them performed worse than subjects who had little motivation to detect a deception. They described highly motivated subjects as using central processing, which means actively analyzing the message for hints of deception. Forrest and Feldman (2002) suggest that those engaged in central processing pay more attention to the central arguments, and verbal message, which has been shown to contain fewer identifiers of deception. On the other hand, minimally motivated subjects engage in peripheral processing, which means they are not actively analyzing the message for deception. This allows them to ignore the message and verbal content and notice the nonverbal cues, which are better predictors of deception (Forrest & Feldman, 2002).

The Elaboration Likelihood Model (ELM) proposed by Petty (1977) proposed a similar effect of higher processing increasing deception detection. The primary component of the ELM is the elaboration continuum, which describes the amount of thought used in a persuasion setting. On polar ends of the continuum are high thought and low thought. Different levels of thought by the subject determines what types of

effects can be seen. Similar to Forrest and Feldman, the ELM postulates that in low thought, cue effects or nonverbal cues have the greatest impact on the subject's attitude. Also similar to Forrest and Feldman (2002), in high thought instances the ELM claims subjects are more persuaded by strong arguments, preconceived notions, and the object as an argument. An example of an object as an argument is if a social engineer is offering something that is attractive to an employee, like a bonus opportunity. The fact that the employee finds this attractive can be enough for them to come up with reasons for themselves to do what it takes to obtain it. This points to employees with high thought processing being more susceptible to deception because they will think about the argument and object, which can be manipulated by the social engineer. In low thought situations cue effects have a larger impact on people, which are much more difficult to manipulate as a social engineer (Petty, 1997).

It has already been shown that it is difficult to train people to detect deception, and without the nonverbal component of a communication it will be even more difficult. Also, training and policy has the goal of making employees more aware of potential deceptions. This could be helpful, or increase central processing and lead to easier deceptions. This needs to be studied in a more direct manner instead of attempting to tie relatively unassociated environments together.

Social Engineering Defense Architecture (SEDA)

The SEDA, as proposed by Hoeschele and Rogers in 2005, is a theoretical architecture designed to log Social Engineering (SE) attacks or attempted attacks carried out over the phone. This attack medium is the focus because Dolan (2004) and Gragg (2003) showed that most attacks are perpetrated through the phone. It also has the

potential side effect of preventing attacks. In theory, the SEDA is broken up into three parts, the first of which is a voice analysis. This creates a voice signature of the caller and converts the entire conversation into text. The next part of the SEDA is the database, which contains all the caller information and call logs. The final part is the attack detection processor. This parses the conversation in either voice or text format to detect deception attempts.

The SEDA would work by first creating a voice signature of the caller and comparing it to the database of known callers to determine if the claimed identity is correct. If it is a new caller then a new entry is created for them in the database. If the caller has changed their identity some action is taken to alert the receiver of the call. To determine the callers claimed name the conversation is converted to text from the onset. Other expositional information about the caller is also gathered from the converted conversation text. At that point, all the SEDA would do is continue to convert the conversation into text and pass this along to the attack detection part of the SEDA. If an attack is detected the caller is notified in some way.

The major premise that the SEDA depends on is the use of deception, and usually impersonation of a trusted identity, in SE attacks (Dolan, 2004). Thus if the SEDA can detect deception or anyone masquerading as someone they are not it will be able to detect a large quantity of SE attacks.

State of the Field

The current solution to Social Engineering (SE) attacks is employee training to resist such attacks coupled with a thorough security policy. As discussed earlier this solution is fundamentally flawed. A good social engineer will convince you that he or

she deserves the requested data, and you are hurting the company by not helping. Training may stop most menial or juvenile attackers, but a seasoned social engineer will never be thought of as a risk when employees talk to them, so the training will never be triggered. This can be seen with the examples of SE attacks in the Social Engineering section above.

Policy

Employing security policy is a good idea in the sense that after someone has broken the policy it is easy to display the policy and show how they violated it. However, it is not realistic to use policy alone to prevent break-ins. Security policy that classifies data into different levels of sensitivity is the most beneficial to prevent SE attacks. This provides a greater level of security for data by forcing a higher level of credentials to gain access. The end result is more work for the social engineer, which will probably deter most inexperienced and casual social engineers. However, an experienced, or more importantly, persistent social engineer will keep working until they have all the necessary credentials to obtain approval for access. Allen (2001) states the following as policy-based counter measures to SE attacks in his paper:

> Security policy - A sound security policy will ensure a clear direction on what is expected of staff within an organization.
>
> Limit Data leakage - Reducing the amount of specific data available will ensure that the attack is not an effortless exercise. (p.5)

Inoculation Training

Another type of training that could make employees less susceptible to SE attacks is inoculation training. This is important to discuss because it is a novel approach to

training and can currently be used to provide some protection against SE attacks. The training works much like vaccines where the defense system is weakly challenged but is able to overcome the challenge. The defense system strengthens it self as a result of this weak challenge. In terms of persuasion techniques this is accomplished in three steps. First the person is warned of an upcoming attack on one of their beliefs. This allows them to prepare rebuttals to potential attacks. Next the actual attack on the person's belief. The key aspect of the attack is that it must be weak enough to not actually persuade the person. The final step is to encourage the person to actively defend their position. This will both make the defense more salient and accepted by the defender. This can be applied to SE by educating employees about SE attacks, then acting out SE attack scenarios allowing employees to defend themselves. This should allow employees to come up with responses of their own to SE attacks, which will be more salient to them. This should make employees less susceptible to SE attacks (Booth-Butterfield, 2005).

This seems somewhat questionable logic, similar to other training methods. A highly skilled social engineer will never trigger the training, rendering it useless. This does not mean that inoculation training is useless, but cannot be expected to stop sophisticated attacks. It does provide a nice complement to a SEDA system not only in aiding in stopping attackers, but also helping in employee acceptance of the system.

Success of Current Solutions

Unfortunately it is impossible with the available data to determine what effect the current methods of attack prevention and detection have had on the success of SE attacks. It could be reasoned that the general lack of data shows the inadequacy of current methods of detection. If there were no Network Intrusion Detection Systems how would

one measure the amount of attacks? This inability to produce data seems to points to the current solution's inadequacy. However unlikely, it could also be a result of there not being any attacks.

Effects of Surveillance in the Workplace

To help gauge people's response to the deployment of a Social Engineering Defense Architecture (SEDA), research on surveillance and its effect on employees in the workplace was consulted. This will aid in employees acceptance of the system and mitigate their desire to subvert it. One study that examined a form of surveillance and employee's reactions to it was American Management Association's 2005 Electronic Monitoring & Surveillance Survey. This study examines Electronic Performance Monitoring (EPM) in the workplace and its effects on employees. EPM is a form of surveillance because various attributes of employee behavior are recorded, like time spent typing, time spent on the phone, both personal and business related, and reviewing of data on computer. The surveillance in this study differs from that of the SEDA, but it was the best study available.

The American Management Association (AMA, 2005) study showed that with most types of EPM, employees had a higher level of psychological strain, including boredom, tension/anxiety, depression, and anger. Job discontent and stressors also increased with the introduction of EPM. Finally, negative supervisory factors and somatic health complaints increased with EPM. Supervisory factors are comprised of problems with supervisor relations, amount of supervisor feedback, and supervisor monitoring. Somatic health complaints are comprised of shoulder soreness, neck pain into shoulder/arm/hand, back pain, racing or pounding heart, acid indigestion, headaches,

depression, and extreme anxiety. Overall the study showed that EPM increased employee discontent and dislike of the job (AMA, 2005).

As noted above, EPM surveillance records attributes of employee activity much in the same way the SEDA will. However, the difference is the purpose of the data collection. In EPM the data is collected to ascertain an employee's performance, or lack of performance. SEDA will use the data to determine if the employee is under an SE attack. The SEDA will perform much in the same way virus protection does, only for the phone line instead of the computer. Virus protection software has the ability to read files on a computer, and does so every day in some environments. This seems to be widely accepted as the 2005 CSI/FBI Computer Crime and Security Survey reports that 96% of the respondents use anti-virus software (Gordon, Loeb, Lucyshyn, & Richardson, 2005). SE attacks may or may not be as prevalent as computer viruses, but they can be just as damaging. It therefore makes sense to scan phone conversations for SE attacks in the same way we scan our computer for viruses. With this in mind, the differences between the surveillance in EPM and the SEDA are clear. Therefore it is speculated that employees would not react to the SEDA in the same way as EPM, but rather would view it more like virus protection.

A final important aspect of the SEDA employee relationship is the actual deployment method to maximize employee acceptance of such a system. Marx (1999) states that:

> Draconian surveillance, particularly when suddenly introduced in a top-down fashion, may lead to resistance and challenges. The process of introducing the surveillance may itself become an issue. Employee

> indignation is likely to be expressed at the surprise discovery that email is monitored or that a hidden video camera is in place, when there has been no prior consultation or even warning. When the surveillance itself is a form of deviance, violating reasonable expectations (as with cameras hidden in bathrooms or locker rooms or an employee posing as a friend who has infiltrated the workplace during a unionization drive) anger may be intensified. (¶ 72)

Marx (1999) further warns that:

> This can lead to attitudes such as "if you don't play fair with us, we won't play fair with you" and "turn about is fair play." To the extent that workers feel they are being treated like children (needing permission to go to the bathroom) and engineered workplaces are interpreted to symbolically say, "we don't trust you to be honest or to make reasoned choices. We expect you to behave irresponsibly, to take advantage, and to screw up unless we remove all temptation and prevent, trick or force you to do otherwise." Some workers will adopt the attitude "you've got the name, play the game." If you are thought to be untrustworthy then behave that way. This can bring new meaning to the self-fulfilling prophecy. (¶ 73)

To avoid this, the new surveillance measures should be introduced at all levels of the organization before implementation. It is important to both discuss the properties of the surveillance system, and the reasons for implementation. Everyone should also be made

aware of how data produced from the system will be used, and not be used. Company policy can be created to govern the use of the data to engender employee support and belief in the new system's real purpose.

These precautions that should be taken into consideration before implementing a SEDA have no affect on the design of the system, or the actual implementation. Those providing a SEDA to an organization must inform management of best practices for implementing such a system. However, it is the duty of the organization's management to prepare employees for the SEDA, and cannot be addressed in the design of the SEDA.

METHODS

The two different types of tests run in this study are attack detection and data parsing. The tests consisted of inputting conversation groupings into the Social Engineering Defense Architecture (SEDA) proof of concept model and analyzing the results. The two types of results were terminal output and database records.

Data Creation

Within the data set for data parsing testing there are two groupings of conversations. In the first grouping the employees follow company call policy and request the caller's name, company, and job title in that order. In the second grouping they do not follow the policy. Within the data set for attack detection testing there are nine different groupings of conversations. Each group corresponds to one type of attack and is designed to comprehensively test it. Conversation data that was used for the attack detection tests is located in Appendix B, and the data for the data parsing tests is in Appendix C.

The conversation text, SE attacks and SE data were artificially generated. An expert in the field Dr. Rogers verified all generated content to ensure it includes valid SE attacks and rated each conversation as a Social Engineering (SE) attack or benign conversation. This expert was chosen based on his number of refereed publications in the area of SE. The SE attacks that were mimicked were structured around those discussed

in the SE section of the literature review. The conversations mimicked language and content found in an educational environment from varying job areas such as system administrator, engineer, and web master. All generated conversations' validity were verified by Dr. Rogers. Not all conversations included SE attacks or violations of corporate policy; some were benign conversations. This ensured that the SEDA proof of concept model could detect both SE attacks and ignore normal conversations.

Constructs

The constructs used in this study were two different forms of SE attack signatures. These were used to determine what conversations the SEDA proof of concept model should flag as containing a SE attack. The two forms of SE attack signatures were data and signature based. Data based signatures trend contextual information gathered on the caller to detect deception, whereas the signature based method use key phrases (see Table 1). The signature based attacks are broken up into outsiders and insiders, based on which part of the conversation the phrases are scanned in. With the outsider attacks, what the outsider says is scanned for phrases that imply an attempted breach in policy or a SE attack. When scanning what the insider says, different phrases are used. An explanation of why this is done can be found in the attack detection testing section.

Table 1

Attack Signature Breakdown

Attack Grouping	Target of Attack	Description	Example
Data Based Attacks	Deceiver caller frequently	A caller repeatedly sets off other attacks.	10 other attacks in the past year.
	Impersonate identity of insider	A caller tries to impersonate an employee.	Caller gives employee's name, company, and job title.
	Impersonate identity of outsider	A caller tries to impersonate a previous caller.	Caller gives previous callers name, company, and job title.
	Change number called from frequently	A caller changes phone number frequently.	More than 5 unique phone numbers or blocked caller ID instances in past year.
	Change name	A caller changes their name.	Caller changes first or last name.
	Company changes too frequently	A caller frequently changes the company they are with.	Company name changed more than once in past year.
Signature Based Attacks: Outsider	Password	Outsider requests insider's password.	Read me your password.
	Sensitive information	Outsider requests either sensitive or restricted information from insider.	Tell me confidential info.
	Software install	Outsider requests insider install software on their computer.	Install this on your computer.
	Secure server information	Outsider requests information regarding the secure server.	Give me the secure server IP address.
Signature Based Attacks: Insider	Password	Insider reads their password to an outsider.	My password is…
	Secure server information	Insider gives information regarding the secure server.	The secure server IP address is…

Instruments

The instruments used in the evaluation and testing were the SEDA proof of concept model and SE scenarios. The SEDA proof of concept model is a software application and database designed and developed by the author. This proof of concept model was the first test of the SEDA (Hoeschele & Rogers, 2005). The SE scenarios were comprised of conversation text and an information security policy for the transmission of data over the telephone. The conversations are comprised of conversation text and data that can be obtained from caller ID information (see Appendix J). This security policy used for all of the conversation texts was a simplified academic security policy that limits the transmission of sensitive and restricted data over the phone (see Appendix H). It was modeled after Purdue University's policy for similar data transmissions (see Appendix G). Tests were conducted in Microsoft Windows. Development and actual tests were preformed using Eclipse and the Java language (see Appendix I). The database used in conjunction with the software was Microsoft Access. The Java source code for the proof of concept model software can be found in Appendix A.

Measurement

This proof of concept model does not have voice to text, voice signature generation, or advanced attack detection algorithms implemented, as a fully developed SEDA would. This was an exploratory endeavor to determine what works with regard to the proof of concept model with the goal of identifying necessary modifications to the original theoretical model.

The proof of concept model was constructed to test the SEDA in general. Specific parts that were tested are the conversation parsing, and data and signature based attack detection. These tests were carried out by the experimenter inputting conversation text into the proof of concept model and then viewing all records in the database. What was found in the database was compared to what was expected to be there. Any discrepancies were noted and classified as either a false positive or negative. Next the terminal output of the software was examined to ensure it corroborated each attack noted in the database with the correct type of attack message.

Attack Detection Testing. The SEDA proof of concept model was designed to detect eight different types of SE attacks. The first six types of attacks are triggered by trends detected in data gathered about the caller over multiple calls –data based attacks. These attacks attempt to detect the caller exhibiting potentially deceptive behavior (see Table 2).

Table 2

Data based attacks

Attack Name	Description	Thresholds	Test Cases
Habitual deceiver	A caller repeatedly sets off other attacks.	10 other attacks in the past year	14
Identity impersonation of insider	A caller tries to impersonate an employee.	Caller gives employee's name, company, and job title	20
Identity impersonation of outsider	A caller tries to impersonate a previous caller.	Caller gives previous callers name, company, and job title	20
Too many phone numbers	A caller changes phone number frequently.	More than 5 unique phone numbers or blocked caller ID instances in past year	22
Name change	A caller changes their name.	Caller changes first or last name	11
Too frequent company change	A caller frequently changes the company they are with.	Company name changed more than once in past year	11

The other two types of SE attacks that can be detected are signature based. These attacks are triggered by a phrase said during the conversation that implies attempts to breach the aforementioned security policy (see Table 3).

Table 3

Signature Based Attacks

Attack Name	Description	Thresholds	Test Cases
Outsider attacks	Phrases the caller says to the receiver of the call.	1 matching signature	20
Insider attacks	Phrases the receiver of the call says to the caller.	1 matching signature	20

When analyzing the conversation, both what the outsider and insider say are scanned for key phrases. There are two different sets of key phrases that are scanned for. One set is used on the outside of the call, and the other for the inside. An example is the request

from an outsider to an insider to read them their password. The outsider phrase would be some form of, "read me your password." Whereas the insider phrase would be some form of, "my password is" (see Appendix F). This allows for both faster processing of conversations and more intelligent attack detection. If the outsider can ask for a password without setting off one of the attacks, the SEDA can still catch the attack by analyzing what the insider says when they read the password back. Even though the insider will not be initiating or knowingly taking part in the attack, they can still respond in a way that implies that they have fallen victim to a SE attack. This is very similar to monitoring IP packets leaving a network in addition to entering to determine if an attack is in progress.

The attack detection testing for the SEDA proof of concept model was performed by inputting one of the conversation groupings into the model and comparing output in the command line and database to expected results, as previously determined by Dr. Rogers. The expert rated conversations as an SE attack based on language, key words, and key phrases used. These attributes that were used to rate conversations were derived from SE attacks and methods. The advantage to testing each type of attack separately is that it is much easier to ensure that the proof of concept model is acting correctly. It should be noted that there is one more conversation grouping than attack. This is because the "too many phone numbers" attack is tested with and without the possibility of caller ID information being blocked. When calculating attack detection accuracy in Equation 1, the total refers to the total number of conversations because as soon as an attack is detected, parsing of that conversation stops, so there can only be one attack in each conversation.

Data Parsing Testing. As a result of only having twenty test conversations, all of the data parsing test conversations were inputted to the SEDA proof of concept model at the same time. The expected data that the SEDA should parse from a conversation were determined through parsing each conversation by hand in the same manner the proof of concept model was expected to work.

Testing of the SEDA proof of concept model was measured by whether or not it detected SE attacks and stored all pertinent SE data correctly. This information was gathered by examining database records created by the proof of concept model. Each SE attack detected and data stored correctly was counted as a success. An analysis of which SE attacks and SE data the proof of concept model could and could not detect or correctly parse were recorded to aid in improving the model. For the data parsing tests of the SEDA, SE data refers to expositional information regarding the caller that is parsed out of a conversation. Specifically, each conversation has a possible four pieces of data that can be parsed, the caller's first name, last name, company, and job title. When calculating accuracy for data parsing in Equation 1, total refers to the total number of these pieces of data.

Data Analysis. Accuracy of the attack detection was determined by dividing the number of attacks detected minus the total attack false positives and false negatives, by the actual number of attacks in the conversations (see Equation 1).

$$\frac{correct - (type1 + type2)}{total} \tag{1}$$

The accuracy of the data parsing was determined using the same equation, but dividing the number of data points correctly parsed minus the total data parsing false positives and false negatives, by the number of possible data points that could exist in the conversations. The error rate for attack detection was calculated by summing the number of false positives and negatives, and then dividing that value by the number of conversations. This method was chosen because each conversation could contain an attack. The error rate for data parsing was calculated by summing the number of false positives and negatives, and dividing that value by the total number of possible data points to parse. The false positives and negatives were calculated separately for both attack and data parsing results by dividing the total of false positives or negatives by the total number of instances that could have been either an attack or data to be parsed (see Equation 2).

$$\frac{false\ \ positive\ or\ \ negative}{total} \tag{2}$$

RESULTS

Attack Detection Testing

As can be seen in Table 4 the SEDA proof of concept model achieved a 100% accuracy score for the attack detection tests. There were no false positives or negatives in the 138 tests and all conversations were correctly identified as either benign or the correct type of attack. As there were no false positives or negatives, the error rate was 0%. The terminal output from the tests can be seen in Appendix D.

Table 4
Model Accuracy

Test	Accuracy	Error Rate	False Positives	False Negatives
Attack Detection	100%	0%	0%	0%
Data Parsing (Phone policy followed)	100%	0%	0%	0%
Data Parsing (Phone policy not followed)	20%	40%	37.5%	2.5%

Test Cases of Interest. The following test conversations are of interest because, while they were correctly flagged as insider signature based attacks, they also contain outsider signature based attacks that were not identified.

> ***Outsider***: Hi this is Mike Jones from orange. I am the sales rep. just to make sure our records are up to date we need you to confirm the IP address for your secure server. Looks good. Bye.

Insider: Hi what is your name and who are you with. What is your job title? What can I do for you? The secure server's IP is twenty-three dot ten dot ten dot one. Bye.

Outsider: This is john candy from apple. I am a Sales Rep. would you like to buy some software. Great I am just going to need your password. Thanks bye.
Insider: What is your name and who are you with. What is your job title? Yes we need three copies. Ok my password is passport. Bye.

Outsider: This is Tim rose with rose industries. I am the CEO. I was told I could connect to your secure server but I do not know its name. Thanks bye.
Insider: What is your name and who are you with. What is your job title? What can I do for you? Sure the name of our secure server is jormugand. Bye.

Outsider: This is sally smith with rose industries. I am a network engineer. Can you tell me the address of your secure server? Thanks. Bye.
Insider: What is your name and who are you with. What is your job title? What can I do for you? Sure the secure server IP address is twenty-three dot ten dot ten dot one. Bye.

Outsider: Hi this is Auric Wilson with the Auric enterprise. I am a journalist. I am writing a story about hackers and was wondering if you could use the name of your secure server. Great what is it? Thanks.

Insider: What is your name and who are you with. What is your job title? What can I do for you? Sure I don't see why not. The secure server is named jourmungand. Bye.

Outsider: This is Nick Leshay with sun Microsystems. I am a computer support tech**.** to ensure you do not get locked out of you secure server I can keep you passphrase for safekeeping. Great. Thanks bye.

Insider: What is your name and who are you with. What is your job title? Great I was worried I might forget it. My passphrase is donttellanyone. Thanks bye.

Outsider: Hi this is terry Lamph from Notre dame. I am an admissions representative. I need to check some of your records with the user name and login given to me but I cannot remember the name of the server. Thanks.

Insider: What is your name and who are you with. What is your job title? What can I do for you? No problem the name of our secure server is jourmugand. Bye.

Outsider: This is jerry Johnson from orange. I am a computer engineer. Can you tell me the address of your secure server? Thanks bye.

Insider: What is your name and who are you with. Thanks and what is your job title. What can I do for you? Sure the secure server's IP is twenty-three dot ten dot ten dot one. Bye.

Data Parsing Testing

The first part of the data set where phone policy was followed resulted in the proof of concept model achieving a 100% accuracy score. There were no false positive

or false negatives and all four pieces of data were correctly parsed from each conversation, resulting in an error rate of 0%.

In the second part of the data set where phone policy was not followed, the proof of concept model achieved a 20% accuracy score. There were fifteen false positives and one false negative found. Twenty-four pieces of data were correctly parsed out of the conversations resulting in an error rate of 40% (see Table 4).

Test Cases of Interest. The following conversations are of interest because, while they do not follow company policy, they were still parsed for caller data incorrectly. As such they warrant closer inspection to improve the SEDA proof of concept model. Analysis of these conversations is located in the discussion section.

Outsider: This is Tim Jones with rose industries where I am the CEO. Please disregard my last call I had the wrong university.

Insider: Hi could you tell my your name company and job title. What can I do for you? No problem. Bye.

Outsider: Hi this is James Moore with the London financial times. I am a journalist. Do you have any more demographical data of the university population? Thanks.

Insider: What is your name and who are you with. What can I do for you? No we have already sent you everything we had. Bye.

Outsider: This is Bauer Jack Bauer. I am an operative with the counter terrorism unit. I forgot who you connected me to last time, can you tell me their name. Thanks.

Insider: What is your name and who are you with. What can I do for you? Sure it was Sean Natavaty. Bye.

Outsider: Peter Coreo here the head chef at Princeton university. Can you cancel my last order; we don't need it any more. Thanks. Bye.

Insider: What is your name and who are you with. What can I do for you? Sure. I just made a note of it. Your welcome. Bye.

Outsider: Hi this is jenny Heights from Notre Dame. I am with student services looking for a for a new student information database system. Do you have time to tell me about yours? Thanks bye.

Insider: What is your name and who are you with. What is your job title? Sorry I am pretty busy today. Bye.

Outsider: Hi this is your sales rep Samuel Pitt from rocket software. Would you like to buy some software?

Insider: Hello. What can I do for you? No thanks we are good.

Outsider: Hello I am a sales rep from orange. Oh sorry Ben Bert. Would you like to buy some software?

Insider: Hi. And what is your name. What can I do for you? No thanks we are good.

Outsider: Hi this is Justin Jones I am a sales rep from pear. Would you like a firewall? Thanks bye.

Insider: Hi what is your name and who are you with. What can I do for you? No thanks. Bye.

DISCUSSION

The two functions of the SEDA proof of concept model that were tested both performed well within the study's constraints, but not as well outside of these constraints. The constraints used can be seen in the delimitations sections above. The attack detection tests did not miss any attacks that the proof of concept model was designed to detect. However, it did miss a few outsider signature based attacks even though it correctly flagged the conversation as an attack based on insider signature based attacks. The data parsing tests resulted in a relatively high rate of failure when company phone policy was not followed but when policy was followed, the results were perfect.

As stated above, the attack detection tests missed a few outsider attacks, but still categorized the conversation as an attack based on insider attacks. This shows how the SEDA can detect attacks even without an extensive attack signature collection, as it analyzes both what the caller and receiver say. The outsider attacks were missed because the method of attack detection is only signature and data based. Attacks that can be detected through data include identity impersonation and frequent changes in name. These are considered attacks in that the caller is attempting to deceive an insider with information they provide. If the caller exhibits none of this type of behavior, currently the only way to detect an attack is by attack signatures; if the caller or receiver says a certain phrase it sets off an attack. In the case of the attacks that were missed, the

attacker requested the information, for example a password, in a way that the proof of concept model was not aware of. This can be solved either by expanding the number of attack phrases the proof of concept model is aware of, or moving away from only signature and data based attack detection by introducing a method like Natural Language Processing (Raskin, Hempelmann, & Triezenberg, 2004). This will allow a deeper understand of what ideas are being expressed in a conversation. For example, if the caller expresses the idea of giving them a password, in whatever format they choose, it will trigger an attack. This is much more difficult, but also greatly increases the comprehensiveness of attack detection.

Much like the attack detection tests, the data parsing tests had near flawless results. When the constraints, like requesting caller information at the start of the call, were placed on the study, the accuracy was 100%. The constraints greatly reduce the complexity of correctly parsing out information. However, this also introduces an apparent duplicity; the SEDA was designed as a result of employees not being able to follow company policy, but currently only functions correctly if company policy is followed. It seems that company policy in place to disallow employees from giving out certain information is far more difficult to follow than a script of how to start each phone call. It also seems that requesting such information at the start of each call will become a habit for employees. Whereas resisting giving out certain information is not practiced in each call, and requires much more thought than asking the same questions every time.

The results indicate that it is possible to parse conversation text and detect certain types of SE attacks. This conclusion is contingent upon having a conversation that was

correctly converted to text, an accurate and reproducible voice signature algorithm, and employees that followed company phone policy.

The findings have also shown that it is more difficult to parse caller information from a conversation than it is to detect actual attacks. This is because for the SE attacks to work the caller has to either deceive the receiver of the call as to their identity, or request they do something against company policy, as discussed in the review of literature section. Detecting identity deception is a simple matter once all contextual information about a caller has been correctly gathered. The SEDA just searches for changes in the data, allowing a reasonable amount to exist as a result of legitimate change. An example of legitimate change is getting a new job or company every few years. The definition of a reasonable amount of change depends on the environment the SEDA is deployed in, and can be easily altered.

Just as easily altered are the attack signatures used to catch SE attacks that request the receiver of the call to break company policy. The attack signatures work well because they can be tailored to enforce a specific security policy in addition to general SE attacks. The proficiency with which these general SE attacks are detected is a result of the attacks using key phrases, or requesting certain types of information, like passwords. This makes attack signature generation much easier. Also, consider the example where a social engineer is calling multiple employees at a company trying to gain access to certain data; once this is detected, a new rule could be introduced based on the phrases that were being used or data that was sough after.

Overall the results indicate that the proof of concept model's architecture is sound and is of merit to continue with future research and refinement of the model. The results have also shown that the theoretical foundations on which the SEDA depended for attack detection are valid and provide accurate results.

CONCLUSION

The purpose of the testing was to determine if the Social Engineering Defense Architecture (SEDA) was feasible, and to measure its accuracy. The results conclusively show that the study was a success; the SEDA design is viable and very accurate.

When the SEDA was proposed as a potential solution to Social Engineering (SE) by Hoeschele & Rogers (2005), its greatest downfall was that it was completely theoretical in nature. This study takes the SEDA out of the theoretical, and moves it much closer to being a true, applied solution to SE. The study does this by introducing a working proof of concept model, and showing that the theoretical foundations of SE attack detection that support the SEDA are valid and produce accurate attack detection. This study is important because by proving the viability of the SEDA others will become interested in conducting research in the area of security controls for countering and detecting social engineering attacks.

Two of the most impressive aspects of the current study are the comprehensive nature of the SE attack detection, and the database backend. The data based attack detection is limited by the amount of information gathered regarding each call. The SEDA proof of concept model has an attack specific to almost every piece of contextual data gathered. This includes the caller's phone number and date of the call. It is conceivable that there were other data based attacks that could be implemented, but these

are complicated and produce false positives. [Take for example someone calling different numbers in a short period of time. This could either be an attacker probing for information, or a customer looking for support].

The database backend is used by the SEDA to trend data, and detect attacks, but it also has another important role. It acts as a log of every phone call as it stores the date, time, where the call originated and terminated, and what was said by both parties. This presents a great opportunity for the application of data mining. It is also the first time there has been a way to log SE attacks. Even if the SEDA missed every attack, there would be benefit from having the logs of the calls; just like firewall or intrusion detection system logs network activity.

Future Research

As this was a proof of concept study, there are many areas that are candidates for further research. The results of this study make it clear that the data-parsing algorithm must be improved to allow more flexible and natural conversation. The current proof of concept model depends on employees following company policy and requesting caller name, company, and job title in that order. While this is reasonable for employees to do, it would be better if the SEDA could handle deviations from this policy to provide a greater degree of protection. Subsequent research should also consider improving the overall algorithm to be able to parse conversations that do not follow the company policy used in this research. It would also be advantageous to analyze the conversations that were parsed incorrectly.

Integration into a telecommunications network was avoided because of the level of complexity it added. The specific telecommunications integration features that should

be researched for the SEDA include obtaining all caller ID information and parsing voice conversations into text for analysis. Not directly associated with telecommunications integration is the ability to calculate a voice signature of the caller. This is critical for attack detection and must be implemented for the SEDA to be fully functional.

The database is an additional area for further study. The database containing the call logs was primarily used to track the output of the SEDA proof of concept model. This database not only has all the caller's information like name and company, it also contains the entire text of the conversation, both outside and inside. With such a wealth of information, data mining could aid in detecting illicit behavior of both insiders and outsiders. Consider the example where somehow a social engineer bypasses the SEDA and steals a piece of data named alpha. With data mining techniques the database could be searched for references to alpha or related data. Then the "crime scene" could be reconstructed to determine how the social engineer gained access. This could aid in making the SEDA repel these attacks, or if a legal case was opened regarding the theft, act as another form of digital evidence.

REFERENCES

Interception and disclosure of wire, oral, or electronic communications prohibited, US Department Of Justice (1994).

(AMA), American Management Association (2005). 2005 *Electronic monitoring & surveillance survey: many companies monitoring, recording, videotaping—and firing—employees*. Retrieved August 10, 2005, from http://www.amanet.org/press/amanews/ems05.htm

Allen, M. (2001). *The use of 'social engineering' as a means of violating computer systems*. Retrieved October 10, 2004, from http://www.sans.org/rr/catindex.php?cat_id=51

Arthurs, W. (2001). *A proactive defense to social engineering*. Retrieved October 10, 2004, from http://www.sans.org/rr/catindex.php?cat_id=51

Booth-Butterfield, S. (2005). *Inoculation theory [Electronic Version]*. Retrieved January 19, 2006 from http://www.as.wvu.edu/~sbb/comm221/chapters/inocul.htm.

Bort, J. (1997). *Liar, liar*. Client Server Computing, 4(5).

DePaulo, B., Stone, J., & Lassiter, G. (1985a). Deceiving and detecting deceit. In B. Schlenker (Ed.), *The Self and Social Life* (pp. pp. 323-327). New York: McGraw-Hill.

Dirani, B. (2006). *Reading and writing text files*. Retrieved October 2, 2006, from http://www.javapractices.com/Topic42.cjp

Dolan, A. (2004). *Social engineering*. Retrieved October 10, 2004, from http://www.sans.org/rr/catindex.php?cat_id=51

Ekman, P., & O'Sullivan, M. (1991). *Who can catch a liar?*. American Psychologist, 46, pp. 913-920.

Forrest, J., & Feldman, R. (2000). *Detecting deception and judge's involvement: lower task involvement leads to better lie detection*. Personality and Social Psychology Bulletin, 26(1), pp. 118-125.

Gordon, L., Loeb, M., Lucyshyn, W., & Richardson, R. (2004). *CSI/FBI computer crime and security survey*. Retrieved October 10, 2004, from http://i.cmpnet.com/gocsi/db_area/pdfs/fbi/FBI2004.pdf

Gordon, L., Loeb, M., Lucyshyn, W., & Richardson, R. (2005). *CSI/FBI computer crime and security survey*. Retrieved October 27, 2005, from http://i.cmpnet.com/gocsi/db_area/pdfs/fbi/FBI2005.pdf

Gragg, D. (2003). *A multi-level defense against social engineering*. Retrieved October 10, 2004, from http://www.sans.org/rr/catindex.php?cat_id=51

Hoeschele, M., & Rogers, M. (2004). *Detecting social engineering*. Paper presented at the Advances in Digital Forensics : IFIP International Conference on Digital Forensics, National Center for Forensic Science, Orlando, Florida.

Kraut, R. (1980). *Humans as lie detectors: some second thoughts*. Journal of Communication, 30(4), 209-214.

Marx, G., & Simpson, R. (1999). *Measuring everything that moves: the new surveillance at work*. Retrieved October 21, 2005, from http://web.mit.edu/gtmarx/www/ida6.html

Mentor, T. (1986). *The hackers manifesto*. Retrieved October 10, 2004, from http://www.geocities.com/SiliconValley/Heights/1926/mentor.html

Mitnick, K. (2003). *The art of deception: controlling the human element of security.* Indinanapolis, IN: John Wiley & Sons.

Petty, R. (1977). *A cognitive response analysis of the temporal persistence of attitude changes induced by persuasive communications*: Ohio State University, Columbus, OH.

Power, R. (2002). *CSI/FBI computer crime and security survey*. Retrieved September 10, 2004, from http://i.cmpnet.com/gocsi/db_area/pdfs/fbi/FBI2002.pdf

Raskin, V., Hempelmann, C. F., Christian, F., & Triezenberg, K. (2004). *Semantic forensics: An application of ontological semantics to information assurance*. Paper presented at the 42nd Annual Meeting of the Association for Computational Linguistics, Barcelona, Spain.

Richardson, R. (2003). *CSI/FBI computer crime and security survey*. Retrieved September 10, 2004, from http://i.cmpnet.com/gocsi/db_area/pdfs/fbi/FBI2003.pdf

Rogers, M., & Berti, J. (2002). Social engineering: The forgotten risk. In Tipton, F. & Krause, M. (Ed.), *Information Security Management Handbook* (4 ed., Vol. 3). New York: CRC Press LLC.

Digital Forensics Research Workshop (2001). *A roadmap for digital forensic research*. Paper presented at the First Digital Forensic Research Workshop, Utica, New York. August 7-8, 2001.

Zuckerman, M., & Driver, R. (1985). Telling lies: Verbal and nonverbal communication of deception. In A. W. Siegman & S. Feldstein (Eds.), *Multichannel Integrations of Nonverbal Behavior* (pp. 129-147). Hillsdale, NJ: Lawrence Erlbaum.

Appendix A

SEDA Proof of Concept Model Source Code

```
import java.io.*;
import java.sql.*;

/**
 * This is a proof of concept model of the Social Engineering Defense
 * Architecture (SEDA) proposed by Hoeschele & Rogers 2005. This model is
 * designed to test the attack detection and data parsing articulated in the
 * SEDA.  There are three command line parameters.  The first is the input data,
 * the second is the outsider attack signatures, and the third is the insider
 * attack signatures.  This version was completed on 04/11/06.
 *
 * @author Michael Hoeschele
 * @versoin 1.6
 */
public class SEDApc3 {

	// these are used for accessing the database
	static Connection con;
	static Statement stmt;
	static ResultSet rs;

	public static void main(String args[]) {

		// the number of possible conversations
		int numConvs = 200;
		int numAttacks = 100;

		boolean debug = false;

		// step 1: load driver
		loadDriver();

		// step 3: establish connection and create statement
		makeConnection();

		// number of SE attacks detected
		int SEattacks = 0;

		// total number of possible attacks
```

```
int total = 0;

// read in the conversations from the command line specified file and
// parse them into a String array
String[] contents = getContents(args[0], numConvs);

// read in the outsider attack signatures from the command line
// specified file and parse them into a String array
String[] oAttackSigs = getContents(args[1], numAttacks);

// read in the insider attack signatures from the command line specified
// file and parse them into a String array
String[] iAttackSigs = getContents(args[2], numAttacks);

// this is a 2D string array that will hold data retrieved from the DB
// on prevous calls from a specific caller
String[][] callLogs;

// this int array holds the dates corresponding to the call log entries
int[][] dates;

// if in debug mode print out reference information
if (debug) {

        // print out the data file for reference
        System.out.println("+++++++ Data Input +++++++++");
        for (int i = 0; i < numConvs; i++) {

                System.out.println(contents[i]);

                if (contents[i + 1] == null) {
                        i = numConvs;
                        System.out.println(" " + '\n');
                }
        }

        // print out the outsider attack signatures for reference
        System.out.println("+++++++ Outsider Attack Signatures
                        +++++++++");
        for (int j = 0; j < numAttacks; j++) {

                System.out.println(oAttackSigs[j]);

                if (oAttackSigs[j + 1] == null) {
                        j = numAttacks;
```

```
                        System.out.println(" " + '\n' + '\n');
                }
        }

        // print out the insider attack signatures for reference
        System.out.println("+++++++ Insider Attack Signatures
                        +++++++++");
        for (int j = 0; j < numAttacks; j++) {

                System.out.println(iAttackSigs[j]);

                if (iAttackSigs[j + 1] == null) {
                        j = numAttacks;
                        System.out.println(" " + '\n' + '\n');
                }
        }

}

// the values of the current start and end index
int startIndex, endIndex;

// var that hold caller information
String cID, vID, oNumber, iNumber, date, time, fName, lName, company,
        job, oConvo, iConvo;
String[] context = new String[3];

// is set to true if an attack is detected in the conversation
boolean attackDetected;

// loop that iterates through all the conversations in the String array
int j = 0;
while (contents[j] != null) {

        // parse out contextual information from start of call
        vID = contents[j].substring(0, 10);
        oNumber = contents[j].substring(11, 21);
        iNumber = contents[j].substring(22, 32);
        date = contents[j].substring(33, 41);
        time = contents[j].substring(42, 46);
        cID = vID + date + time;

        // this is a fixed value based on the contextual data preceding the
        // conversation text
        startIndex = 55;
```

```
                    // reset the attack detected value to false
                    attackDetected = false;

                    // parse out the outsider side of the conversation
                    startIndex = 54;
                    endIndex = contents[j].indexOf("#inside#") - 1;
                    oConvo = contents[j].substring(startIndex, endIndex);
                    oConvo = oConvo.replaceAll("'", "");

                    // parse out the insider side of the conversation
                    startIndex = endIndex + 9;
                    iConvo = contents[j].substring(startIndex);
                    iConvo = iConvo.replaceAll("'", "");

                    // remove all common words from first few sentences and leave the
                    // context info
                    context = parseContext(oConvo, iConvo);
                    fName = context[0];
                    lName = context[1];
                    company = context[2];
                    job = context[3];

                    // if in debug mode print out all caller information
                    if (debug) {

                            System.out.println("call ID: " + cID);
                            System.out.println("voice ID: " + vID);
                            System.out.println("number called from: " + oNumber);
                            System.out.println("number called: " + iNumber);
                            System.out.println("date: " + date);
                            System.out.println("time: " + time);
                            System.out.println("name: " + fName + " " + lName);
                            System.out.println("company name: " + company);
                            System.out.println("job title: " + job);
                            System.out.println("outside conversation: " + oConvo);
                            System.out.println("inside conversation: " + iConvo);
                    }

                    // store all call information in DB
                    insertContextData(cID, vID, oNumber, fName, lName, company,
                                    job, date, time, iNumber, oConvo, iConvo);

                    //
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
                    // query DB based on voice ID to determine if there is a data based
                    // attack
                    //
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

                    // check to determine if this is a habitual deceiver
                    attackDetected = previousAttacks(vID, date);
                    if (attackDetected) {
                            SEattacks++;

                            // update the call log in the DB to reflect detecting an attack
                            attackDetected(cID, "habitual deceiver");
                    }

                    // if no attacks have been detected check for ID impersonation
                    if (!attackDetected) {
                            // check for ID impersonation, either of employees or
                            // outsiders
                            int idT = idCheck(vID, fName, lName, company, job);

                            // make note in the DB of an attempt to impersonatate an
                            // insider's ID
                            if (idT == 1) {
                                    SEattacks++;
                                    attackDetected = true;
                                    attackDetected(cID, "id impersonation of insider: "
                                                    + fName
                                                    + " " + lName);
                            }

                            // make note in the DB of an attempt to impersonatate an
                            // outsiders's ID
                            else if (idT == 2) {
                                    SEattacks++;
                                    attackDetected = true;
                                    attackDetected(cID, "id impersonation of outsider: "
                                                    + fName + " " + lName);
                            }
                    }

                    // locate all information associated with the callers voice
                    // signature
                    callLogs = callerCheck(vID);

                    // parse out the dates into numerical form so they can be
```

```
// compared
dates = parseDate(callLogs);

// if no new attacks have been detected yet, check for the too many
// new numbers attack
if (!attackDetected) {
        attackDetected = newNumbers(vID, dates[0][0],
        dates[0][1]);

        // update the call log in the DB to reflect detecting an attack
        if (attackDetected) {
                SEattacks++;
                attackDetected(cID, "too many new numbers");
        }
}

// if no attack has yet been detected check for too many name
// changes attack
if (!attackDetected) {
        attackDetected = namesUsed(vID);

        // update the call log in the DB to reflect detecting an attack
        if (attackDetected) {
                SEattacks++;
                attackDetected(cID, "caller name change");
        }
}

// search for two changes in company name within the past year
// if any records are found search the data for attacks
if (callLogs != null && !attackDetected) {

        // if the person has called before look for attacks
        if (dates.length > 2) {

                attackDetected = coUsed(callLogs, dates,
                                        company);

                // update the call log in the DB to reflect detecting
                // an attack
                if (attackDetected) {
                        SEattacks++;
                        attackDetected = true;
                        attackDetected(cID, "too frequent company
                                        change");
```

```
                    }
                }
            }

            // parse the conversation for outsider attacks until an attack is
            // detected
            int k = 0;
            while (oAttackSigs[k] != null && !attackDetected) {

                // update the call log in the DB to reflect detecting an attack
                if (oConvo.indexOf(oAttackSigs[k]) != -1) {
                    attackDetected = true;
                    SEattacks++;
                    attackDetected(cID, oAttackSigs[k]);
                }

                k++;
            }

            // search for insider attacks as long as no attacks have already
            // been detected
            k = 0;
            while (iAttackSigs[k] != null && !attackDetected) {

                if (iConvo.indexOf(iAttackSigs[k]) != -1) {
                    attackDetected = true;
                    SEattacks++;

                    // update the caller's DB entry to reflect the attack
                    // detected
                    attackDetected(cID, iAttackSigs[k]);
                }

                k++;
            }

            // increment the total possible attacks
            total++;

            // increment the index in the conversation String[]
            j++;
        }

        // print out stats
```

```
		System.out.println("number of attacks found: " + SEattacks);
		System.out.println("total possible attacks: " + total);

		// close all database resources
		closeAll();
}

/**
 * this method is used to make queries to the database this is primarily for
 * testing purpose
 *
 * @param SELECT	This is the first part of the SELECT statement
 * @param FROM		This is the second part of the SELECT statement
 * @param WHERE	This is the third part of the SELECT statement
 * @return void
 */
static void retrieveData(String SELECT, String FROM, String WHERE) {
		try {

			// this builds the select statement to query the data base
			String gdta = SELECT + FROM + WHERE;

			// query the database
			rs = stmt.executeQuery(gdta);
			while (rs.next()) {
				String firstName = rs.getString("FirstName");
				String lastName = rs.getString("LastName");
				System.out.println(firstName + " " + lastName);
			}
		} catch (SQLException ex) {
			System.err.println("RetrieveData: " + ex.getMessage());
		}
}

/**
 * This method inserts all contextual data parsed from the conversation into
 * the database.
 *
 * @param cID		This is the caller ID number
 * @param vID		This is the callers voice signature
 * @param cNum		This is the callers phone number
 * @param fName		This is the callers first name
 * @param lName		This is the callers last name
 * @param company	This is the callers company
 * @param job		This is the callers job title
```

```
 * @param date                This is the date of the call
 * @param time                This is the time of the call
 * @param rNum                This is the phone number called
 * @param oConvo    This is the conversation text from the caller
 * @param iConvo    This is the conversation text from the reciever of the call
 * @return void
 */
static void insertContextData(String cID, String vID, String cNum,
                String fName, String lName, String company, String job,
                String date, String time, String rNum, String oConvo, String
                iConvo) {
        try {

                // this builds the insert statement and then applies the insert
                stmt.executeUpdate("INSERT INTO CALLINFO " + "VALUES
                                ('" + cID
                                + "', '" + vID + "', '" + cNum + "', '" + fName + "', '"
                                + lName + "', '" + company + "', '" + job + "', '" +
                                date
                                + "', '" + time + "', '" + rNum + "', " + "'none', '"
                                + oConvo + "', '" + iConvo + "')");

        } catch (SQLException ex) {
                System.err.println("InsertData: " + ex.getMessage());
        }
}

/**
 * This method checks if the caller is attempting identity impersonation.
 *
 * @param vID                This is the callers voice signature
 * @param fName              This is the callers first name
 * @param lName              This is the callers last name
 * @param company   This is the callers company
 * @param job       This is the callers job title
 * @return                   1 is returned if ID impersonation of an employee's
 *                           ID is detected. 2 is returned if ID impersonation of
 *                           an outsider's is detected. -1 is returned if no ID
 *                           impersonation is detected
 */
static int idCheck(String vID, String fName, String lName, String company,
                String job) {

        try {
                // this builds the select statement to query the callInfo table in
```

```
            // the database for the number of records that do not match the
            // voice sig, but do match the first and last name, company, and job
            // title
            rs = stmt.executeQuery("SELECT Count(*) AS ctr FROM
                                    CallInfo "
                        + "WHERE VoiceSig <> '" + vID + "' and
                        CFirstName = '"
                        + fName + "' and CLastName = '" + lName
                        + "' and Company = '" + company + "' and JobTitle
                        = '" + job + "'");

            // this returns 2 if more than one record is returned
            while (rs.next()) {
                    // if any records are returned then there was a match
                    // so return 2 for a ID impersonation of an outsider
                    if (rs.getInt("ctr") >= 1) {
                            return 2;
                    }
            }

            // this builds the select statement to query the EmpInfo table in
            // the database for the number of records that do not match the
            // voice sig, but do match the first and last name, company, and job
            // title
            rs = stmt.executeQuery("SELECT Count(*) AS ctr FROM
                                    EmpInfo "
                        + "WHERE EVoiceSig <> '" + vID + "' and
                        EFirstName = '"
                        + fName + "' and ELastName = '" + lName
                        + "' and Company = '" + company + "' and JobTitle
                        = '" + job + "'");

            // this returns 1 if more than one record is returned
            while (rs.next()) {
                    // if any records are returned then there was a match
                    // so return 1 for a ID impersonation of an employee
                    if (rs.getInt("ctr") >= 1) {
                            return 1;
                    }

            }
    }

    catch (SQLException ex) {
            System.err.println("SelectDataIDthft: " + ex.getMessage());
```

```
		}

		// no ID impersonation was detected so return -1
		return -1;
}

/**
 * This method will retrieve all data associated with a given voice signature
 *
 * @param vID		This is the voice signature that is used to query the database
 * @return		This is a 2-D String array with the retrieved call logs
 *			Null will be returned if there are no previous records found
 */
static String[][] callerCheck(String vID) {

		// an index used when storing data in 2-D array
		int i = 0;
		// find the number of previous calls from the given voice signature
		int count = callCount(vID);

		// if no previous calls are found for the given voice signature return null
		if (count == 0) {
			return null;
		}

		// build a new 2-D array to hold the records found
		// make it the length of the number of previous calls
		String[][] data = new String[count][6];

		// query the database for call logs
		try {
			rs = stmt.executeQuery("SELECT CFirstName, CLastName,
						CNumber,"
					+ " Deception, Date, Time, Company FROM
					CallInfo "
					+ "WHERE VoiceSig = '" + vID
					+ "' order by Date desc, Time desc");

			// store the retrieved data in the 2-D array
			while (rs.next()) {
				data[i][0] = rs.getString("CFirstName");
				data[i][1] = rs.getString("CLastName");
				data[i][2] = rs.getString("CNumber");
				data[i][3] = rs.getString("Date");
				data[i][4] = rs.getString("Deception");
```

```
                        data[i][5] = rs.getString("Company");

                        i++;
                }
        }

        catch (SQLException ex) {
                System.err.println("SelectDataCallerdat: " + ex.getMessage());
        }

        return data;
}

/**
 * the checks to see how many distinct numbers the caller has used over the
 * past year. If it is greater than 5 it triggers an attack
 *
 * @param vID           This is the callers voice signature
 * @param year          This is the year of the current call
 * @param month         This is the month of the current call
 * @return              True is returned if an attack is found, and false if
 *                      not
 */
static boolean newNumbers(String vID, int year, int month) {

        // this counds the number of new phone numbers given by the current
        //  caller
        int count = 0;

        // this is used to compare dates of phone numbers
        Integer temp = new Integer(year - 1);
        String nYear = temp.toString();
        temp = new Integer(month);
        String nMonth = temp.toString();

        // if number of months is less than 10 concatenate a 0
        //  in front of the nMonth string
        if (month < 10) {
                nMonth = "0" + nMonth;
        }

        // query the database for the number of unique phone numbers
        //  that match the current caller's voice signature and are a year or less old
        try {
                rs = stmt.executeQuery("SELECT DISTINCT CNumber FROM
```

```
                                CallInfo "
                        + "WHERE VoiceSig = '" + vID + "' and Date > '" +
                        nYear
                        + nMonth + "00' and CNumber <> 'xxblockedx'");

        // count the number of records returned
        while (rs.next()) {
                count++;

                // if there are 5 or more records returned
                //  (5 or more unique phone numbers in
                //  the past year) return true
                if (count >= 5) {
                        return true;
                }
        }

        // query the database for the number of times caller ID was blocked
        rs = stmt.executeQuery("SELECT Count (*) as ctr FROM CallInfo
                                "
                        + "WHERE VoiceSig = '" + vID + "' and Date > '" +
                        nYear
                        + nMonth + "00' and CNumber = 'xxblockedx'");

        // add the number of times caller ID was blocked
        //  to the number of new phone numbers
        while (rs.next()) {
                count = count + rs.getInt("ctr");

                // if the count of new phone number and
                //  caller IDs blocked is 5 or more return true
                if (count >= 5) {
                        return true;
                }

        }
        return false;
}

catch (SQLException ex) {
        System.err.println("SelectDataNewNumbers: " +
        ex.getMessage());
}

return false;
```

```
}

/**
 * This method checks to see how many times the caller has set off an attack
 *  flag in the past year.  If more than 10 are found an attack is triggered.
 *
 * @param vID              This is the voice signature of the caller
 * @param date             This is the date of the current call
 * @return                 True is returned if an attack is detected, and false if
 *                         not
 */
static boolean previousAttacks(String vID, String date) {

        // this is used to convert the data to a year less than the current date
        int convert;
        convert = ((int) date.charAt(3)) - 49;
        date = date.substring(0, 3) + (new Integer(convert).toString())
                        + date.substring(4);

        // query the database for the number of times the caller has triggered an
        // attack other than habitual deceiver
        try {
                rs = stmt
                                .executeQuery("SELECT Count(*) as ctr FROM
                                                CallInfo "
                                                + "WHERE VoiceSig = '" + vID
                                                + "' and Deception <> 'none' and
                                                Deception <> 'habitual deceiver' and
                                                Date >= '"
                                                + date + "'");
                while (rs.next()) {

                        // if the number of attacks other than habitual deceiver is 10
                        // or more trigger a habitual deceiver attack
                        if (rs.getInt("ctr") >= 10) {

                                return true;
                        }
                        return false;
                }
                return false;
        }

        catch (SQLException ex) {
                System.err.println("SelectDataPreviousAt: " + ex.getMessage());
```

```
        }

        return false;
}

/**
 * This method check to see how many time the caller has changed companies.
 *  If they have changed more than XX times in the past year trigger an attack.
 *
 * @param callLogs   This is all of the call logs
 * @param dates              This is all of the call dates
 * @param company   The callers company
 * @return                   If an attack is detected return true, if not return false
 */
static boolean coUsed(String[][] callLogs, int[][] dates, String company) {

        // this is used to find calls in the last year
        int months = 0;

        // Iterate through the call logs
        for (int p = 0; p < callLogs.length; p++) {

                // now actually look for attacks
                months = (dates[0][1] + ((dates[0][0] - dates[p][0]) * 12))
                                - dates[p][1];

                // if the caller has called within 24 months
                // and their company name has changed since then
                // and their company name was different before
                // that call, then they are attacking
                if (months <= 24 && (!company.equals(callLogs[p][5]))) {

                        for (int q = p + 1; q < callLogs.length - 1; q++) {
                                months = (dates[p][1] + ((dates[p][0] - dates[q][0])
                                        * 12))
                                                - dates[q][1];

                                if (months <= 24
                                                &&
                                   (!callLogs[p][5].equals(callLogs[q][5]))) {
                                        return true;
                                }
                        }
                }
```

```
        }

        return false;
}

/**
 * This method determnines if the caller has ever used a different name,
 *  not including when they don't give a first or last name, ir no name is found
 *
 * @param vID               This is the callers voice signature
 * @return                  True is returned if an attack is detected, and false if
 *                          not
 */
static boolean namesUsed(String vID) {

        // this is used to count the number of unique names given by a caller
        int count = 0;

        // query the database for the all unique first names associated with a voice
        //  signature
        try {
                rs = stmt
                                .executeQuery("SELECT DISTINCT CFirstName
                                                FROM CallInfo "
                                                + "WHERE VoiceSig = '" + vID
                                                + "' and CFirstName <> 'none'");

                // count the number of records returned
                while (rs.next()) {
                        count++;

                        // if the caller has ever changed their name flag an attack
                        if (count > 1) {
                                return true;
                        }
                }

                // query the database for all unique last names associated wtih a
                //  voice signature
                rs = stmt.executeQuery("SELECT DISTINCT CLastName FROM
                                        CallInfo "
                                + "WHERE VoiceSig = '" + vID + "' and
                                CLastName <> 'none'");

                // count the nubmber of records returned
```

```
                while (rs.next()) {
                        count++;

                        // if the caller has ever changed their name flag an attack
                        if (count > 2) {
                                return true;
                        }
                }
        }

        catch (SQLException ex) {
                System.err.println("SelectDataNamesUsed: " + ex.getMessage());
        }

        return false;
}

/**
 * This method determines the number of calls associated with
 *  a given voice signature.
 *
 * @param vID          This is the callers voice signature
 * @return             Return the number of rows that will be returned for
 *                             a given voice signature
 */
static int callCount(String vID) {

        // this is used to cound the number of calls found
        int count = 0;

        // query the database for the number of calls associated with the given
        // voice signature
        try {
                rs = stmt.executeQuery("SELECT count(*) as ctr FROM CallInfo
                                                "
                                + "WHERE VoiceSig = '" + vID + "'");

                // extract the number of calls from the query results
                while (rs.next()) {
                        count = rs.getInt("ctr");
                }
        }

        catch (SQLException ex) {
                System.err.println("SelectDataCallCnt: " + ex.getMessage());
```

```
        }

        return count;
}

/**
 * This method updates the current call record in the database to reflect
 *  the attack that was detected.  It prints a message including the unique
 *  call ID and the attack detected.
 *
 * @param cID                   This is the unique call ID
 * @param attack        This is the attack that was triggered
 */
static void attackDetected(String cID, String attack) {

        // print out that an attack was detected
        System.out.println("++++++++++++++++++ Attack Detected
                                ++++++++++++++");
        System.out.println("Attack: " + attack);
        System.out.println("Call ID: " + cID + '\n');

        // insert an update to the table callInfo for the entry cID in the
        // deception field
        try {
                stmt.executeUpdate("UPDATE CALLINFO SET DECEPTION =
                                        '" + attack
                                + "' WHERE CALLID = '" + cID + "'");
        } catch (SQLException ex) {
                System.err.println("UpdateData: " + ex.getMessage());
        }

}

/**
 * This method removes all words form a string except possible caller names,
 *  company, and job titles.
 *
 * @param toParse      This is the string to parse contextual information out of
 * @param inConvo    This is what the employee says in the conversation
 * @return                This is all of the context information parsed from the call
 */
public static String[] parseContext(String toParse, String inConvo) {

        // these are used to parse context out of the conversatoin
        String[] parsed = new String[4];
```

```
String temp = "error";
int first = toParse.indexOf(".");
int second = toParse.indexOf(".", first + 1);
int start, end;

// this is part of company policy
// these are used to check if company policy was followed
String nameQ = "what is your name";
String compQ1 = "who are you with";
String compQ2 = "what company do you work for";
String jobQ = "what is your job";
int nQindex = inConvo.indexOf(nameQ);
int cQindex = inConvo.indexOf(compQ1);

// if company policy is followed
if (cQindex == -1) {
        cQindex = inConvo.indexOf(compQ2);
}
int jQindex = inConvo.indexOf(jobQ);

// parse the context out if the employee does not ask for the caller's
// name, company, or job
// or they ask for it in the order: name, company, job
boolean noneFound = nQindex == -1 && cQindex == -1 && jQindex ==
                        -1;
boolean rightOrder = nQindex != -1
                && (nQindex < cQindex && cQindex < jQindex);

// if company policy is followed, or none of the questions are asked
if (noneFound || (nQindex != -1 && rightOrder)) {
        // parse out the caller information from the conversation
        temp = toParse.substring(0, first);
        temp = temp.replaceAll(" this ", " ");
        temp = temp.replaceAll(" hi ", " ");
        temp = temp.replaceAll(" is ", " ");
        temp = temp.replaceAll(" my ", " ");
        temp = temp.replaceAll(" name ", " ");
        temp = temp.replaceAll(" i ", " ");
        temp = temp.replaceAll(" am ", " ");
        temp = temp.replaceAll(" here", " ");
        temp = temp.replaceAll(" hello ", " ");
        temp = temp.replaceAll(" the ", " ");
        temp = temp.replaceAll(" an ", " ");
        temp = temp.replaceAll(" where", " ");
```

```
            // store the name in the first position of the parsed array,
            // and company in the third position.
            end = temp.indexOf(" with ");
            start = end + 6;
            if (end == -1) {
                  end = temp.indexOf(" from ");
                  start = end + 6;
            }
            if (end == -1) {
                  end = temp.indexOf(" at ");
                  start = end + 4;
            }
            if (end == -1) {
                  end = temp.indexOf(" of ");
                  start = end + 4;
            }
            if (end == -1) {
                  parsed[0] = "error";
                  parsed[2] = "error";
            } else {
                  parsed[0] = temp.substring(1, end);
                  parsed[2] = temp.substring(start);
            }

            // parse job title and store it in the fourth position of the parsed
            // array
            temp = toParse.substring(first + 1, second);
            temp = temp.replaceAll(" i ", " ");
            temp = temp.replaceAll(" am ", " ");
            temp = temp.replaceAll(" with ", " ");
            temp = temp.replaceAll(" my ", " ");
            temp = temp.replaceAll(" is ", " ");
            temp = temp.replaceAll(" a ", " ");
            temp = temp.replaceAll(" job ", " ");
            temp = temp.replaceAll(" title ", " ");
            temp = temp.replaceAll(" position ", " ");
            temp = temp.replaceAll(" the ", " ");
            temp = temp.replaceAll(" an ", " ");
            temp = temp.replaceFirst(" ", "");

            parsed[3] = temp;

      }

      // else if the caller is not asked to give their information, or are
```

```
// asked in the wrong order, try to find the name in the text
else {

        temp = toParse;

        // look for the name based on what could be said before the name
        // add more to be more comprehensive
        if (-1 != (start = temp.indexOf("my name is "))) {
                start = start + 11;
        } else if (-1 != (start = temp.indexOf("this is "))) {
                start = start + 8;
        } else if (-1 != (start = temp.indexOf("i am "))) {
                start = start + 5;
        } else {
                start = 0;
        }

        // make a new string that is the first two words after the above key
        // string
        end = temp.indexOf(" ", temp.indexOf(" ", start) + 1);
        parsed[0] = temp.substring(start, end);

        // look for the company based on what could have been said before
        //  it
        if (-1 != (start = temp.indexOf(" from "))) {
                start = start + 6;
        } else if (-1 != (start = temp.indexOf(" with "))) {
                start = start + 6;
        } else if (-1 != (start = temp.indexOf(" i work at "))) {
                start = start + 11;
        } else if (-1 != (start = temp.indexOf(" at "))) {
                start = start + 4;
        } else {
                start = 0;
        }

        // make a new string that is the first three work after the above
        // key strings
        temp = temp.substring(start);
        start = 0;
        if (-1 != (end = temp.indexOf(" as the "))) {
        } else if (-1 != (end = temp.indexOf(" as "))) {
        } else if (-1 != (end = temp.indexOf(" where "))) {
        } else if (-1 != (end = temp.indexOf("."))) {
        }
```

```
                parsed[2] = temp.substring(start, end);

                // look for the job based on what could have been said before it
                if (-1 != (start = temp.indexOf("i am a "))) {
                        start = start + 7;
                } else if (-1 != (start = temp.indexOf(" as a "))) {
                        start = start + 6;
                } else if (-1 != (start = temp.indexOf("i am the "))) {
                        start = start + 9;
                } else {
                        start = 0;
                }

                // make a new string that is starts with the job title after the
                // above key strings
                temp = temp.substring(start);
                start = 0;
                if (-1 != (end = temp.indexOf(" with"))) {
                } else if (-1 != (end = temp.indexOf(" at"))) {
                } else if (-1 != (end = temp.indexOf("."))) {
                }

                parsed[3] = temp.substring(start, end);

        }

        // if a first and last name were found seperate them and put the last
        // name in the second position of the array
        int x = parsed[0].indexOf(" ");
        if (x != -1) {
                temp = parsed[0].substring(0, x);
                parsed[1] = parsed[0].substring(x + 1);
                parsed[0] = temp;
        } else {
                parsed[1] = "none";
        }

        return parsed;
}

// parse out the date from the 2D string array
/**
 * This method parses out the dates from the given 2-D string array
 *  and returns them in a 2-D int array.
 *
```

```
     * @param toParse     This is information to parse the date out of
     * @return            This is the 2-D int array used to hold the parsed dates
     */
    public static int[][] parseDate(String[][] toParse) {

            // This holds the parsed dates to be returned
            int[][] parsed = new int[toParse.length][3];

            // these are used to convert and parse the dates
            String temp;
            char[] convert;
            int place;

            // walk through the toParse array converting the dates
            for (int p = 0; p < toParse.length; p++) {
                    temp = toParse[p][3];
                    convert = temp.toCharArray();

                    place = 1;
                    for (int i = 3; i >= 0; i--) {

                            // parse out the year
                            parsed[p][0] = parsed[p][0] + (((int) convert[i] - 48) *
                                            place);
                            place = place * 10;
                    }

                    place = 1;
                    for (int i = 5; i >= 4; i--) {
                            // parse out the month
                            parsed[p][1] = parsed[p][1] + (((int) convert[i] - 48) *
                                            place);

                            // parse out the day
                            parsed[p][2] = parsed[p][2]
                                            + (((int) convert[i + 2] - 48) * place);

                            place = place * 10;
                    }

            }

            return parsed;
    }
```

```
/**
 * This method loads the database driver
 *
 *@return void
 */
static void loadDriver() {
	try {
		// define connection URL
		Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
	} catch (java.lang.ClassNotFoundException e) {
		System.err.print("ClassNotFoundException: ");
		System.err.println(e.getMessage());
	}
}

/**
 * This method makes a connection to the database
 *
 *@return void
 */
static void makeConnection() {
	// for how to set up data source name see below.
	String dsn = "SEDAdb";
	String url = "jdbc:odbc:" + dsn;
	try {
		con = DriverManager.getConnection(url, "", "");
	} catch (SQLException ex) {
		System.err.println("database connection: " + ex.getMessage());
	}

	// create a statement that can be used to query the database
	try {
		stmt = con.createStatement();
	} catch (SQLException ex) {
		System.err.println("CreateTable: " + ex.getMessage());
	}

}

/**
 * This method closes all database connections
 *
 *@return void
 */
static void closeAll() {
```

```
		// close all databases
		try {
			stmt.close();
			con.close();
		} catch (SQLException ex) {
			System.err.println("closeAll: " + ex.getMessage());
		}
}


/**
 * This method reads in a file and places each line of the file in a section of a
 * String array.  The method is modified from code retrieved from
 * http://www.javapractices.com/Topic42.cjp on 2/10/2006 with permission
 * of the author.
 *
 * @param file		This is the name of the file to read in
 * @param arrayLen	This is the length of the String array to return
 * @return		This is the String array with all each line of the file in a
 *			spot in the array
 */
public static String[] getContents(String file, int arrayLen) {

		// where the file will be stored
		String[] contents = new String[arrayLen];

		// declared here to make visible to finally clause
		BufferedReader input = null;
		try {
			// the reads in one line of the file at a time
			input = new BufferedReader(new FileReader(file));
			String line = null;
			int i = 0;
			while ((line = input.readLine()) != null) {
				contents[i] = line.toLowerCase();
				i++;
			}
		} catch (FileNotFoundException ex) {
			ex.printStackTrace();
		} catch (IOException ex) {
			ex.printStackTrace();
		} finally {
			try {
				if (input != null) {
```

```
                                        // flush and close both "input" and its underlying
                                        //  FileReader
                                        input.close();
                                }
                        } catch (IOException ex) {
                                ex.printStackTrace();
                        }
                }

                return contents;
        }
}
```

Appendix B

Attack Detection Data Set

Signature based attacks

Outsider Attacks

There are twenty conversations with nine attacks. The numbers of the conversations with "outsider" attacks are 3, 4, 7, 9, 10, 11, 15, 18, and 20.

1. 4444444444 6094660035 9082456893 20050215 2050 #start# This is mike smith from orange. I am a sales rep. would you like to buy some software. #inside# what is your name and who are you with. thanks and what is your job title. what can I do for you. No thanks we are good.
2. 5555555555 6094660036 9082456847 20050221 1254 #start# Hi this is mike jones from orange. I am the sales rep. would you like a firewall. thanks bye #inside# hi what is your name and who are you with. what is your job title. what can I do for you. no thanks. bye.
3. 3333333333 6094660038 9082456829 20050422 1315 #start# I am john doe from apple computers where I am a sales rep. could you read me your passcode to make sure our records are up to date. thanks. bye. #inside# what is your name and who are you with. what can I do for you. ok it is one two three four five. bye.
4. 2222222222 6094660035 9082456896 20050515 1150 #start# This is john candy from apple. I am a computer support tech. to keep you machine up to date I need you to install this on your computer. thanks bye. #inside# what is your name and who are you with. what is your job title. what can I do for you. ok I have it installed. your welcome. bye.
5. 1111111111 4879301873 9082456894 20050524 1433 #start# hi this is moe sislack from moes tavern. I am the bartender. I am interested in advertising on campus. ok thanks. #inside# what is your name and who are you with. what is your job title. what can I do for you. sorry that is not allowed. bye.
6. 6666666666 4879301856 9082456895 20050605 0834 #start# this is lisa coreo with princeton university. I am a chef. I need to purchase six hundred yams. thanks. #inside# what is your name and who are you with. what is your job title. what can I do for you. ok I will send them out tomorrow. bye
7. 9999999999 4879301867 9082456898 20050822 1436 #start# this is tim rose with rose industries. I am a security officer. I need you to read me your password so I can make sure it is secure. thanks bye. #inside# what is your name and who are you with. what is your job title. sure it is password one two three. bye.
8. 1010101010 4879301889 9082456845 20051012 0948 #start# I am willard white from willard whilte industries. I am the owner. I am interested in donating to the university. please send me the details. bye. #inside# what is your name and who

are you with. what is your job title. what can I do for you. great I will send out a packet today. thanks bye.

9. 8888888888 4879301254 9082456878 20051111 1551 #start# hello this is james stock of the london financial times. I am a journalist. can you tell me some confidential information. thanks. bye. #inside# what is your name and who are you with. what is your job title. what can I do for you. sorry I am not allowed to do that. bye.
10. 7777777777 4879301874 9082456824 20051228 1421 #start# this is jack bauer with the c t u. I am an operative. I need you to give me confidential information regarding margot fitzsimmons. thanks. #inside# what is your name and who are you with. what is your job title. what can I do for you. we do not have any information on her. bye.
11. 2444444444 4879301811 9082456889 20060205 1706 #start# this is sally smith with rose industries. I am a purchaser. before placing an order can you give me some private information as to the manufacturing process. thanks bye. #inside# what is your name and who are you with. what is your job title. what can I do for you. well the only part that is private is that we use carbon fiber. bye.
12. 2555555555 4879301800 9082456836 20060209 1219 #start# this is willard white from ww industries. I am the owner. I decided I would like to name my building blowfeld labs. great. thanks. #inside# what is your name and who are you with. what is your job title. what can I do for you. I will be sure to make the changes. bye.
13. 2333333333 4879301820 9082456890 20060210 1041 #start# hi this is auric wilson with the auric enterprise. I am a journalist. do you have any more demographical data of the university population. thanks. #inside# what is your name and who are you with. what is your job title. what can I do for you. no we have already sent you everything we had. bye.
14. 2122222222 4879301890 9082456894 20060212 0933 #start# this is nick leshay with sun microsytems. I am an operative. I forgot who you connected me to last time, can you tell me their name. thanks. #inside# what is your name and who are you with. what is your job title. what can I do for you. sure it was sean natavaty. bye.
15. 2111111111 4879301833 9082456896 20060215 1631 #start# hello this is amanda hill with verizon. I am sales rep. we are updating our records and noticed we do not have your passcode for you mailbox. can you please read me your passcode. thanks. #inside# what is your name and who are you with. what is your job title. what can I do for you. oh do I need to find it. ok it is two seven nine five. bye.
16. 2666666666 6094660035 9082456830 20060416 0842 #start# hi this is terry lamph from notre dame. I am an admissions representative. do you have any informatoin regarding the new s a t tests. thanks bye. #inside# what is your name and who are you with. what is your job title. what can I do for you. no we have not recieved any. bye.
17. 2777777777 6094660035 9082456867 20060518 1403 #start# This is samuel blake from apple. I am a sales rep. would you like to buy some software. ok bye.

#inside# what is your name and who are you with. what is your job title. what can I do for you. No thanks we are good.

18. 2999999999 6094660035 9082456890 20060520 1126 #start# This is jerry johnson from orange. I am a sales rep. can you tell me the ip of your secure server so I can make sure you have the newest model. thanks. bye. #inside# what is your name and who are you with. thanks and what is your job title. what can I do for you. sure it is twenty three dot ten dot ten dot one. bye.
19. 2010101010 6094660035 9082456892 20060819 1638 #start# Hi this is mike jones from pear. I am the sales rep. would you like a firewall. thanks bye #inside# hi what is your name and who are you with. what is your job title. what can I do for you. no thanks. bye.
20. 2888888888 6094660035 9082456891 20061015 0952 #start# I am john doe from apple where I am a sales rep. can you give me the name of your secure server so I can make sure you have the newest model. thanks. bye. #inside# what is your name and who are you with. thanks and what is your job title. what can I do for you. sure it is jorgumand. bye.

Insider Attacks

There are twenty conversations with eight attacks. The numbers of the conversations with "insider" attacks are 2, 4, 7, 11, 13, 14, 16, and 18.

1. 4444444444 6094660035 9082456893 20050215 2050 #start# This is mike smith from orange. I am a Sales Rep. would you like to buy some software. #inside# what is your name and who are you with. thanks and what is your job title. what can I do for you. No thanks we are good.
2. 5555555555 6094660036 9082456847 20050221 1254 #start# Hi this is mike jones from orange. I am the sales rep. just to make sure our records are up to date we need you to confirm the ip address for your secure server. looks good. bye. #inside# hi what is your name and who are you with. what is your job title. what can I do for you. the secure servers ip is twenty three dot ten dot ten dot one. bye.
3. 3333333333 6094660038 9082456829 20050422 1315 #start# I am john doe from apple computers where I am a sales rep. would you like to buy some software. great. #inside# what is your name and who are you with. what can I do for you. Sure I will take ten.
4. 2222222222 6094660035 9082456896 20050515 1150 #start# This is john candy from apple. I am a Sales Rep. would you like to buy some software. great I am just going to need your password. thanks bye. #inside# what is your name and who are you with. what is your job title. yes we need three copies. ok my password is passport. bye.
5. 1111111111 4879301873 9082456894 20050524 1433 #start# hi this is moe sislack from moes tavern. I am the bartender. I am interested in advertising on campus. thanks bye. #inside# what is your name and who are you with. what is your job title. what can I do for you. sorry that is not allowed. bye.

6. 6666666666 4879301856 9082456895 20050605 0834 #start# this is lisa coreo with princeton university. I am a chef. I need to purchase six hundred yams. thanks. #inside# what is your name and who are you with. what is your job title. what can I do for you. ok I will send them out tomorrow. bye
7. 9999999999 4879301867 9082456898 20050822 1436 #start# this is tim rose with rose industries. I am the ceo. I was told I could connect to your secure server but I do not know its name. thanks bye. #inside# what is your name and who are you with. what is your job title. what can I do for you. sure the name of our secure server is jormugand. bye.
8. 1010101010 4879301889 9082456845 20051012 0948 #start# I am willard white from willard whilte industries. I am the owner. I am interested in donating to the university. please send me the details. bye. #inside# what is your name and who are you with. what is your job title. what can I do for you. great I will send out a packet today. thanks bye.
9. 8888888888 4879301254 9082456878 20051111 1551 #start# hello this is james stock of the london financial times. I am a journalist. I was wondering if I could schedule an interview with president jiskey. ok thanks. #inside# what is your name and who are you with. what is your job title. what can I do for you. sorry he has no free time slots for the next year. your welcome bye.
10. 7777777777 4879301874 9082456824 20051228 1421 #start# this is jack bauer with the ctu. I am an operative. Can you connect me to serious for help subverting a security system. thanks. #inside# what is your name and who are you with. what is your job title. what can I do for you. sure please hold.
11. 2444444444 4879301811 9082456889 20060205 1706 #start# this is sally smith with rose industries. I am a network engineer. can you tell me the address of your secure server. thanks. bye. #inside# what is your name and who are you with. what is your job title. what can I do for you. sure the secure server ip address is twenty three dot ten dot ten dot one. bye.
12. 2555555555 4879301800 9082456836 20060209 1219 #start# this is willard white from ww industries. I am the owner. I decided I would like to name my building blowfeld labs. great. thanks. #inside# what is your name and who are you with. what is your job title. what can I do for you. I will be sure to make the changes. bye.
13. 2333333333 4879301820 9082456890 20060210 1041 #start# hi this is auric wilson with the auric enterprise. I am a journalist. I am writing a story about hackers and was wondering if you could use the name of your secure server. great what is it. thanks. #inside# what is your name and who are you with. what is your job title. what can I do for you. sure I dont see why not. the secure server is named jourmungand. bye.
14. 2122222222 4879301890 9082456894 20060212 0933 #start# this is nick leshay with sun microsytems. I am an computer support tech. to ensure you do not get locked out of you secure server I can keep you passphrase for safe keeping. great. thanks bye. #inside# what is your name and who are you with. what is your job title. great I was worried I might forget it. my passphrase is donttellanyone. thanks bye.

15. 2111111111 4879301833 9082456896 20060215 1631 #start# hello this is amanda hill with verizon. I am sales rep. can you cancel my last order, we do not need it any more. thanks. bye. #inside# what is your name and who are you with. what is your job title. what can I do for you. sure. I just made a note of it. your welcome. bye.
16. 2666666666 6094660035 9082456830 20060416 0842 #start# hi this is terry lamph from notre dame. I am an admissions representative. I need to check some of your records with the user name and login given to me but I can not remember the name of the server. thanks. #inside# what is your name and who are you with. what is your job title. what can I do for you. no problem the name of our secure server is jourmugand. bye.
17. 2777777777 6094660035 9082456867 20060518 1403 #start# This is samuel blake from apple. I am a Sales Rep. would you like to buy some software. #inside# what is your name and who are you with. what is your job title. what can I do for you. No thanks we are good.
18. 2999999999 6094660035 9082456890 20060520 1126 #start# This is jerry johnson from orange. I am a computer engineer. can you tell me the address of your secure server. thanks bye. #inside# what is your name and who are you with. thanks and what is your job title. what can I do for you. sure the secure servers ip is twenty three dot ten dot ten dot one. bye.
19. 2010101010 6094660035 9082456892 20060819 1638 #start# Hi this is mike jones from pear. I am the sales rep. would you like a firewall. thanks bye #inside# hi what is your name and who are you with. what is your job title. what can I do for you. no thanks. bye.
20. 2888888888 6094660035 9082456891 20061015 0952 #start# I am john doe from apple where I am a sales rep. would you like to buy some software. great. #inside# what is your name and who are you with. what can I do for you. Sure I will take ten.

Data Based Attacks

Habitual Deceiver

There are fourteen calls. Signature based attacks are used to build up the number of deceptions to ten, which sets off the “habitual attacker” attack. These attacks are triggered in the 12 and 14 calls. There are also eleven other signature based attacks.

1. 1111111111 8542359874 9082456657 20050101 1048 #start# hi this is ned flanders from notre dame. I am a help desk technician. can you read me your password. no not today. bye. #inside# hi what is your name and who are you with. what is your job title. sure it is donttellanyone. would you like my social security number too. ok bye.

2. 1111111111 8542359874 9082456657 20050223 0926 #start# hi this is ned flanders from notre dame. I am a help desk technician. would you give me your phone number. no not today. bye. #inside# hi what is your name and who are you with. what is your job title. sure it is donttellanyone. would you like my name. ok bye.
3. 1111111111 8542359874 9082456657 20050311 1432 #start# hi this is ned flanders from notre dame. I am a help desk technician. could you tell me your passcode. no not today. bye. #inside# hi what is your name and who are you with. what is your job title. sure it is donttellanyone. would you like my social security number too. ok bye.
4. 1111111111 8542359874 9082456657 20050417 1655 #start# hi this is ned flanders from notre dame. I am a help desk technician. can you give me some confidential information on student joe smith. ok thanks. bye. #inside# hi what is your name and who are you with. what is your job title. well we are not supposed to give that out. no problem bye.
5. 1111111111 8542359874 9082456657 20050505 1727 #start# hi this is ned flanders from notre dame. I am a help desk technician. would you please install this on your computer for me. nope that is all today. bye. #inside# hi what is your name and who are you with. what is your job title. sure just a second. is there anything else I should do. ok bye.
6. 1111111111 8542359874 9082456657 20050603 1234 #start# hi this is ned flanders from notre dame. I am a help desk technician. i need you to give me your password. bye. #inside# hi what is your name and who are you with. what is your job title. sure it is donttellanyone. ok bye.
7. 1111111111 8542359874 9082456657 20050724 0847 #start# hi this is ned flanders from notre dame. I am a help desk technician. could you read me your passphrase. no not today. bye. #inside# hi what is your name and who are you with. what is your job title. sure it is donttellanyone. would you like my social security number too. ok bye.
8. 1111111111 8542359874 9082456657 20050826 1138 #start# hi this is ned flanders from notre dame. I am a help desk technician. can you give me some confidential information on student terry jones. ok thanks. bye. #inside# hi what is your name and who are you with. what is your job title. well we are not supposed to give that out. no problem bye.
9. 1111111111 8542359874 9082456657 20050907 1349 #start# hi this is ned flanders from notre dame. I am a help desk technician. would you tell me your passcode. thanks bye. #inside# hi what is your name and who are you with. what is your job title. sure it is donttellanyone. ok bye.
10. 1111111111 8542359874 9082456657 20051019 1150 #start# hi this is ned flanders from notre dame. I am a help desk technician. i need you to install this on your computer for me. nope that is all today. bye. #inside# hi what is your name and who are you with. what is your job title. sure just a second. is there anything else I should do. ok bye.
11. 1111111111 8542359874 9082456657 20051124 0937 #start# hi this is ned flanders from notre dame. I am a help desk technician. would you read me your

password. no not today. bye. #inside# hi what is your name and who are you with. what is your job title. sure it is donttellanyone. would you like my social security number too. ok bye.

12. 1111111111 8542359874 9082456657 20051228 1438 #start# hi this is ned flanders from notre dame. I am a help desk technician. can you read me your phone number. no not today. bye. #inside# hi what is your name and who are you with. what is your job title. sure it is eight five four two three five nine eight seven four. would you like my name. ok bye.
13. 1111111111 8542359874 9082456657 20060105 1014 #start# hi this is ned flanders from notre dame. I am a help desk technician. could you read me your password. no not today. bye. #inside# hi what is your name and who are you with. what is your job title. sure it is donttellanyone. would you like my social security number too. ok bye.
14. 1111111111 8542359874 9082456657 20060216 1208 #start# hi this is ned flanders from notre dame. I am a help desk technician. would you give me your passphrase. bye. #inside# hi what is your name and who are you with. what is your job title. sure it is donttellanyone. ok bye.

ID Impersonation: Insider

There are twenty calls. The first ten should set off "id impersonation insider" attacks, and the last ten should not.

1. 5555555555 6094660036 9082456847 20060202 1254 #start# Hi this is michael hoeschele from purdue university. I am the security analyst. can you send me one of your white papers on security. Thanks. bye. #inside# hi what is your name and who are you with. what is your job title. what can I do for you. sure I will send one out today. bye.
2. 2222222222 6094660036 9082456847 20060204 1135 #start# Hi this is margot fitzsimmons from purdue university. I am a public relations agent. I wanted to confirm that I know of any potentially embarrassing events that have occurred recently. thanks bye. #inside# hi what is your name and who are you with. what is your job title. what can I do for you. there are none that I know of. bye.
3. 3333333333 6094660036 9082456847 20060208 0843 #start# Hi this is trevor smith from purdue university. I am a consultant. I would like to change the address that my paychecks are sent to. just the house number to six. thanks bye. #inside# hi what is your name and who are you with. what is your job title. what can I do for you. sure I have noted the change. bye.
4. 4444444444 6094660036 9082456847 20060210 1434 #start# Hi this is emily thurston from purdue university. I am a travel agent. I need the billing number for your department. thanks bye. #inside# hi what is your name and who are you with. what is your job title. what can I do for you. ok it is sixty three fifty nine eight. bye.
5. 8888888888 6094660036 9082456847 20060210 1658 #start# Hi this is claire foster from purdue university. I am a disaster recovery specialist. I have an

updated sop for your department. I will send it over today. thanks bye #inside# hi what is your name and who are you with. what is your job title. what can I do for you. ok I will look for it. bye.

6. 6666666666 6094660036 9082456847 20060211 1744 #start# Hi this is doug rose from purdue university. I am an engineer. I need you to tell me how to access webmail. thanks bye. #inside# hi what is your name and who are you with. what is your job title. what can I do for you. sure just do type w w w dot webmail dot com. bye.
7. 1111111111 6094660036 9082456847 20060213 0922 #start# Hi this is greg hoeschele from purdue university. I am an engineer. can you read me the specs for the alpha project. thanks. #inside# hi what is your name and who are you with. what is your job title. how can I help you. sure I can send those over. not at all bye.
8. 7777777777 6094660036 9082456847 20060224 1015 #start# Hi this is kevin hoeschele from purdue university. I am a system administrator. can you help me gain access to our password generation software. thanks. #inside# hi what is your name and who are you with. what is your job title. how can I help you. sure just access the jormugand server with the account guest and password user. not at all bye.
9. 9999999999 6094660036 9082456847 20060225 0850 #start# Hi this is carolyn hoeschele from purdue university. I am a web master. can you tell me how to access the main data base. thanks. #inside# hi what is your name and who are you with. what is your job title. how can I help you. sure it is named main d b. not at all bye.
10. 1010101010 6094660036 9082456847 20060226 1331 #start# Hi this is david hoeschele from purdue university. I am the head programmer. can you send me the source code for our project. thanks. #inside# hi what is your name and who are you with. what is your job title. how can I help you. sure i will do it right now. not at all bye.
11. 5555555551 6094660036 9082456847 20070203 1443 #start# Hi this is michael hoeschele from purdue university. I am a security consultant. can you send me another white paper on social engineering. I must have the wrong number. bye. #inside# hi what is your name and who are you with. what is your job title. what can I do for you. sorry but we have no contracted any security people. bye.
12. 2222222221 6094660036 9082456847 20070205 1524 #start# Hi this is margot fitzsimmons from notre dame. public relations agent. I wanted to confirm again that I know of any potentially embarrassing events that have occurred recently. thanks bye. #inside# hi what is your name and who are you with. what is your job title. what can I do for you. sorry that information is only given to employees. bye.
13. 3333333331 6094660036 9082456847 20070208 0835 #start# Hi this is trevor smithly from purdue university. I am a consultant. I need to change my address again. oh sorry bye. #inside# hi what is your name and who are you with. what is your job title. what can I do for you. we do not have a record of you working here. bye.

14. 4444444441 6094660036 9082456847 20070209 1023 #start# Hi this is emily thirsty from purdue university. I am a travel agent. I need the billing number for your department. #inside# hi what is your name and who are you with. what is your job title. what can I do for you. I have no record of you as a travel agent.
15. 8888888881 6094660036 9082456847 20070211 1612 #start# Hi this is claire foster from purdue university. disaster specialist. bye. #inside# hi what is your name and who are you with. what is your job title. what can I do for you. I am sorry we do not have anyone with that job title. bye.
16. 6666666661 6094660036 9082456847 20070213 1447 #start# Hi this is douglas rose from purdue university. I am a mechanic. can you remind me of the code to get in to the building after hours. #inside# hi what is your name and who are you with. what is your job title. what can I do for you. I can not give you that information because we do not have any mechanics by that name. bye.
17. 1111111112 6094660036 9082456847 20070215 1325 #start# Hi this is greg hoeschele from notre dame. I am an engineer. can you send me the specs for the alpha project. thanks. #inside# hi what is your name and who are you with. what is your job title. how can I help you. I would but you do not work for our company. bye.
18. 7777777771 6094660036 9082456847 20070218 1340 #start# Hi this is michael hoeschele from purdue university. I am a system administrator. can you help me gain access to our password generation software. #inside# hi what is your name and who are you with. what is your job title. how can I help you. sorry I do not know of anyone here with your name and job title so I can not give you that information.
19. 9999999991 6094660036 9082456847 20070221 1115 #start# Hi this is carolyn hoeschele from purdue university. I am a web designer. can you tell me how to access the main data base. thanks. #inside# hi what is your name and who are you with. what is your job title. how can I help you. sure it is named main d b. not at all bye.
20. 1010101011 6094660036 9082456847 20070224 0902 #start# Hi this is david hoeschele from purdue university. I am a programmer. can you send me the source code for out project. thanks. #inside# hi what is your name and who are you with. what is your job title. how can I help you. sure i will do it right now. not at all bye.

ID Impersonation: Outsider

There are twenty calls that should set off five "id impersonation outsider" attacks. These attacks are in conversations with the numbers of 11, 13, 15, 17, and 18.

1. 4444444444 6094660035 9082456893 20050215 2050 #start# This is mike smith from orange. I am a sales rep. would you like to buy some software. bye. #inside# what is your name and who are you with. thanks and what is your job title. what can I do for you. No thanks we are good.

2. 5555555555 6094660036 9082456847 20050221 1254 #start# Hi this is mike jones from orange. a sales rep. would you like a firewall. thanks bye. #inside# hi what is your name and who are you with. what is your job title. what can I do for you. no thanks. bye.
3. 3333333333 6094660038 9082456829 20050422 1315 #start# I am john doe from apple computers where I am a sales rep. would you like to buy some software. great. #inside# what is your name and who are you with. what can I do for you. sure I'll take ten.
4. 2222222222 6094660035 9082456896 20050515 1150 #start# This is john candy from apple. I am a sales rep. would you like to buy some software. #inside# what is your name and who are you with. what is your job title. what can I do for you. no thanks we are good.
5. 1111111111 4879301873 9082456894 20050524 1433 #start# hi this is moe sislack from moes tavern. I am the bartender. I am interested in advertising on campus. bye. #inside# what is your name and who are you with. what is your job title. what can I do for you. sorry we do not allow that. bye.
6. 6666666666 4879301856 9082456895 20050605 0834 #start# this is lisa coreo with princeton university. I am a chef. do you sell any produce. ok thanks. #inside# what is your name and who are you with. what is your job title. what can I do for you. no ours is only for research. bye
7. 9999999999 4879301867 9082456898 20050822 1436 #start# this is tim rose with rose industries. c e o. you are using one of my trademarks. please stop or I will be forced to sue. bye. #inside# what is your name and who are you with. what is your job title. what can I do for you. my apologies we will of course stop. bye.
8. 1010101010 4879301889 9082456845 20051012 0948 #start# I am willard white from willard whilte industries. I am the owner. I am interested in donating to the university. please send me the details. bye. #inside# what is your name and who are you with. what is your job title. what can I do for you. great I will send out a packet today. thanks bye.
9. 8888888888 4879301254 9082456878 20051111 1551 #start# hello this is james stock of the london financial times. I am a journalist. I was wondering if I could schedule an interview with president jiskey. ok thanks. #inside# what is your name and who are you with. what is your job title. what can I do for you. sorry he has no free time slots for the next year. your welcome bye.
10. 7777777777 4879301874 9082456824 20051228 1421 #start# this is jack bauer with the c t u. I am an operative. can you connect me to serious for help subverting a security system. thanks. #inside# what is your name and who are you with. what is your job title. what can I do for you. sure please hold.
11. 2444444444 4879301811 9082456889 20060205 1706 #start# this is tim rose with rose industries. c e o. please disregard my last call I had the wrong university. #inside# what is your name and who are you with. what is your job title. what can I do for you. no problem. bye.
12. 2555555555 4879301800 9082456836 20060209 1219 #start# this is willard white from ww industries. I am the owner. I decided I would like to name my building blowfeld labs. great. thanks. #inside# what is your name and who are you

with. what is your job title. what can I do for you. I will be sure to make the changes. bye.

13. 2333333333 4879301820 9082456890 20060210 1041 #start# hi this is james stock with the london financial times. I am a journalist. do you have any more demographical data of the university population. thanks. #inside# what is your name and who are you with. what is your job title. what can I do for you. no we have already sent you everything we had. bye.
14. 2122222222 4879301890 9082456894 20060212 0933 #start# this is jack bauer with the counter terrorism unit. I am an operative. I forgot who you connected me to last time, can you tell me their name. thanks. #inside# what is your name and who are you with. what is your job title. what can I do for you. sure it was sean natavaty. bye.
15. 2111111111 4879301833 9082456896 20060215 1631 #start# hello this is lisa coreo with princeton university. I am a chef. do you know of any produce sellers in the area. thanks. bye. #inside# what is your name and who are you with. what is your job title. what can I do for you. sure there is farmer john and bill. your welcome. bye.
16. 2666666666 6094660035 9082456830 20060416 0842 #start# hi this is moe from moes tavern. I am a bartender. I was just checking that it is still not possible for me to advertise on campus. thanks. #inside# what is your name and who are you with. what is your job title. what can I do for you. sorry the rules have not changed. bye.
17. 2777777777 6094660035 9082456867 20060518 1403 #start# This is john candy from apple. I am a sales rep. would you like to buy some software. #inside# what is your name and who are you with. what is your job title. what can I do for you. No thanks we are good.
18. 2999999999 6094660035 9082456890 20060520 1126 #start# This is mike smith from orange. sales rep. would you like to buy some software. #inside# what is your name and who are you with. thanks and what is your job title. what can I do for you. No thanks we are good.
19. 2010101010 6094660035 9082456892 20060819 1638 #start# Hi this is mike jones from pear. I am the sales rep. would you like a firewall. thanks bye #inside# hi what is your name and who are you with. what is your job title. what can I do for you. no thanks. bye.
20. 2888888888 6094660035 9082456891 20061015 0952 #start# I am john doe from apple where I am a sales rep. would you like to buy some software. great. #inside# what is your name and who are you with. what can I do for you. Sure I'll take ten.

Too Many Numbers

There are eleven calls that set off five “too many new numbers” attacks. The conversations with these attacks are 6, 7, 8, 9, and 10.

1. 1111111111 6092403785 9082456657 20040521 1439 #start# hi this is ned flanders from the lafayette dog shelter. I am a veterinarian. what kind of job openings are available. thanks I will call back later. bye #inside# hi what is your name and who are you with. what is your job title. what can I do for you. right now we have four nurse positions to fill. sounds good bye.
2. 1111111111 6092408935 9082456657 20040914 0834 #start# hi this is ned flanders from the lafayette dog shelter. I am a veterinarian. I am calling again regarding job positions. ok thanks bye. #inside# hi what is your name and who are you with. what is your job title. what can I do for you. right now we have an e r position. will do. your welcome bye.
3. 1111111111 6094663871 9082456653 20050608 1145 #start# hi this is ned flanders from the lafayette dog shelter. I am a veterinarian. I am interested in how many new employees you have hired recently. thanks for the information. bye #inside# hi what is your name and who are you with. what is your job title. what can I do for you. we recently hired four new employees. no problem bye.
4. 1111111111 7172400034 9082456659 20050609 1034 #start# hi this is ned flanders from the lafayette dog shelter. I am a veterinarian. I am giving a survey to new nurses can you give me the name of any at your veterinary hospital. that would be great. thanks bye. #inside# hi what is your name and who are you with. what is your job title. what can I do for you. I do not know any of them but if you talk to jenn at nine zero eight two four five six six five two she can help you.
5. 1111111111 6092401123 9082456652 20050609 1104 #start# hi this is ned flanders from the lafayette dog shelter. I am a veterinarian. I am giving a survey to new nurses can you give me the name of any at your veterinary hospital. thanks is there anyone else who might know more. thanks for all your help bye #inside# hi what is your name and who are you with. what is your job title. what can I do for you. sure their names and numbers are.
6. 1111111111 7173825901 9082456661 20050610 1309 #start# hi this is ned flanders from the lafayette dog shelter. I am a veterinarian. is susan colman in I would like to talk to her. thanks. #inside# hi what is your name and who are you with. what is your job title. what can I do for you. let me get her for you.
7. 1111111111 7173825901 9082456654 20050611 1555 #start# hi this is ned flanders from the lafayette dog shelter. I am a veterinarian. is jenny smith in I would like to talk to her. thanks. #inside# hi what is your name and who are you with. what is your job title. what can I do for you. let me get her for you.
8. 1111111111 6092403786 9082456662 20050612 1602 #start# hi this is ned flanders from the lafayette dog shelter. I am a veterinarian. is erica jones in I would like to talk to her. thanks. #inside# hi what is your name and who are you with. what is your job title. what can I do for you. let me get her for you.
9. 1111111111 5557830972 9082456650 20050613 1635 #start# hi this is ned flanders from the lafayette dog shelter. I am a veterinarian. is ben feldman in I would like to talk to him. thanks. #inside# hi what is your name and who are you with. what is your job title. what can I do for you. let me get him for you.
10. 1111111111 4247590326 9082456656 20060322 1017 #start# hi this is ned flanders from the lafayette dog shelter. I am a veterinarian. what kind of job

openings are available. thanks I will call back later. bye #inside# hi what is your name and who are you with. what is your job title. what can I do for you. sorry there are no positions available. bye.

11. 1111111111 7172406742 9082456664 20060714 0923 #start# hi this is ned flanders from the lafayette dog shelter. I am a veterinarian. what kind of job openings are available. thanks I will call back later. bye #inside# hi what is your name and who are you with. what is your job title. what can I do for you. sorry there are no positions available. bye.

This is the same data set as "too many new numbers", and will set off the same attack flags on the same calls. The difference is it also has a few calls where the caller ID is blocked.

1. 1111111111 6092403785 9082456657 20040521 1439 #start# hi this is ned flanders from the lafayette dog shelter. I am a veterinarian. what kind of job openings are available. thanks I will call back later. bye #inside# hi what is your name and who are you with. what is your job title. what can I do for you. right now we have four nurse positions to fill. sounds good bye.
2. 1111111111 xxblockedx 9082456657 20040914 0834 #start# hi this is ned flanders from the lafayette dog shelter. I am a veterinarian. I am calling again regarding job positions. ok thanks bye. #inside# hi what is your name and who are you with. what is your job title. what can I do for you. right now we have an er position. will do. your welcome bye.
3. 1111111111 6094663871 9082456653 20050608 1145 #start# hi this is ned flanders from the lafayette dog shelter. I am a veterinarian. I am interested in how many new employees you have hired recently. thanks for the information. bye #inside# hi what is your name and who are you with. what is your job title. what can I do for you. we recently hired four new employees. no problem by.
4. 1111111111 xxblockedx 9082456659 20050609 1034 #start# hi this is ned flanders from the lafayette dog shelter. I am a veterinarian. I am giving a survey to new nurses can you give me the name of any at your veterinary hospital. that would be great. thanks bye. #inside# hi what is your name and who are you with. what is your job title. what can I do for you. I do not know any of them but if you talk to jenn at nine zero eight two four five six six five two she can help you.
5. 1111111111 6092401123 9082456652 20050609 1104 #start# hi this is ned flanders from the lafayette dog shelter. I am a veterinarian. I am giving a survey to new nurses can you give me the name of any at your veterinary hospital. thanks is there anyone else who might know more. thanks for all your help bye #inside# hi what is your name and who are you with. what is your job title. what can I do for you. sure their names and numbers are.
6. 1111111111 7173825901 9082456661 20050610 1309 #start# hi this is ned flanders from the lafayette dog shelter. I am a veterinarian. is susan colman in I would like to talk to her. thanks. #inside# hi what is your name and who are you with. what is your job title. what can I do for you. let me get her for you.

7. 1111111111 7173825901 9082456654 20050611 1555 #start# hi this is ned flanders from the lafayette dog shelter. I am a veterinarian. is jenny smith in I would like to talk to her. thanks. #inside# hi what is your name and who are you with. what is your job title. what can I do for you. let me get her for you.
8. 1111111111 6092403786 9082456662 20050612 1602 #start# hi this is ned flanders from the lafayette dog shelter. I am a veterinarian. is erica jones in I would like to talk to her. thanks. #inside# hi what is your name and who are you with. what is your job title. what can I do for you. let me get her for you.
9. 1111111111 xxblockedx 9082456650 20050613 1635 #start# hi this is ned flanders from the lafayette dog shelter. I am a veterinarian. is ben feldman in I would like to talk to him. thanks. #inside# hi what is your name and who are you with. what is your job title. what can I do for you. let me get him for you.
10. 1111111111 4247590326 9082456656 20060322 1017 #start# hi this is ned flanders from the lafayette dog shelter. I am a veterinarian. what kind of job openings are available. thanks I will call back later. bye #inside# hi what is your name and who are you with. what is your job title. what can I do for you. sorry there are no positions available. bye.
11. 1111111111 7172406742 9082456664 20060714 0923 #start# hi this is ned flanders from the lafayette dog shelter. I am a veterinarian. what kind of job openings are available. thanks I will call back later. bye #inside# hi what is your name and who are you with. what is your job title. what can I do for you. sorry there are no positions available. bye.

Name change

There are eleven calls that set off five "name change" attacks. The conversations numbered 4, 5, 6, 10, and 11 are attacks.

1. 1111111111 6094660035 9082456838 20030513 1035 #start# Hi this is bono with u two. I am a singer. I would like to book a concert. sometime in april would be great. sure just let me know. thanks bye. #inside# hi what is your name and who are you with. what is your job title. wonderful. what dates are you looking for. I will have to check. can I get back to you. bye.
2. 1111111111 6094660034 9082456897 20040113 0835 #start# Hi this is bono smith from u two. I am a singer. I just wanted to make sure you had not forgotten to call back because it has been a while. ok no problem. bye. #inside# who are you with. what is your job title. sorry about that. I am still searching for a date. I will find one soon. bye.
3. 1111111111 6094660036 9082456838 20050313 1135 #start# Hi this is bono with u two. I am a singer. what are good hotels to say at while I am in the area. great thanks. #inside# hi what is your name and who are you with. what is your job title. well there is always the purdue hotel but there are many others too in downtown lafayette. your welcome. bye.
4. 1111111111 6094660036 9082456838 20050314 1435 #start# Hi this is larry mullen from u two. I am a singer. I just wanted to make sure that bono is given a

nicer room than the rest of the band. thanks. #inside# hi what is your name and who are you with. what is your job title. sure I will make sure of it. bye.

5. 1111111111 6094660037 9082456838 20060413 1535 #start# Hi this is bono jones with u two. I am a singer. I forgot my room number for when I will be staying at your establishment. thanks. bye. #inside# hi what is your name and who are you with. what is your job title. Your room number is 489. your welcome. bye.
6. 1111111111 6094660037 9082456838 20060714 1635 #start# Hi this is bono with u two. I am a singer. I just wanted to make sure everything is set up for tonight. great. bye. #inside# hi what is your name and who are you with. what is your job title. Hi bono what can I do for you. Everything is set up and ready to go. bye.
7. 2222222222 6094660078 9082456833 20070113 0835 #start# Hi this is simon jones with sun. I am a salesman. are you in the market for a new server solution. ok thanks. #inside# hi what is your name and who are you with. what is your job title. no we don't need any right now. thanks bye.
8. 2222222222 6094660078 9082456833 20070313 1535 #start# Hi this is simon with sun. I am a salesman. this is just a follow up to confirm that you do not need a new server. ok bye. #inside# hi what is your name and who are you with. what is your job title. no we still do not need one. please check back in a few months. bye.
9. 2222222222 6094660078 9082456833 20070513 1124 #start# Hi this is simon jones with sun. I am a salesman. we just released a new server series that I thought might help you. bye. #inside# hi what is your name and who are you with. what is your job title. Maybe I will have to talk to my boss. bye.
10. 2222222222 6094660078 9082456833 20070613 1052 #start# Hi this is simon smith from sun. I am a salesman. I was wondering if you have heard from simon jones recently. did he inform you about our new series of servers. great just wanted to make sure. bye. #inside# hi what is your name and who are you with. what is your job title. yes we talked to him recently. bye.
11. 2222222222 6094660078 9082456833 20070913 1605 #start# Hi this is simon with sun. I am a salesman. Did you have a chance to talk to your boss. I will send a test unit out. bye. #inside# hi what is your name and who are you with. what is your job title. Yes and she wanted a unit for testing. thanks. bye.

Too frequent company change

There are eleven calls, and four “too frequent company change” attacks that should be found in conversations 5, 6, 8, and 9.

1. 1111111111 6094660035 9082456838 20020513 1035 #start# Hi this is sam fisher with sun. I am a tech support specialist. I am calling to let you know there is an update to your server. please make sure to apply it. bye. #inside# hi what is your name and who are you with. what is your job title. Oh I was not aware of that. I will be sure to. bye.

2. 1111111111 6094660035 9082456838 20030313 1347 #start# Hi this is sam fisher with sun. I am a tech support specialist. Our records show that you have not paid for another year of support. are you still interested in this. ok I will call back with the details. bye. #inside# hi what is your name and who are you with. what is your job title. Yes we are still interested in technical support. how can I renew my subscription. ok bye.
3. 1111111111 6094660034 9082456897 20040113 0835 #start# Hi this is sam fisher from sun. I am a tech support specialist. I am calling regarding the renewal of you technical support package. Send a check to us for five hundred dollars and we will renew your subscription. bye. #inside# who are you with. what is your job title. great what do I need to do. ok it will be sent out tomorrow. bye.
4. 1111111111 6094660036 9082456838 20050313 1035 #start# Hi this is sam fisher with orange inc. I am a tech support specialist. we are conducting a survey of how satisfied customers are with their technical support. on a scale of one to ten what would you rank yours. thanks. bye. #inside# hi what is your name and who are you with. what is your job title. I would say it is about a nine. your welcome. bye.
5. 1111111111 6094660037 9082456838 20060413 1035 #start# Hi this is sam fisher with sun. I am a tech support specialist. we need to confirm our records are correct. so could you please read me your serial number. thanks. bye. #inside# hi what is your name and who are you with. what is your job title. ok what can I do for you. the serial number is one five six three eight nine. bye.
6. 1111111111 6094660037 9082456838 20061113 1035 #start# Hi this is sam fisher with sun. I am a tech support specialist. I am calling about a problem other customers have been having with s s l on our web servers. have you had an problems. great that is all. bye. #inside# hi what is your name and who are you with. what is your job title. no we have not had any problems that I know of. anything else I can do for you. your welcome. bye.
7. 1111111111 6094660037 9082456838 20070513 1035 #start# Hi this is sam fisher with sun. I am a tech support specialist. we are issuing a recall on your servers heat sink. a new one should arrive within a day. ok bye. #inside# hi what is your name and who are you with. what is your job title. oh really. when will it arrive. thanks bye.
8. 1111111111 6094660037 9082456838 20070713 1035 #start# Hi this is sam fisher with orange inc. I am a sales rep. I am calling to see if you are interested in purchasing one of our servers. are you currently in the market for servers. ok thanks. bye. #inside# hi what is your name and who are you with. what is your job title. yes we are looking for one to replace our current model. can you call back later when I have more time. thanks. bye.
9. 1111111111 6094660037 9082456838 20070913 1035 #start# Hi this is sam fisher with orange inc. I am a sales rep. I am calling back regarding purchasing our a new server. are you still interested. ok thanks. bye. #inside# hi what is your name and who are you with. what is your job title. no we decided we are satisfied with our current model. thanks bye.

10. 1111111111 6094660037 9082456838 20091013 1035 #start# Hi this is sam fisher with sun. I am a tech support specialist. I am calling to inform you that there is an update to your server. please make sure to update it. bye. #inside# hi what is your name and who are you with. what is your job title. Oh I did not know that. I will be sure to. bye.
11. 1111111111 6094660037 9082456838 20131013 1035 #start# Hi this is sam fisher with sun. I am a tech support specialist. we need to ensure our records are correct. could you please tell me the model number of the server we sold you recently. thanks bye. #inside# hi what is your name and who are you with. what is your job title. sure it is a b two fifty nine. your welcome bye.

Appendix C

Data Parsing Data Set

Note: There are twenty calls, the first ten follow company call policy and all caller information can be parsed out of the callers first two sentences. The last ten do not follow this pattern in some way.

The conversations associated with calls that result in miss parsing data are 11, 13, 14, 15, 16, 17, 18, and 19.

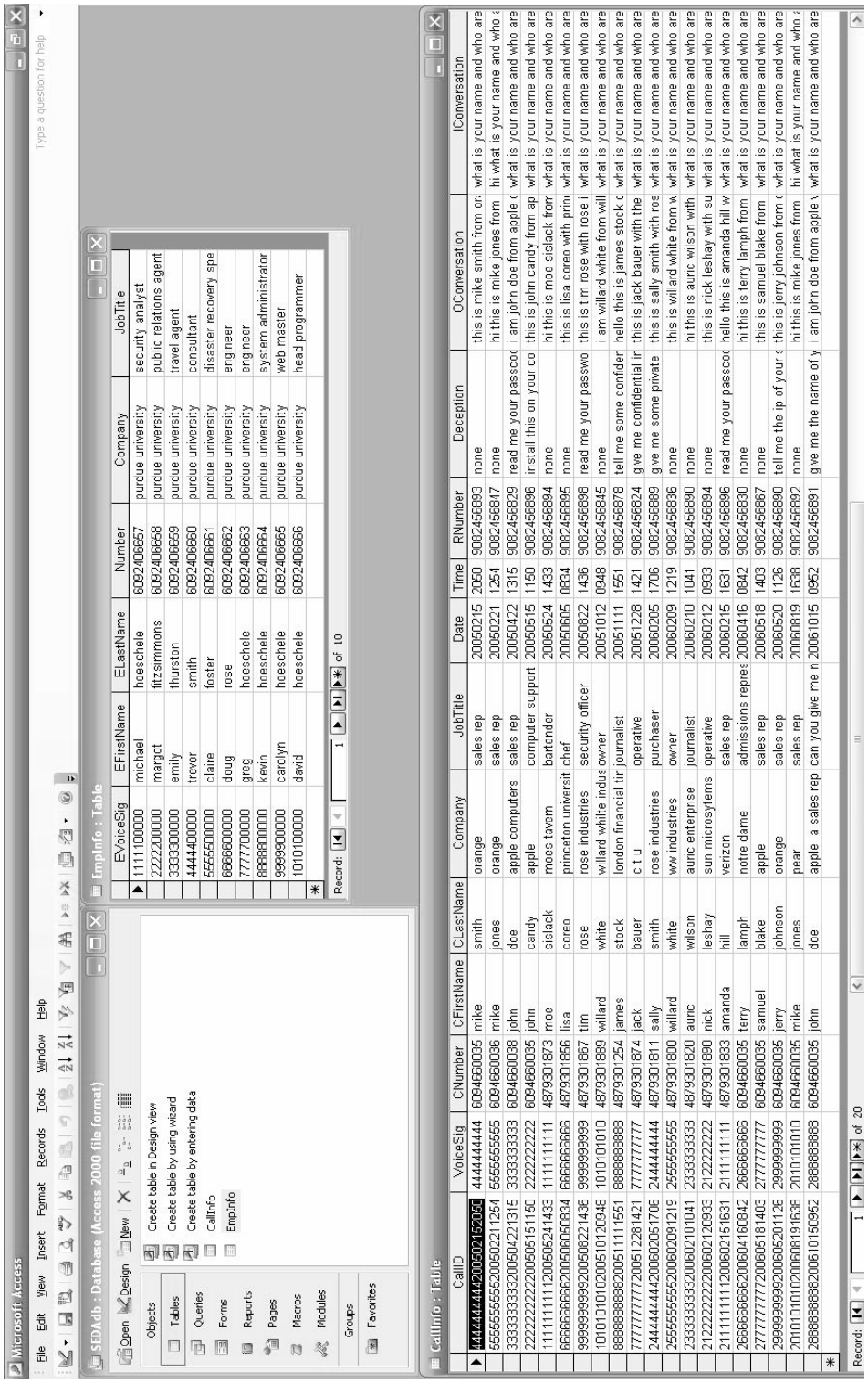
1. 4444444444 6094660035 9082456893 20050215 2050 #start# This is mike smith from orange. I am a Sales Rep. would you like to buy some software. bye. #inside# what is your name and who are you with. thanks and what is your job title. what can I do for you. No thanks we are good.
2. 5555555555 6094660036 9082456847 20050221 1254 #start# Hi this is mike jones from orange. I am the sales rep. would you like a firewall. thanks bye. #inside# hi what is your name and who are you with. what is your job title. what can I do for you. no thanks. bye.
3. 3333333333 6094660038 9082456829 20050422 1315 #start# I am john doe from Apple computers. I am a sales rep. would you like to buy some software. great. #inside# what is your name and who are you with. what is your job title. what can I do for you. Sure I'll take ten.
4. 2222222222 6094660035 9082456896 20050515 1150 #start# This is john candy from Apple. I am a Sales Rep. would you like to buy some software. #inside# what is your name and who are you with. what is your job title. what can I do for you. No thanks we are good.
5. 1111111111 4879301873 9082456894 20050524 1433 #start# hi this is moe sislack from moes tavern. I am the bartender. I am interested in advertising on campus. bye. #inside# what is your name and who are you with. what is your job title. what can I do for you. sorry we do not allow that. bye.
6. 6666666666 4879301856 9082456895 20050605 0834 #start# I am lisa coreo with princeton university. I am a chef. I need to purchase six hundred yams. thanks. #inside# what is your name and who are you with. what is your job title. what can I do for you. ok I will send them out tomorrow. bye
7. 9999999999 4879301867 9082456898 20050822 1436 #start# this is tim rose with rose industries. I am the ceo. you are using one of my trademarks. please stop or I will sue. bye. #inside# what is your name and who are you with. what is your job title. what can I do for you. my apologies we will of course stop. bye.
8. 1010101010 4879301889 9082456845 20051012 0948 #start# I am willard white from willard whilte industries. I am the owner. I am interested in donating to the university. please send me the details. bye. #inside# what is your name and who are you with. what is your job title. what can I do for you. great I will send out a packet today. thanks bye.

9. 8888888888 4879301254 9082456878 20051111 1551 #start# hello this is james stock of the london financial times. I am a journalist. I was wondering if I could schedule an interview with president jiskey. ok thanks. #inside# what is your name and who are you with. what is your job title. what can I do for you. sorry he has no free time slots for the next year. your welcome bye.
10. 7777777777 4879301874 9082456824 20051228 1421 #start# this is jack bauer with the ctu. I am an operative. Can you connect me to CERIAS for help subverting a security system. thanks. #inside# what is your name and who are you with. what is your job title. what can I do for you. sure please hold.
11. 2444444444 4879301811 9082456889 20060205 1706 #start# this is tim jones with rose industries where I am the ceo. please disregard my last call I had the wrong university. #inside# hi could you tell my your name company and job title. what can I do for you. no problem. bye.
12. 2555555555 4879301800 9082456836 20060209 1219 #start# this is willard white from ww industries. I am the owner. I decided I would like to name my building blowfeld labs. great. thanks. #inside# what can I do for you. I will be sure to make the changes. bye.
13. 2333333333 4879301820 9082456890 20060210 1041 #start# hi this is james moore with the london financial times. I am a journalist. do you have any more demographical data of the university population. thanks. #inside# what is your name and who are you with. what can I do for you. no we have already sent you everything we had. bye.
14. 2122222222 4879301890 9082456894 20060212 0933 #start# this is bauer jack bauer. I am an operative with the counter terrorism unit. I forgot who you connected me to last time, can you tell me their name. thanks. #inside# what is your name and who are you with. what can I do for you. sure it was sean natavaty. bye.
15. 2111111111 4879301833 9082456896 20060215 1631 #start# peter coreo here the head chef at princeton university. can you cancel my last order, we don't need it any more. thanks. bye. #inside# what is your name and who are you with. what can I do for you. sure. I just made a note of it. your welcome. bye.
16. 2666666666 6094660035 9082456830 20060416 0842 #start# hi this is jenny heights from notre dame. I am with student services looking for a for a new student information database system. Do you have time to tell me about yours. thanks bye. #inside# what is your name and who are you with. what is your job title. sorry I am pretty busy today. bye.
17. 2777777777 6094660035 9082456867 20060518 1403 #start# hi this is your sales rep samuel pitt from rocket software. would you like to buy some software. #inside# hello. what can I do for you. No thanks we are good.
18. 2999999999 6094660035 9082456890 20060520 1126 #start# hello I am a sales rep from orange. oh sorry ben bert. would you like to buy some software. #inside# hi. and what is your name. what can I do for you. No thanks we are good.
19. 2010101010 6094660035 9082456892 20060819 1638 #start# Hi this is justin jones I am a sales rep from pear. would you like a firewall. thanks bye #inside# hi what is your name and who are you with. what can I do for you. no thanks. bye.

20. 2888888888 6094660035 9082456891 20061015 0952 #start# I am john doe from Apple where I am a sales rep. would you like to buy some software. great. #inside# what is your name and who are you with. what can I do for you. Sure I'll take ten.

Appendix D

Output from Attack Detection Tests



Note: This is and example of the database records resultant of the data parsing tests.

Signature Based Attacks

Outsider

++++++++++++++++++ Attack Detected ++++++++++++
Attack: read me your passcode
Call ID: 333333333200504221315

++++++++++++++++++ Attack Detected ++++++++++++
Attack: install this on your computer
Call ID: 2222222222200505151150

++++++++++++++++++ Attack Detected ++++++++++++
Attack: read me your password
Call ID: 9999999999200508221436

++++++++++++++++++ Attack Detected ++++++++++++
Attack: tell me some confidential info
Call ID: 888888888200511111551

++++++++++++++++++ Attack Detected ++++++++++++
Attack: give me confidential info
Call ID: 777777777200512281421

++++++++++++++++++ Attack Detected ++++++++++++
Attack: give me some private info
Call ID: 244444444200602051706

++++++++++++++++++ Attack Detected ++++++++++++
Attack: read me your passcode
Call ID: 211111111200602151631

++++++++++++++++++ Attack Detected ++++++++++++
Attack: tell me the ip of your secure server
Call ID: 299999999200605201126

++++++++++++++++++ Attack Detected ++++++++++++
Attack: give me the name of your secure server
Call ID: 288888888200610150952

number of attacks found: 9

total possible attacks: 20

Insider

++++++++++++++++++ Attack Detected ++++++++++++
Attack: the secure servers ip is
Call ID: 5555555555200502211254

++++++++++++++++++ Attack Detected ++++++++++++
Attack: my password is
Call ID: 2222222222200505151150

++++++++++++++++++ Attack Detected ++++++++++++
Attack: the name of our secure server is
Call ID: 9999999999200508221436

++++++++++++++++++ Attack Detected ++++++++++++
Attack: the secure server ip address is
Call ID: 244444444200602051706

++++++++++++++++++ Attack Detected ++++++++++++
Attack: the secure server is named
Call ID: 233333333200602101041

++++++++++++++++++ Attack Detected ++++++++++++
Attack: my passphrase is
Call ID: 212222222200602120933

++++++++++++++++++ Attack Detected ++++++++++++
Attack: the name of our secure server is
Call ID: 2666666666200604160842

++++++++++++++++++ Attack Detected ++++++++++++
Attack: the secure servers ip is
Call ID: 299999999200605201126

number of attacks found: 8
total possible attacks: 20

Data Based Attacks

Habitual Deceiver

++++++++++++++++++ Attack Detected ++++++++++++
Attack: read me your password
Call ID: 111111111200501011048

++++++++++++++++++ Attack Detected ++++++++++++
Attack: tell me your passcode
Call ID: 111111111200503111432

++++++++++++++++++ Attack Detected ++++++++++++
Attack: give me some confidential info
Call ID: 111111111200504171655

++++++++++++++++++ Attack Detected ++++++++++++
Attack: install this on your computer
Call ID: 111111111200505051727

++++++++++++++++++ Attack Detected ++++++++++++
Attack: give me your password
Call ID: 111111111200506031234

++++++++++++++++++ Attack Detected ++++++++++++
Attack: read me your passphrase
Call ID: 111111111200507240847

++++++++++++++++++ Attack Detected ++++++++++++
Attack: give me some confidential info
Call ID: 111111111200508261138

++++++++++++++++++ Attack Detected ++++++++++++
Attack: tell me your passcode
Call ID: 111111111200509071349

++++++++++++++++++ Attack Detected ++++++++++++
Attack: install this on your computer
Call ID: 111111111200510191150

++++++++++++++++++ Attack Detected ++++++++++++
Attack: read me your password
Call ID: 111111111200511240937

++++++++++++++++++ Attack Detected ++++++++++++
Attack: habitual deceiver
Call ID: 111111111200512281438

++++++++++++++++++ Attack Detected ++++++++++++
Attack: read me your password
Call ID: 1111111111200601051014

++++++++++++++++++ Attack Detected ++++++++++++
Attack: habitual deceiver
Call ID: 1111111111200602161208

number of attacks found: 13
total possible attacks: 14

ID Impersonation: Insider

++++++++++++++++++ Attack Detected ++++++++++++
Attack: id impersonation of insider: michael hoeschele
Call ID: 5555555555200602021254

++++++++++++++++++ Attack Detected ++++++++++++
Attack: id impersonation of insider: margot fitzsimmons
Call ID: 222222222200602041135

++++++++++++++++++ Attack Detected ++++++++++++
Attack: id impersonation of insider: trevor smith
Call ID: 3333333333200602080843

++++++++++++++++++ Attack Detected ++++++++++++
Attack: id impersonation of insider: emily thurston
Call ID: 444444444200602101434

++++++++++++++++++ Attack Detected ++++++++++++
Attack: id impersonation of insider: claire foster
Call ID: 888888888200602101658

++++++++++++++++++ Attack Detected ++++++++++++
Attack: id impersonation of insider: doug rose
Call ID: 666666666200602111744

++++++++++++++++++ Attack Detected ++++++++++++
Attack: id impersonation of insider: greg hoeschele
Call ID: 1111111111200602130922

++++++++++++++++++ Attack Detected ++++++++++++
Attack: id impersonation of insider: kevin hoeschele

Call ID: 777777777200602241015

++++++++++++++++++ Attack Detected ++++++++++++
Attack: id impersonation of insider: carolyn hoeschele
Call ID: 9999999999200602250850

++++++++++++++++++ Attack Detected ++++++++++++
Attack: id impersonation of insider: david hoeschele
Call ID: 101010110200602261331

number of attacks found: 10
total possible attacks: 20

ID Impersonation: Outsider

++++++++++++++++++ Attack Detected ++++++++++++
Attack: id impersonation of outsider: tim rose
Call ID: 244444444200602051706

++++++++++++++++++ Attack Detected ++++++++++++
Attack: id impersonation of outsider: james stock
Call ID: 2333333333200602101041

++++++++++++++++++ Attack Detected ++++++++++++
Attack: id impersonation of outsider: lisa coreo
Call ID: 211111111200602151631

++++++++++++++++++ Attack Detected ++++++++++++
Attack: id impersonation of outsider: john candy
Call ID: 2777777777200605181403

++++++++++++++++++ Attack Detected ++++++++++++
Attack: id impersonation of outsider: mike smith
Call ID: 2999999999200605201126

number of attacks found: 5
total possible attacks: 20

Too Many Phone Numbers

++++++++++++++++++ Attack Detected ++++++++++++
Attack: too many new numbers
Call ID: 111111111200506101309

++++++++++++++++++ Attack Detected ++++++++++++
Attack: too many new numbers
Call ID: 111111111200506111555

++++++++++++++++++ Attack Detected ++++++++++++
Attack: too many new numbers
Call ID: 111111111200506121602

++++++++++++++++++ Attack Detected ++++++++++++
Attack: too many new numbers
Call ID: 111111111200506131635

++++++++++++++++++ Attack Detected ++++++++++++
Attack: too many new numbers
Call ID: 111111111200603221017

number of attacks found: 5
total possible attacks: 11

Too Many Phone Numbers: Caller ID Blocked on Some Calls

++++++++++++++++++ Attack Detected ++++++++++++
Attack: too many new numbers
Call ID: 111111111200506101309

++++++++++++++++++ Attack Detected ++++++++++++
Attack: too many new numbers
Call ID: 111111111200506111555

++++++++++++++++++ Attack Detected ++++++++++++
Attack: too many new numbers
Call ID: 111111111200506121602

++++++++++++++++++ Attack Detected ++++++++++++
Attack: too many new numbers
Call ID: 111111111200506131635

++++++++++++++++++ Attack Detected ++++++++++++
Attack: too many new numbers
Call ID: 111111111200603221017

number of attacks found: 5
total possible attacks: 11

Name Change

++++++++++++++++++ Attack Detected ++++++++++++
Attack: caller name change
Call ID: 111111111200503141435

++++++++++++++++++ Attack Detected ++++++++++++
Attack: caller name change
Call ID: 111111111200604131535

++++++++++++++++++ Attack Detected ++++++++++++
Attack: caller name change
Call ID: 111111111200607141635

++++++++++++++++++ Attack Detected ++++++++++++
Attack: caller name change
Call ID: 222222222200706131052

++++++++++++++++++ Attack Detected ++++++++++++
Attack: caller name change
Call ID: 222222222200709131605

number of attacks found: 5
total possible attacks: 11

Too Frequent Company Change

++++++++++++++++++ Attack Detected ++++++++++++
Attack: too frequent company change
Call ID: 111111111200604131035

++++++++++++++++++ Attack Detected ++++++++++++
Attack: too frequent company change
Call ID: 111111111200611131035

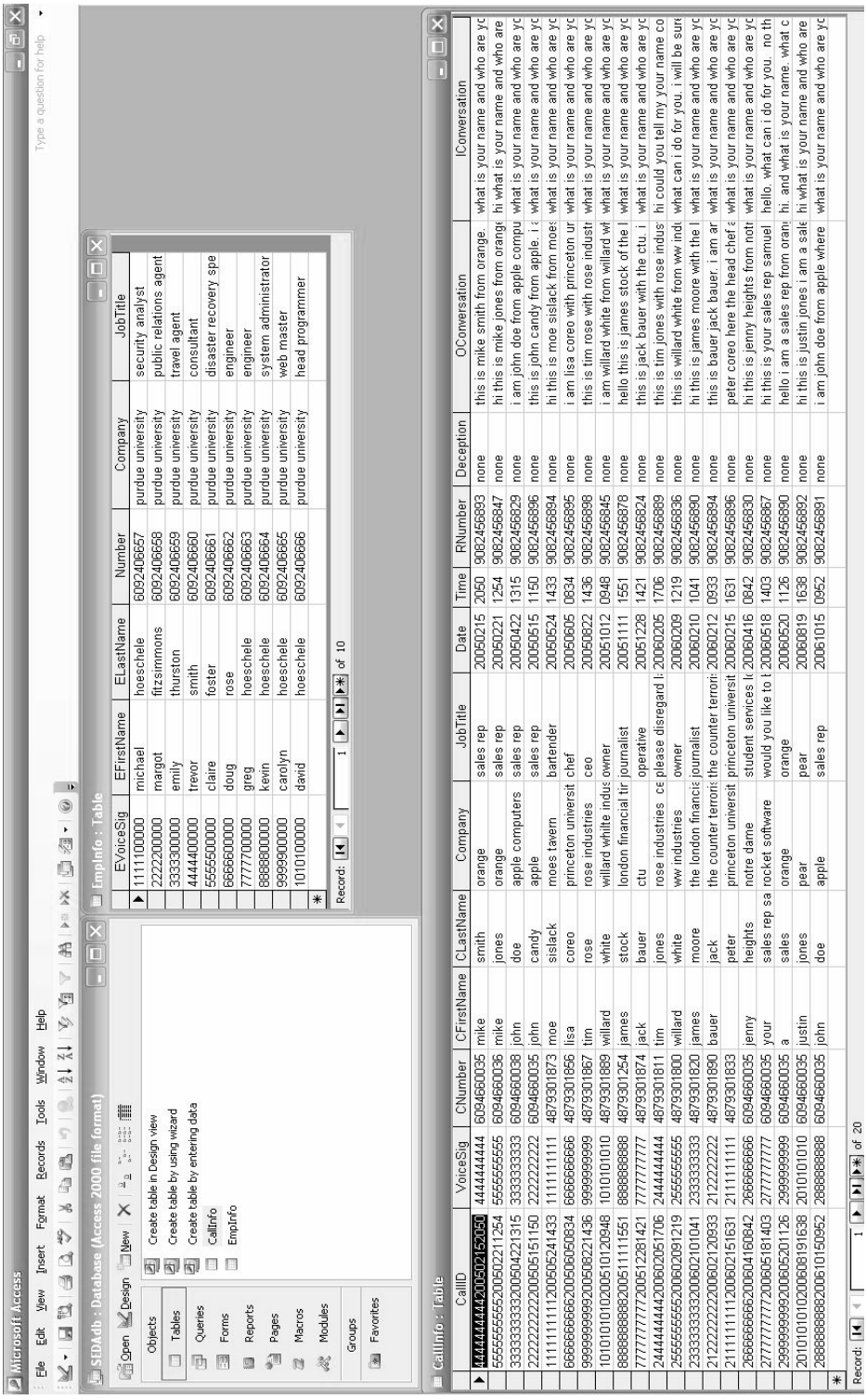
++++++++++++++++++ Attack Detected ++++++++++++
Attack: too frequent company change
Call ID: 111111111200707131035

++++++++++++++++++ Attack Detected ++++++++++++
Attack: too frequent company change
Call ID: 111111111200709131035

number of attacks found: 4
total possible attacks: 11

Appendix E

Output from Data Parsing Tests



Note: This is and example of the database records resultant of the data parsing tests.

number of attacks found: 0
total possible attacks: 20

Appendix F

Attack Signatures

Outsider Attack Signatures

read me your password
read me your passcode
read me your passphrase
tell me your password
tell me your passcode
tell me your passphrase
give me your password
give me your passcode
give me your passphrase
give me confidential info
give me secret info
give me top secret info
give me private info
tell me confidential info
tell me secret info
tell me top secret info
tell me private info
give me some confidential info
give me some secret info
give me some top secret info
give me some private info
tell me some confidential info
tell me some secret info
tell me some top secret info
tell me some private info
install this for me
install this on your system
install this on your machine
install this on your computer
tell me the secure server ip
tell me the secure server ip address
tell me the secure server name
give me the secure server ip
give me the secure server ip address
give me the secure server name
tell me your secure server ip
tell me your secure server ip address

tell me your secure server name
give me your secure server ip
give me your secure server ip address
give me your secure server name
tell me the ip of your secure server
give me the ip of your secure server
tell me the ip of the secure server
give me the ip of the secure server
tell me the ip address of your secure server
give me the ip address of your secure server
tell me the ip address of the secure server
give me the ip address of the secure server
give me the name of your secure server
tell me the name of your secure server
give me the name of the secure server
tell me the name of the secure server

Insider Attack Signatures

my password is
my passphrase is
my passcode is
the secure server name is
the name of the secure server is
the secure servers name is
the secure server ip address is
the secure server ip is
the secure servers ip address is
the secure servers ip is
the name of our secure server is
the secure server is named
our secure server is named

Appendix G

Purdue University Security Policy

This is the section of the policy that addresses electronically transmitted information.

Electronically Transmitted Information			
Action	**Public**	**Sensitive**	**Restricted**
			Recommendations on handling of restricted data doesn't apply to financial restricted accounts.
By Fax	No special requirement.	Machine must have limited access such that only those authorized can view. Otherwise, recipient must first agree that an authorized person will be present when the material is sent.	Machine must have limited access such that only those authorized can view. Otherwise, recipient must first agree that an authorized person will be present when the material is sent.
By Voice Mail	No special requirement.	No special requirement	Do not leave restricted information in voice mail message. Request call back.
By wireless or cellular technology	No special requirement.	Be aware that wireless cellular technology are not very secure. (Encryption suggested where applicable)	Do not transmit.
All other electronic communications (email, FTP, connections to administrative applications, etc).	No special requirement.	Encryption suggested	Encryption Required. **
** Note: It is expected that departments will (a) move toward encryption over time as new files are created or existing files modified and (b) pressure their vendors and exchange agencies to move in this same direction.			

Appendix H

Test Security Policy

Electronically Transmitted Information			
Action	**Public**	**Sensitive***	**Restricted****
By wireless or telecommunications technology	No special requirement.	Do not transmit.	Do not transmit.
* Data labeled as confidential or private, including company passwords, codes, or phrases.		** Data labeled as Top Secret or Secret, including the IP address, network name, or any login information for the secure server.	

Note: General phone policy: When receiving a call, employees must request the caller's name, company, and then job title. The order of these questions must be maintained.

Appendix I

System Configuration

System: Dell Latitude D600

OS: Microsoft Windows XP Media Center Edition version 2002 Service Pack 2

Database: Microsoft Access 2003 Service Pack 2

Development and testing environment: Eclipse version 3.14

Programming Language: Sun Microsystems Java 1.4.2_10

Appendix J

Structure of a Test Conversation