Detecting Telephone-based Social Engineering Attacks using Scam Signatures

Ali Derakhshan
aderakh1@uci.edu
University of California, Irvine

Ian G. Harris
harris@ics.uci.edu
University of California, Irvine

Mitra Behzadi
mbehzadi@uci.edu
University of California, Irvine

ABSTRACT

As social engineering attacks have become prevalent, people are increasingly convinced to give their important personal or financial information to attackers. Telephone scams are common and less well-studied than phishing emails. We have found that social engineering attacks can be characterized by a set of *speech acts* which are performed as part of the scam. A speech act is statements or utterances expressed by an individual that not only conveys information but also performs an action [7]. Although attackers adjust their delivery and wording on the phone to match the victim, scams can be grouped into classes that all share common speech acts. Each scam type is identified by a set of speech acts that are collectively referred to as a **scam signature**. We present a social engineering detection approach called the Anti-Social Engineering Tool (**ASsET**), which detects attacks based on the semantic content of the conversation. Our approach uses word embedding techniques from natural language processing to determine if the meaning of a scam signature is contained in a conversation. In order to evaluate our approach, a dataset of telephone scams has been gathered which are written by volunteers based on examples of real scams from official websites. This dataset is the first telephone-based scam dataset, to the best of our knowledge. Our detection method was able to distinguish scam and non-scam calls with high accuracy.

CCS CONCEPTS

• **Security and privacy → Social engineering attacks**.

KEYWORDS

Social engineering attacks; Natural language processing; Scam call detection

ACM Reference Format:
Ali Derakhshan, Ian G. Harris, and Mitra Behzadi. 2021. Detecting Telephone-based Social Engineering Attacks using Scam Signatures. In *Proceedings of the 2021 ACM International Workshop on Security and Privacy Analytics (IWSPA'21), April 28, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3445970.3451152

1 INTRODUCTION

Social engineering is the act of manipulating people in order to gain something of value [21, 30]. Social engineering is not new, but with the advent of computer-mediated communication, it becomes possible to protect users from these attacks. Email phishing has been shown to be an effective attack over the years, consistently deceiving a broad range of people [23]. Attackers increasingly seek financial and sometimes political benefit by stealing personal information from individuals and organizations [24]. The Verizon 2019 Data Breach Investigations Report [37] states that 32% of all breaches included phishing and 78% of all cyber-espionage which involved state-affiliated actors.

Telephone-based social engineering attacks have several properties which distinguish them from phishing email attacks and make their detection difficult. Telephone calls do not have header information or other meta-data which is often used to identify suspicious email attributes such as a false message origin. Telephone calls do have caller ID but this is easily spoofed [19]. Telephone scams may be pre-recorded, but they often involve real-time communication with a live attacker. When a live attacker is involved, the content of the single type of attack can vary as the attacker adjusts the dialogue according to the match the vulnerabilities of the victim [14, 20].

Although the detailed wording of a telephone scam will change with each victim, telephone scams do have recognizable patterns which can be used for detection. For instance, an IRS scam will include the attacker claiming to represent the IRS, or a romance scam will include the attacker expressing affection for the victim. Many governmental organizations actively track and characterize existing scams in order to notify the public. Federal organizations which announce the properties of different scams include the Federal Trade Commission [18], the Federal Communications Commission [41], and the U.S. Marshals Service [42]. In spite of the work that these agencies perform to notify the public, many people are either never exposed to their warnings, or they do not remember the warnings in the moment of an actual attack.

1.1 Scam Signature

The basic building block of an attack is a *speech act* [7], an utterance something expressed by an individual that not only conveys information but also performs an action. For example, the sentence "I would like pizza, can you pass to me?" is a directive speech act since it commands the listener to deliver a pizza. We define a **scam signature** as a set of utterances that perform speech acts that are collectively unique to a class of social engineering attacks. These utterances are the key points, fulfilling the goal of the scammer for that attack. A scam signature uniquely identifies a class of social engineering attacks in the same way that a malware signature uniquely identifies a class of malware.

A sample of an IRS scam is shown in Figure 1. Although the text of this example may vary, it performs the same essential speech

IWSPA'21, April 28, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8320-2/21/04.
https://doi.org/10.1145/3445970.3451152

acts which are shown in Figure 2. The sentences which perform the speech acts are labeled with superscripts a and b.

Scammer: Hello, is this Bob Smith?
Victim: Yes, this is he.
Scammer: "I'm John, calling from the Social Security Administration. It has come to our attention that there has been suspicious activity on your account spread out over the last six months. Due to participation in highly illicit activities, we have contacted law enforcement agencies to suspend your social security number effective immediately."[a]
Victim: What? I haven't done anything illegal. Could my identity have gotten stolen? I was emailed about a breach in a government database recently...
Scammer: "Tracing your account activity over the past 10 years, it does seem likely that this recent string of activity is due to someone else using your identity for their illicit activity."[b]

[a]First
[b]Second

Figure 1: Example of IRS scam.

First: we will suspend your social security number on an immediate basis
Second: as we have received suspicious trail of information in your names

Figure 2: IRS Scam Signature

1.2 Scam Detection Approach

We present the Anti-Social Engineering Tool (ASsET) which detects attacks by using scam signatures in the same way that malware signatures are used by anti-virus tools. The key aspect of our approach is the use of *word embeddings* and *sentence embeddings* to identify statements in a conversation which have the same meaning as sentences in a signature [27]. It is common in natural language processing to represent the meaning of text as an n-dimensional vector. A word embedding is a vector representation of words with the property that two words with the same meaning are represented by vectors which are close to each other. Word embeddings have been extended to sentence embeddings [10] which represent the meanings of phrases and sentences.By using sentence embeddings, we can identify sentences in a conversation with the same meaning as signature sentences, even when the meaning is expressed differently in English.

1.3 Scam Dataset

To evaluate our scam call detection system, we need samples of scam calls. Although telephone scams occur frequently, we could not find a large number of actual telephone scam conversations. The main reason for lack of such a dataset is that victims do not usually record their phone calls, and if they did, they are embarrassed to share their conversation with a scammer that leads to their financial or identity loss. As there is no dataset of scam calls, we performed a study to create a dataset. We have collected scam conversation samples from legitimate official websites, which advertise scams to increase public awareness. Then we gave these samples to volunteers to write versions of these attacks in their own words. The scam call dataset is presented in section 4 and we believe that it is the first such dataset available.

2 SCAM SIGNATURES

We define a **scam signature**, as a set of utterances that each contains one or more speech acts[7], that collectively fulfill the goal of scammer in a scam call. A speech act is some utterance that not only conveys meaning but also performs an action. For example, the sentence "Can, you pass me the pizza, I want some.", is a directive speech act which commands the listener to perform an action. Attackers can change elements of an attack, but key characteristics of the attack remain consistent. There are many law enforcement and news organizations which identify key characteristics of common scams in order to increase public awareness [1, 25].

For example, in the IRS scam calls, the "lawforseniors" website[1], created a few highlights that indicates what could be a scam. One highlight is that the IRS never asks for credit card credentials on the phone. Another highlight is that IRS never says that "If you don't pay money immediately, you would be arrested". Moreover, you always have the opportunity to appeal. Based on these points, if someone calls and claims to be from IRS and asks for credit card credentials, or ask for money and treats, you would be arrested if you don't pay, the call is certainly a scam call.

2.1 Manual Scam Signatures

The key characteristics of many common scams are well known and are available in the public domain. Using these scam descriptions, it is straightforward to manually create scam signatures. In order to demonstrate this process, we have created scam signatures for five known scams by selecting subsets of their sentences found in publicly-available documents. Table 1 shows the sentences contained in each of the scam signatures that we created. The *Signature Number* column identifies each scam as well as the public source where the scam description was found. Each signature is a set of 1 to 3 sentences that were taken directly from the published scams. When the number of scams examples available is limited, creating manual scam signatures is an option.

2.2 Scam signatures using clustering

When a sufficient number of scam examples are available, a scam signature can be defined automatically by clustering based on utterance similarity. Using clustering techniques, we can find the patterns in conversations vectors by finding the clusters' centroids and using them as signature vectors, identifying the scam.

We used k-means clustering for this purpose. For the number of samples that we had, we set the number of clusters as the square root of the length of shortest conversation for that scam. We performed k-means clustering to generate a set of centroids which are used as the signature vectors. For example, in the IRS example in Figure 1, the conversation has ten utterances. If we have five conversations with ten utterances, the number of clusters would be 3, as it is the

Signature Number	Utterances in each Signature
Signature 1[5]	we will suspend your social security number on an immediate basis
	as we have received suspicious trail of information in your names
Signature 2[2]	revert as soon as possible on our number, before we begin with the legal proceedings.
Signature 3[4]	So you never received anything showing dollars
	we have audit of your taxes between some years.
	your bank account will be seized, your credit report will be spoiled, your passport along with State ID will be seized.
	Do you have the money with you?
Signature 4[2]	we will suspend all bank accounts and tax returns bearing your name and social security number
	To review immediate rights and details, and avoid all further proceedings, please contact our firm
Signature 5[3]	you can make a lot of money in a few short months
	you invest you will receive dollar return on your money in just six months and there is no risk of loss
	the deal is for today only. The opportunity will be gone tomorrow

Table 1: Scam Signatures

floor square root of the 10. we would have 50 utterance vectors, and we would have 3 clusters.

We obtained the cluster centroids on the training data for each scam. These centroids become scam signatures. So in this approach, instead of manually creating sentences as a scam signature and then vectorizing them, we obtain the signature vectors directly using clustering, so they do not correspond to equivalent sentences in a scam.

In the test data, we compare the conversation similarity to each signature in the same way we did it for manual signatures, presented in Algorithm 1. For each scam, the number of clusters is different. So the average value of the "f_similarity" score for train conversations would be different for various scams(scam signatures), which might lead to a bias to prefer to select some scams over others. So we normalize this score for each scam on the test set by the average similarity scores in the training set.

To find the threshold, we calculated the f_similarity score of the training conversations to the signatures. The scam conversations generally have a higher similarity than non-scam conversations to the signatures. So the threshold must be in the middle to minimize misclassification error. We get the average similarity of the scam and non-scam conversations to the signatures and set the middle of the averages to be the threshold.

3 SCAM DETECTION

We present a social engineering detection approach called Anti-Social Engineering Tool (ASsET), which detects attacks based on the conversation's semantic content.

Each scam signature contains a set of utterances, and we check the existence of each of them in the conversation. Each of these utterances includes one or more speech acts that their presence in a conversation is a sign of a scam. These speech acts themselves can be expressed in many different ways.

3.1 Finding Meaning in Text

We need a way to compare the meanings of two utterances to determine if they perform the same speech acts. A single speech act can be expressed in many different ways, and our comparison approach must be independent of this variation. To perform these comparisons, we use the concept of *word embeddings* and *sentence embeddings*, which is a well-accepted approach to capturing the meaning of an utterance as an n-dimensional vector [27]. The great benefit of word embeddings is that the embedding vectors have the property that two utterances with similar meaning will have similar vector representations. Using embeddings allows the meaning of two utterances to be compared by simply computing the dot product of the two vectors. The first accepted word embedding approach was *word2vec* [29], but there have been several approaches more recently including Glove [32] and ELMo [33]. The assumption is that two words have similar meaning if they are used in the same context inside utterances. Words which are used in the same context, over a large corpus of utterances, will be represented by similar embedding vectors. Word embeddings can be used to compare the meanings of individual words, but we need to compare the meanings of more complex utterances such as entire sentences. Several approaches have been proposed to produce embeddings for sentences, and we have chosen to use the Universal Sentence Encoder approach presented by Google Research [11]. We use their Deep Averaging Network model, which averages the embeddings of the words and bigrams contained in the sentence and passes the result through a feedforward deep neural network.

3.2 Scam Detection Algorithm

Algorithm 1 shows the pseudo-code of our detection approach. The inputs of this Algorithm are a conversation C and the set of all scam signatures, *allSignatures*. A conversation C is a list of utterances spoken by the communicating parties which must be evaluated to see if a scam is occurring. The set *allSignatures* contains all scam call signatures that we want to detect in a conversation. The set *allSignatures* is shown in Table 1. The output of the algorithm is either the signature of the scam, which is contained in the conversation, or NULL if no scam is detected.

At code lines 1 and 2 we vectorize the sentences in the conversation C and we vectorize the utterances in the *allSignatures* set. Vectorization is performed using the *embed* function which is the the Unviersal Sentence Encoder described in previous work [9]. The vectorization of each sentence gives us a representation of its meaning which can be used for comparison. In code line 3, we have

```
   input  : conversation C
   output : allSignatures
1  C=embed(C);
2  allSignatures=embed(allSignatures);
3  bestSim=0;
4  for signature in allSignatures do
5      sigSimilarity=f_similarity(C, signature);
6      if sigSimilarity> bestSim then
7          bestSim=sigSimilarity;
8          bestSig = signature
9      end
10 end
11 if bestSim > Threshold then
12     Return bestSig;
13 else
14     return NULL;
15 end
   Result: Best Matching Signature Or NULL
```

Algorithm 1: Detecting scam signatures

defined *bestSim=0*, which is used to keep track of the best similarity found between the conversation and a signature.

	v_1	v_2
u_1	0.057	0.075
u_2	0.024	0.027
u_3	0.538	0.307
u_4	0.318	0.298
u_5	0.258	0.357
u_6	0.154	0.110
u_7	0.259	0.175
u_8	0.018	0.026
u_9	0.458	0.301
u_{10}	0.118	0.056

Figure 3: f_similarity for the IRS example in Figures 1 and 2

```
1  Function f_similarity(Conversation C, signature S)
       Data: a conversation C and a signature S
       A conversation is a list of u utterance vectors
       A signature S is a list of v vectors.
2      similarityTable=innerProduct(C, S);
3      maxMatches=signatureUtteranceMax(similarityTable);
4      sigSimilarity=mean(maxMatches);
       Result: A value showing the similarity of a
               conversation to a signature
5  end
```

Algorithm 2: The f_similarity function

On code lines 4 to 10, we find the **most similar signature**, and it's **similarity score** (*bestSim* variable). At first, we find the similarity of a conversation to each signature. The f_similarity function accepts a conversation and a signature and returns a similarity value to that signature. To understand this function better, Figure 3 gives a real example of this calculation based on our IRS example. The conversation in Figure 1 has ten utterances, each of which corresponds to a row in Figure 3. The signature that we are comparing to, shown in Figure 2, contains two signature vectors, v_1 and v_2, each of which corresponds to a column in Figure 3. The contents of the table shown in Figure 3 are the similarity values between the corresponding utterance and signature vectors. The similarity values are computed as the inner products between the corresponding vectors. Then we find the most similar utterance to a signature vector(max of each column), which are highlighted cells in this Figure. The third utterance in the conversation is the most similar utterance to the first vector of the signature. The fifth utterance in the conversation is the most similar utterance to the second vector of the signature. Then we average these values to get the similarity of a conversation and a signature.

In code lines 6 to 9, we keep track of the most similar signature, and it's similarity. After going through all the signatures, we have the most similar signatures, and it's similarity score(*bestSim* variable).

In code lines 11 to 15, we check conversation to see whether it's most similar signature is above a threshold or not. If it is below the threshold, we classify that conversation as a non-scam and return *NULL* to signify this. If it is above the threshold, it is a scam conversation, and we return the signature. We apply this to all the conversations in our test set.

4 SCAM DATASET

To evaluate our detection approach, we need a dataset composed of both scam conversations and non-scam conversations. For the non-scam conversations, we use the CallHome dataset, part of the TalkBank project [28], which contains transcripts of 140 spoken English conversations(120 Participants and 176 files which 140 of them are conversations). Although there exist non-scam datasets, to the best of our knowledge, there is no existing dataset of telephone scam conversations. Although we did find some individual scams which were publicly available, we could not find enough publicly-available scams to enable us to evaluate our approach. There are many datasets of phishing emails, but telephone scams are of particular interest to us due to their unique nature. A likely reason for the lack of real telephone scam datasets is the existence of wiretap laws in some states which prevent the recording of calls without the consent of both parties.

We have conducted a human subject study to generate a set of scams that can be used to evaluate scam detection. We recruited 15 computer science graduate students from our university asked them to write scam conversations in their own words. Each subject was given five prompts taken from public websites, which presented examples of different types of telephone scams. Each participant was requested to write their own version of each of the five types of scams, using the prompts as a guide. The prompts, were derived

from publicly available websites that raise public awareness of scams.

Scam Number	# Conv.	Mean	Variance	Max	Min
1[5]	15	1.6	5.04	10	1
2[2]	15	1	0	1	1
3[4]	15	24.46	41.58	30	9
4[2]	15	1	0	1	1
5[3]	15	17.8	13.35	26	10

Table 2: Statistics of the lenght of scam conversations in the dataset

Table 2 shows statistics describing our scam dataset. The dataset contains 15 conversations of each scam type, with a total of 75 scam conversations in all. The table also shows the variation in the length (lines of text) of the scams produced by the participants. This dataset has been made publicly available.[1]

5 RESULTS

Our dataset consists of the 75 telephone scam call samples generated by our study and the 140 non-scam telephone call transcripts from the CallHome dataset[28].

Training/Test set. To find our model parameters, threshold and the cluster centroids, we need to have a training set separated from the test set to ensure the parameters are selected without observing the test set. We select the training set by choosing 30 non-scam conversations and 40 scam conversations, 8 scam conversations for each of the 5 types of scams. The test set includes 110 non-scam conversations and the remaining 35 scam conversations (7 of each scam type) which are not in the training set.

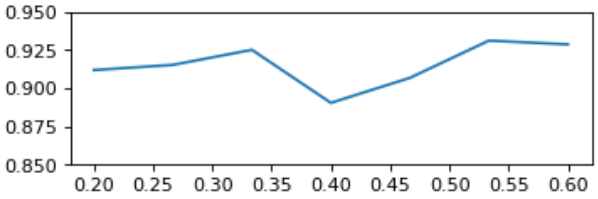

Figure 4: k-means Clustering accuracy by increasing the percentage of the training data

In Figure 4, we gradually increased the percentage of the scam conversations in the training set(The number of non-scam conversations is fixed) and plotted the test accuracy(remaining conversations) using our clustering method. The accuracy is generally high, even with 20 percent of the scam conversations, and it slightly increases when using about half of the scam conversations for training. We chose to use approximately 53.3% of the scam conversations because that it where the peak accuracy is achieved.

[1]The link to the dataset repository is removed for anonymity.

5.1 Similarity Threshold

Our detection approach depends on a similarity threshold which is used to determine whether or not a conversation is sufficiently close to a signature.

To find the threshold we evaluate the similarity scores of the conversations in the training set. To find the similarity threshold we used the Equation 1.

$$Threshold = \frac{averageScore(scam) + averageScore(nonScam)}{2} \quad (1)$$

5.2 Manual Signatures result

We show two sets of results. In the first set of results, we perform multi-class classification, classifying each call as either non-scam or one of the five types of the scam which we consider. In the second set of results, we perform binary classification with two classes, either scam or non-scam.

Table 3 shows the results of multi-class classification. For each scam type, we have calculated the Accuracy, Precision, Recall, and F-measure. The accuracies for all scam types are higher than 90%, but the recall is relatively low for scam type 2. We believe that this is because the prompt provided to the study participants for this scam was only two sentences long. As a result, the participants included a larger amount of elaboration, causing the scams to diverge from the prompt."

	Accuracy	Precision	Recall	F-measure
non-scam	0.903	0.907	0.973	0.939
Scam 1[5]	0.993	0.875	1.000	0.933
Scam 2[2]	0.959	0.667	0.286	0.400
Scam 3[4]	0.979	0.833	0.714	0.769
Scam 4[2]	0.972	0.800	0.571	0.667
Scam 5[3]	0.972	0.800	0.571	0.667
Mean	0.963	0.814	0.686	0.729
Variance	0.001	0.006	0.061	0.034

Table 3: Results of multi-class classification

Table 4 shows the overall performance of multi-class classification by averaging the results in Table 3. Macro averaging computes an average of the F_measure for each class, while micro averaging computes the F_measure from the average precision and recall scores of each class.

	Macro averaging	Micro Averaging
F_measure	0.729	0.889

Table 4: F_measure general performance using Micro and Macro averaging

Table 5 shows the results of binary classification, either scam or non-scam. These results better indicate our method's ability to detect scam calls as some scam types are similar.

	Accuracy	Precision	Recall	F-measure
Non-Scam	0.903	0.906	0.972	0.938
Scam	0.903	0.888	0.685	0.774
AUC		0.829		

Table 5: Results of binary classification

5.3 k-means clustering results

For the clustering results, the train and the test sets are the same. The only difference is the signature vectors in this experiment, and they are automatically being created using the k-means clustering method. In the test conversations, we check if the score of the most similar signature is above a threshold(we obtained using the training set and described in section 5.1); Then, we classify the conversation as a sample of that scam. Otherwise, we classify it as a non-scam conversation.

In Table 6, the accuracy, precision, recall, and f-measure of our classification are presented. As it can be seen in all the scam types, the precision is "1.0", it means that if we have classified a conversation scam, it would be a genuine scam(no false positive), but we might have missed some conversations. Having no false positive is crucial since we do not want to detect a vital call as a scam and interrupt it. We want to detect a call as a scam only when we are sure about it. This result shows that this approach is working with high accuracy without making problems for regular calls. We do not need to create signature sentences manually, as we can get them automatically using the k-means clustering approach.

	Accuracy	Precision	Recall	F_measure
non-scam	0.931	0.917	1.000	0.957
Scam 1	1.000	1.000	1.000	1.000
Scam 2	0.979	1.000	0.571	0.727
Scam 3	0.979	1.000	0.571	0.727
Scam 4	0.986	1.000	0.714	0.833
Scam 5	0.986	1.000	0.714	0.833
Mean	0.977	0.986	0.762	0.846
Variance	0.000	0.001	0.032	0.011

Table 6: Results of multi-class classification with k-mean clustering

By comparing Table 6 with Table 3, we can see that the results for the k-means clustering method achieves better results than the manual signature method. Table 7, which reports F_measure using Macro and Micro averaging, shows that these results are better than manual signatures as well.

	Macro averaging	Micro Averaging
F_measure	0.846	0.931

Table 7: F_measure general performance using Micro and Macro averaging with k-means clustering

Table 8 shows the results when we group all the scams, and perform a binary classification between scam and non-scam classes.

	Accuracy	Precision	Recall	F-measure
Non-Scam	0.931	0.916	1.0	0.956
Scam	0.931	1.0	0.714	0.833
AUC		0.857		

Table 8: Results of binary classification in k-mean clustering

Also, we have provided the AUC of this binary classification. The precision or the detecting scams is "1.0", but the recall is "0.714", which are good results, but the recall can still be improved.

6 RELATED WORK

We summarize related research in the detection of social engineering and phishing attacks, but a more detailed exposition of this research can be found in a survey on the topic [13]. Previous work in the detection of social engineering attacks can be viewed according to the algorithmic approach used as well as the features used by the algorithm. Several of the early approaches are rule-based [12, 22, 43] while most newer techniques use some form of machine learning [6, 8, 17, 26, 31, 34, 35]. Rule-based algorithmic approaches have used statistical methods to identify anomalous emails or web pages. For example, the CANTINA technique [43] evaluates web pages by using the TF-IDF metric to rank the words found on a web page and then sends the top 5 words to a search engine to see if the page is found. An approach to spear phish detection [22] produced excellent detection results by ranking click-in-mail events based on their count of "suspicious" features as compared to all other events. Phishing detection approaches using machine learning extract a set of features from phishing emails and then create a classifier to identify phishing emails based on the features. Most machine learning papers develop many different classifiers for comparison. For instance, researchers in [26] use Multi-Layer Perceptron (MLP) Neural Networks, Naive Bayes Classification, Decision Tree, and Random Forest.

The success of previous approaches to phishing detection depends heavily on their selection of features to extract from the email or web page being evaluated. Many features are based on "metadata" related to an email including MTP headers, NIDS logs, LDAP logs, and cc lists [15, 22, 36]. Some features used consider the content of the email, including whether or not it contains HTML and JavaScript [35] and the structure of UML links found in the email [6, 8]. A common feature of the content is the frequency of certain words whose use are associated with phishing emails or web pages [6, 8, 26, 35]. For example, in [40] researchers associate a set of words with a sense of urgency, which is commonly conveyed in phishing emails. Some researchers used semantic features in the emails to detect scams[38, 39]. Also, they have used NLP-techniques to detect email spams, and they showed that NLP-techniques could robustly detect them[16].

7 CONCLUSIONS

We present the idea of a scam signature to identify a class of social engineering attacks uniquely, in much the same way that malware signatures are used to identify malware. The signatures are based on the content of the conversation rather than any meta-data, so they

can be applied to telephone-based and in-person attacks which have no meta-data. We demonstrate the effectiveness of scam signatures by using them to implement a social engineering detection tool, ASsET, which compares signatures to a conversation to determine if an attack is being performed. A sentence embedding approach used widely in the NLP domain is used to compare the meaning of the signature to the meaning of sentences in the conversation. In order to demonstrate the effectiveness of our approach for the detection of telephone-based attacks, we have gathered a set of realistic attacks by performing a human subject study in which volunteers created scam scripts based on a set of existing telephone scams, which were provided as prompts. By evaluating our detection approach with our scam dataset, together with a set of non-scam conversations, we have shown that our detection approach has high accuracy and F-score. We have also made our telephone scam dataset publicly available to support future research in social engineering attacks.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 1813858.

REFERENCES

[1] [n.d.]. *TAX SCAMS THAT TARGET MILLIONS OF AMERICANS.* https://www.lawforseniors.org/topics/consumer-scams/305-tax-scams-that-target-millions-of-americans

[2] 2020. Exposing Voicemail Call-Back Scams. https://www.fcc.gov/news-events/blog/2019/08/28/exposing-voicemail-call-back-scams

[3] 2020. Investment Fraud Script. https://ag.ny.gov/sites/default/files/pdfs/bureaus/investor_protection/exhibit_k.pdf

[4] 2020. IRS Scam phone Transcript Tax Resolution Institute. https://www.taxresolutioninstitute.com/irs-scam-phone-transcript/

[5] 2020. This is what a Social Security scam sounds like. https://www.consumer.ftc.gov/blog/2018/12/what-social-security-scam-sounds?page=1

[6] Saeed Abu-Nimeh, Dario Nappa, Xinlei Wang, and Suku Nair. 2007. A comparison of machine learning techniques for phishing detection. In *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit*. 60–69.

[7] Kent Bach and Robert M Harnish. 1979. Linguistic communication and speech acts. (1979).

[8] Ram Basnet, Srinivas Mukkamala, and Andrew H Sung. 2008. Detection of phishing attacks: A machine learning approach. In *Soft Computing Applications in Industry*. Springer, 373–383.

[9] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175* (2018).

[10] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*.

[11] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*.

[12] Neil Chou, Robert Ledesma, Yuka Teraguchi, Dan Boneh, and John C. Mitchell. 2004. Client-Side Defense against Web-Based Identity Theft. In *Network and Distributed Systems Security Symposium (NDSS)*.

[13] A. Das, S. Baki, A. El Aassal, R. Verma, and A. Dunbar. 2020. SoK: A Comprehensive Reexamination of Phishing Research From the Security Perspective. *IEEE Communications Surveys Tutorials* 22, 1 (2020), 671–708.

[14] Robin Dreeke. 2013. *It's not all about "me", the top ten techniques for building quick rapport with anyone.* People Formula.

[15] S. Duman, K. Kalkan-Cakmakci, M. Egele, W. Robertson, and E. Kirda. 2016. EmailProfiler: Spearphishing Filtering with Header and Stylometric Features of Emails. In *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, Vol. 1.

[16] Gal Egozi and Rakesh Verma. 2018. Phishing email detection using robust nlp techniques. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 7–12.

[17] Ian Fette, Norman Sadeh, and Anthony Tomasic. 2007. Learning to Detect Phishing Emails. In *Proceedings of the 16th International Conference on World Wide Web*.

[18] ftc 2020. Federal Trade Commission, Scams. Federal Trade Commission. https://www.consumer.ftc.gov/features/scam-alerts

[19] ftc2 2016 (accessed June 11, 2020). Federal Trade Commission, Scams. Federal Trade Commission. https://www.consumer.ftc.gov/blog/2016/05/scammers-can-fake-caller-id-info

[20] Christopher Hadnagy. 2011. *Social Engineering The Art of Human Hacking*. Wiley Publishing Inc.

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

[22] Grant Ho, Aashish Sharma, Mobin Javed, Vern Paxson, and David Wagner. 2017. Detecting Credential Spearphishing in Enterprise Settings. In *26th USENIX Security Symposium (USENIX Security 17)*.

[23] Tom N. Jagatic, Nathaniel A. Johnson, Markus Jakobsson, and Filippo Menczer. 2007. Social phishing. *Commun. ACM* 50, 10 (2007), 94–100.

[24] Martin Kaste. 2019 (accessed June 11, 2020). Cybercrime Booms As Scammers Hack Human Nature To Steal Billions. National Public Radio. https://www.npr.org/2019/11/18/778894491/cybercrime-booms-as-scammers-hack-human-nature-to-steal-billions

[25] Allen Kim. [n.d.]. *A scam targeting Americans over the phone has resulted in millions of dollars lost to hackers. Don't be the next victim.* https://www.cnn.com/2019/10/27/business/phishing-bank-scam-trnd/index.html

[26] Merton Lansley, Francois Mouton, Stelios Kapetanakis, and Nikolaos Polatidis. 2020. SEADer++: social engineering attack detection in online environments using machine learning. *Journal of Information and Telecommunication* (2020).

[27] Yang Li and Tao Yang. 2018. *Word Embedding for Understanding Natural Language: A Survey*. Springer International Publishing.

[28] Brian MacWhinney and Johannes Wagner. 2010. Transcribing, searching and data sharing: The CLAN software and the TalkBank data repository. *Gesprachsforschung: Online-Zeitschrift zur verbalen Interaktion* 11 (2010), 154.

[29] Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. http://arxiv.org/abs/1301.3781

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

[31] Ying Pan and Xuhua Ding. 2006. Anomaly Based Web Phishing Page Detection. In *Computer Security Applications Conference, 2006. ACSAC '06. 22nd Annual*.

[32] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[33] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*.

[34] Venkatesh Ramanathan and Harry Wechsler. 2012. phishGILLNET—phishing detection methodology using probabilistic latent semantic analysis, AdaBoost, and co-training. *EURASIP Journal on Information Security* (2012).

[35] H. Sandouka, A. J. Cullen, and I. Mann. 2009. Social Engineering Detection Using Neural Networks. In *2009 International Conference on CyberWorlds*. 273–278.

[36] Gianluca Stringhini and Olivier Thonnard. 2015. That Ain't You: Blocking Spearphishing Through Behavioral Modelling. In *Proceedings of the 12th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment - Volume 9148 (DIMVA 2015)*.

[37] Verizon. 2019. 2019 Data Breach Investigations Report. https://enterprise.verizon.com/resources/reports/dbir/.

[38] Rakesh Verma and Nabil Hossain. 2013. Semantic feature selection for text with application to phishing email detection. In *International Conference on Information Security and Cryptology*. Springer, 455–468.

[39] Rakesh Verma and Nirmala Rai. 2015. Phish-IDetector: Message-ID based automatic phishing detection. In *2015 12th International Joint Conference on e-Business and Telecommunications (ICETE)*, Vol. 4. IEEE, 427–434.

[40] Rakesh Verma, Narasimha Shashidhar, and Nabil Hossain. 2012. Detecting Phishing Emails the Natural Language Way. In *Computer Security – ESORICS 2012*, Sara Foresti, Moti Yung, and Fabio Martinelli (Eds.).

[41] Patrick Webre. 2019. Exposing Voicemail Call-Back Scams. Federal Communications Commission Blog. https://www.fcc.gov/news-events/blog/2019/08/28/exposing-voicemail-call-back-scams

[42] Patrick Webre. 2020. UPDATED FRAUD ADVISORY (March 2020). U.S. Marshals Service. https://www.usmarshals.gov/news/chron/2019/scam-alerts.htm

[43] Yue Zhang, Jason I. Hong, and Lorrie F. Cranor. 2007. Cantina: A Content-based Approach to Detecting Phishing Web Sites. In *Proceedings of the 16th International Conference on World Wide Web*.