Semantic Analysis of Dialogs to Detect Social Engineering Attacks

Ram Bhakta and Ian G. Harris
Department of Computer Science
University of California Irvine, USA
hiten.bhakta@gmail.com, harris@ics.uci.edu

Abstract

Cyberattackers often attack the weakest point of system, which is increasingly the people who use and interact with a computer-based system. A great deal of research has been dedicated to protection of computer-based assets, but by exploiting human vulnerabilities, an attacker can circumvent many computer-based defenses. Phishing emails are a common form of social engineering attack, but the most effective attacks involve dialog between the attacker and the target. A robust approach to detecting a social engineering attack must be broadly applicable to a range of different attack vectors.

We present an approach to detecting a social engineering attack which uses a pre-defined **Topic Blacklist (TBL)** to verify the discussion topics of each line of text generated by the potential attacker. If a line of text from the attacker involves a topic in the blacklist, an attack is detected and a warning message is generated. Our approach is generally applicable to any attack vector since it relies only on the dialog text. Our approach is robust in the presence of the incorrect grammar often used in casual English dialog. We have applied our approach to analyze the transcripts of several attack dialogs and we have achieved high detection accuracy and low false positive rates in our experiments.

I. Introduction

A critical threat to information security is *social engineering*, the psychological manipulation of people in order to gain access to a system for which the attacker is not authorized [1], [2]. Cyberattackers target the weakest part of a security system, and people are often more vulnerable than a hardened computer system. All manner of system defenses can often be circumvented if a user reveals a password or some other critical information. Social engineering is a modern form of the confidence scam which grifters have always performed. Phishing emails, which fraudulently request private information, are a common version of the attack, but social engineering comes in many forms designed to exploit psychological weaknesses of the target. The use of modern communication technologies, including cellular phones and the internet, have greatly increased the reach of an attacker, and the effectiveness of the attack.

Social engineering attacks involve communication between the attacker and the victim in order to either elicit some information, or persuade the victim to perform a critical action. Information gathered might include explicitly secure information such as a credit card number, or seemingly innocuous information which can support a larger attack, such as the name of a coworker. An attacker might also convince the victim to perform tasks which would support an attack, such as going to a website.

Numerous experimental studies over the years have demonstrated the susceptibility of people to social engineering attacks [3], [4], [5], [6], [7]. The effectiveness of social engineering has encouraged attackers to use it more frequently, relying on social engineering as a component of larger attacks. A study by Verizon of security breaches in 2013 has shown that 29% of all security breaches involve social engineering to extract information for use primarily for phishing, bribery, and extortion [8]. These attacks were executed primarily via email but also in-person, via phone, SMS, websites, and other documents. The frequency and effectiveness of social engineering makes it a serious security issue which must be addressed. Phishing emails and websites are a class of social engineering attack which are simple, attempting to establish trust in a single communication to a victim. Phishing techniques are simple to deploy but they are not as effective as *dialog-based* attacks because gaining trust often requires a two-way communication with a victim. To our knowledge, there are no existing automated approaches to the detection of dialog-based social engineering attacks.

Previous work in the automatic detection of social engineering attacks is limited to emails and websites, and do not attempt to detect the more subtle class of social engineering attacks which are purely dialog-based. Other previous work has focused on the training of individuals about social engineering attacks in order to make them more aware and resistant in the future [9], [10]. User training can provide resistance to a wider range of attack types, but its effectiveness is inconsistent, depending on the abilities of individuals which can vary a great deal. An automated approach for social engineering detection is needed which can be applied to a broad range of attack types, requiring minimal effort from the individual user.

We present an approach to the detection of social engineering attacks by performing semantic analysis of all text transmitted to the victim to guess at the *topics* being discussed on each line. Each topic is then checked for its appropriateness. A statement is considered to be inappropriate if it either requests secure information or requests the performance of

IEEE ICSC 2015, February 7-9, 2015, Anaheim, California, USA
978-1-4799-7935-6/15/$31.00 ©2015 IEEE

a secure operation. The ability to evaluate the appropriateness of a question or command depends on the existence of a **Topic Blacklist (TBL)** which describes all forbidden topics of conversation with the victim. We assume that the TBL will be derived manually either based on a basic understanding common security requirements, or an existing security policy document associated with a system. All US federal agencies are required to provide information security by the "Federal Information Security Act of 2002" [11], and policy documentation is part of that requirement [12]. Although private industry is not required to provide such documentation, the significant cost associated with cyberattacks has led many companies to document their security policies as well.

The appropriateness of a statement with respect to security depends on the identity of the speaker and his relationship to the listener. If a trusted party asks for sensitive information then it may not represent a social engineering attack. In order to consider the speaker's identity during attack detection we would need to implement some form of authentication. Authentication is outside the scope of this paper, so we assume that the speaker is untrusted. However, the ideas presented in this paper can easily be extended to consider speaker identity if an authentication system is in use.

The remainder of the document is organized as follows. Section II summarizes previous related research efforts. The nature of the topic blacklist is described in Section III together with an example of how a blacklist is used to detect social engineering attacks. Section IV presents the detection algorithm we use to identify inappropriate topics in text. Experimental results are presented in Section V, and Section VI describes our conclusions.

II. Related Work

A number of approaches have been presented to identify phishing websites and phishing emails. Phishing website identification approaches observe the features of the website and apply a set of rules which distinguish anomalous website properties. Identifying features used include the existence of misleading URLs, the existence of specific images, client-side search history, and password requests [13]. Detection rules consider values of individual features and correlations between feature values, such as the inclusion of a company logo at a website whose URL is not related to the company. Attackers can duplicate many of the features of a good website, so researchers have explored the use of W3C DOM objects which are more difficult to duplicate [14]. In [15] each website is characterized by the term frequency/inverse document frequency (TF-IDF) value of the text found within the page. The words with the top TF-IDF values in a page are supplied to Google's search engine to find the page, assuming that a phishing website would not be produced as a Google search result.

Several techniques have been proposed to detect phishing emails by extracting features of the email header and body. Commonly used features include the use of IP-based URLs, URLs linked to new domains, HREF values which do not match the displayed link, and HTML emails which allows URL names to be masked [16]. Researchers in [17] focus on the detection of HTML phishing emails by automatically posting fake data to the associated websites and verifying the correctness of the responses. Machine learning techniques have been used to distinguish phishing emails based on term document frequencies of key terms in the email, and their similarity to known phishing emails [18].

Existing techniques for detection and prevention of the broader class of social engineering attacks depend on training potential victims to be resistant to manipulation. Training regimens have been proposed in previous work [10], [9] which educate users on the techniques used in previous attacks, and the importance of various pieces of information. A multi-level training approach has been presented which matches the degree of training to the responsibility level of the user [9]. Training techniques depend on the users awareness of his/her mental state and thought processes, referred to as *metacognition* [9]. A social engineering detection approach presented in previous work also depends on the user's metacognition in order to answer a sequence of security-related questions before providing data to an external agent [19]. In general, approaches which rely on the the cognitive abilities of the user will be unreliable since the mental state of users vary greatly, and can often be manipulated by a clever attacker.

III. Topic Blacklists

The **Topic Blacklist (TBL)** is the list of statement topics whose discussion would indicate a violation of security protocol. Each topic either describes sensitive data or a sensitive operation. Each topic is composed of two elements, an *action* and a *resource*. The *action* describes an operation which the subject may wish to perform. The *resource* is the resource inside the system to which access is restricted.

We assume that the TBL will be derived manually either based on a basic understanding common security requirements, or from an existing security policy document associated with a system. The TBL would be created to match the security requirements of the system which it is being used to protect. A TBL used to protect a regular individual would contain generic topics such as {"send", "money"} or {"tell", "social security number"}. A TBL used by a corporate entity would contain topics related to the business of that entity.

A. Blacklist Detection Example

To concretize our discussion on detecting social engineering attacks, we next motivate our approach with an example of attack detection. For this example we assume that social engineering is used to attack the information technology infrastructure of a corporate entity. We assume that there is a security policy document which has been manually processed to define an appropriate TBL. Figure 1 shows a statement found in the policy document which is used to define a TBL entry.

The statement in Figure 1 is manually analyzed to generate the TBL entry shown in Table I.

TBL 1: Networking equipment must not be manipulated.

Fig. 1. Security policy statement

	Action	Resource
Entry 1	manipulate	networking equipment

TABLE I
TBL ENTRY

For this example we assume that the social engineering attack is launched via texting, social media chat, or email, so that the dialog is already provided in text form. At some point during the dialog, the attacker makes the statement shown in Figure 2. The statement is a command which matches Entry 1 in Table I. The predicate of ATT 1, "reset", is a type of manipulation, so it will match the action "manipulate" in Entry 1. Also, the direct object of the predicate of ATT 1, "the router", is a type of networking equipment, so it matches the resource in Entry 1. Since the statement matches a TBL entry, access will be denied and a warning message is transmitted to the victim.

IV. Detection Algorithm

The top-level algorithm for scanning the dialog to identify attacks is shown in Figure 3. The outer loop scans through each line l in the text spoken by the attacker, $TEXT$. The inner loop iterates through each entry t in the TBL. Line 3 invokes the $MatchTopic$ function to determine if the topic of entry t is contained in line l. If the topic is found then the attack is detected and a warning message is produced.

The algorithm for the $MatchTopic$ function is shown in Figure 4. The purpose of the outer loop (line 1) is to scan through each token in the line and compare it to $t.action$, the action associated with TBL entry t. The comparison to $t.action$ is performed on line 2. If the action is found then the inner loop is entered (line 3) to scan through the remaining tokens on the line and compare them to $t.resource$, the resource associated with t. If both the action and resource are found then the function returns TRUE on line 5. The $Compare$ function called on lines 2 and 4 performs a string comparison, but it also compares the first argument to any synonyms of the second argument.

V. Results and Discussion

We have evaluated our system by applying it to three social engineering dialogs which we refer to as Corpus A [20], Corpus B, [21], and Corpus C [22]. Corpus A and B are transcripts of dialogs via Facebook and Corpus C is an email

ATT 1: Please reset the router.

Fig. 2. Inappropriate statement within a social engineering attack

1. foreach $l \in TEXT$
2. foreach $t \in TBL$
3. if MatchTopic(l, t)
4. ATTACK_DETECTED

Fig. 3. Social Engineering Detection Algorithm

1. for $i = 0$ to |tokens in l|
2. if Compare($l[i]$, $t.action$)
3. for $j = i + 1$ to |tokens in l|
4. if Compare($l[j]$, $t.resource$)
5. return TRUE
6. return FALSE

Fig. 4. MatchTopic Algorithm

dialog. Table II shows information about each corpus. The second column is the number of communications in the dialog, which is the number of chat messages for A and B, and the number of emails for C. The third column is the number of lines of text in each corpus. The fourth column is the number of lines in the corpus which contain inappropriate topics.

Corpus	Comms	Lines	Violations
A	136	190	2
B	24	30	4
C	15	325	2

TABLE II
SOCIAL ENGINEERING ATTACK TRANSCRIPTS

Table III shows the TBL which we use. The TBL was manually generated based on our understanding of common generic social engineering attack goals. Each root word in the TBL is associated with a set of synonyms which are considered to be equivalent. The synonyms for each root word which are considered by our tool are not shown.

Action	**Resource**
send	money
send	sensitive data
call	number
visit	website

TABLE III
TOPIC BLACKLIST

Table IV contains the suspicious parts of the lines in each corpus which were found to be inappropriate. In each line, the words highlighted in bold are the action and resource words which matched TBL entries.

Line
"can you please **loan** me some **money**?"
"if you **upload $500**" "**SEND** ME A **CUT**" "**SEND** ME THE **14 NUMBERS**" "**TEXT** ME RIGHT NOW SO I CAN HAVE YOUR **NUMBER**"
"**contact** our family attorney on his direct roaming telephone **number**" "**give** me your tel **number**"

TABLE IV
INAPPROPRIATE STATEMENTS

As shown in Table II, there are a total of 8 inappropriate lines out of a total of 545 lines of text. Our tool detected 7 of the 8 inappropriate lines, with no false positives. We compute the *precision* and *recall* metrics according to their standard definitions shown in equations 1 and 2. In these equations, tp is the number of true positives, fp is the number of false positives, and fn is the number of false negatives.

$$precision = \frac{tp}{tp + fp} \quad (1)$$

$$recall = \frac{tp}{tp + fn} \quad (2)$$

Our results contain 8 true positives, 0 false positives, and 1 false negative. The precision of our approach is 100% and the recall is 88.9%. The CPU time on an Intel i7 processor, 3.5 GHz, to evaluate each corpus is 0.31 seconds, 0.28 seconds, and 0.34 seconds for A, B, and C, respectively. The performance clearly indicates that this approach can be used to perform detection during a dialog in real time.

The inappropriate sentence which went undetected by our tool in Corpus A was the following, "go to western union office and **send it** ok". This was not detected because the tool could not match the word "it" to any resource in the TBL. By examining the context of the sentence in the dialog, it is clear that the pronoun "it" refers to the term "300dollars" which the attacker mentioned previously. The problem of mapping a pronoun to the noun to which it refers is a well understood problem in natural language processing called anaphora resolution [23]. Recall can be improved by implementing an existing approach for anaphora resultion in order to detect this type of violation.

In the future we intend to identify more social engineering attack transcripts which we can use to identify weaknesses in our detection approach. The examples which we use here are all examples of relatively generic attacks which could be broadly applied to most people. We will identify *targeted* social engineering attacks which are designed for specific people or institutions. Detection of targeted attacks will present challenges in in defining the topic blacklist to protect institution-specific assets.

In order to avoid false positives in more complex attack examples, it may be desirable to increase the *granularity* of the topic matching process. It is possible to imagine examples where a blacklisted action and resource are mentioned in a sentence without being inappropriate. For example, a question like, "How do you shut down the firewall?" is likely to be inappropriate, but the sentence, "I tried to shut down the process but the firewall stopped me" is not a problem. Both sentences include the action "shut down" and the resource "firewall", but the words serve different roles in each sentence. The word "firewall" is the direct object of the action in the first sentence, but not in the sentence. Precision may be improved in complex examples by considering the semantic roles of each word in the sentence when identifying the sentence topic. The analysis necessary to determine the semantic roles of words in a sentence is a potential topic for future work.

VI. Conclusions

We present an approach to detect social engineering attacks by verifying discussion topics against a topic blacklist. The approach is robust enough to effectively analyze the language of real attacks, including the incorrect English which is often used in casual conversation. The definition of the topic blacklist is manual but we do not believe that it is onerous for a person with an understanding of the security requirements of the system being protected. The performance of our tool is good enough to provide attack warnings in real time during a conversation to prevent the victim from violating security protocol.

References

[1] C. Hadnagy and P. Wilson, *Social Engineering: The Art of Human Hacking*. Wiley, 2010.

[2] K. Mitnick and W. Simon, *The Art of Intrusion: The Real Stories Behind the Exploits of Hackers, Intruders and Deceivers*. Wiley, 2009.

[3] T. Greening, "Ask and ye shall receive: A study in social engineering," *SIGSAC Rev.*, vol. 14, no. 2, Apr. 1996.

[4] A. Karakasilitiosis, S. M. Furnell, and M. Papadaki, "Assessing end-user awareness of social engineering and phishing," in *Australian Information Warfare and Security Conference*, 2006.

[5] M. Workman, "A test of interventions for security threats from social engineering," *Inf. Manag. Comput. Security*, vol. 16, no. 5, 2008.

[6] G. L. Orgill, G. W. Romney, M. G. Bailey, and P. M. Orgill, "The urgency for effective user privacy-education to counter social engineering attacks on secure computer systems," in *Proceedings of the 5th Conference on Information Technology Education*, 2004.

[7] T. Bakhshi, M. Papadaki, and S. Furnell, "A practical assessment of social engineering vulnerabilities." in *Human Aspects of Information Security and Assurance (HAISA)*, 2008.

[8] *2013 Data Breach Investigations Report*. Verizon, 2013. [Online]. Available: http://books.google.com/books?id=YXi0nQEACAAJ

[9] D. Gragg, "A multi-level defense against social engineering," SANS Institute, Tech. Rep., December 2002.

[10] J. Scheeres, *Establishing the Human Firewall: Reducing an Individual's Vulnerability to Social Engineering Attacks*. Bibliioscholar, 2012.

[11] "Federal information security management act of 2002," 2002, title III of the E-Government Act of 2002 (Public Law 107-347).

[12] "Minimum security requirements for federal information and information systems," National Institute of Standards, Tech. Rep., March 2006, fIPS Pub 200.

[13] N. Chou, R. Ledesma, Y. Teraguchi, D. Boneh, and J. C. Mitchell, "Client-side defense against web-based identity theft," in *Network and Distributed Systems Security Symposium (NDSS)*, 2004.

[14] Y. Pan and X. Ding, "Anomaly based web phishing page detection," in *Computer Security Applications Conference, 2006. ACSAC '06. 22nd Annual*, 2006.

[15] Y. Zhang, J. I. Hong, and L. F. Cranor, "Cantina: A content-based approach to detecting phishing web sites," in *Proceedings of the 16th International Conference on World Wide Web*, 2007.

[16] I. Fette, N. Sadeh, and A. Tomasic, "Learning to detect phishing emails," in *Proceedings of the 16th International Conference on World Wide Web*, 2007.

[17] M. Chandrasekaran, R. Chinchani, and S. Upadhyaya, "Phoney: mimicking user response to detect phishing attacks," in *International Symposium on a World of Wireless, Mobile and Multimedia Networks, 2006 (WoWMoM)*, 2006.

[18] V. Ramanathan and H. Wechsler, "phishgillnetphishing detection methodology using probabilistic latent semantic analysis, adaboost, and co-training," *EURASIP Journal on Information Security*, 2012.

[19] M. Bezuidenhout, F. Mouton, and H. S. Venter, "Social engineering attack detection model: Seadm." in *Information Security for South Africa (ISSA)*, 2010.

[20] T. Wayne, "My Three-Month Facebook Dialogue With A Scammer From Malaysia Pretending To Be A Beautiful Woman," December 2011, http://www.theawl.com/2011/12/my-three-month-facebook-dialogue-with-a-scammer-from-malaysia-pretending-to-be-a-beautiful-woman.

[21] D. C. Morch, "Scam conversation started today," June 2013, https://www.facebook.com/4Catash/posts/4917971716176.

[22] T. Morrel, "Conversations with a Nigerian Bank Scammer," February 2002, http://www.yrad.com/convo2.htm.

[23] R. Mitkov, "Anaphora resolution: the state of the art, working paper," 1999, based on the COLING'98/ACL'98 tutorial on anaphora resolution.