2016 IEEE 40th Annual Computer Software and Applications Conference

EmailProfiler: Spearphishing Filtering with Header and Stylometric Features of Emails

Sevtap Duman*, Kubra Kalkan Cakmakci†, Manuel Egele‡, William Robertson* and Engin Kirda*
*College of Computer and Information Science
Northeastern University, Boston, MA, USA
Email: {sevtap,wkr,ek}@ccs.neu.edu
†Department of Engineering
Bogazici University, Istanbul, Turkey
Email: kubrakalkan@brown.edu
‡Electrical and Computer Engineering
Boston University, Boston, MA, USA
Email: megele@bu.edu

***Abstract*—Spearphishing is a prominent targeted attack vector in today's Internet. By impersonating trusted email senders through carefully crafted messages and spoofed metadata, adversaries can trick victims into launching attachments containing malicious code or into clicking on malicious links that grant attackers a foothold into otherwise well-protected networks. Spearphishing is effective because it is fundamentally difficult for users to distinguish legitimate emails from spearphishing emails without additional defensive mechanisms. However, such mechanisms, such as cryptographic signatures, have found limited use in practice due to their perceived difficulty of use for normal users.**

In this paper, we present a novel *automated* approach to defending users against spearphishing attacks. The approach first builds probabilistic models of both email metadata and stylometric features of email content. Then, subsequent emails are compared to these models to detect characteristic indicators of spearphishing attacks. Several instantiations of this approach are possible, including performing model learning and evaluation solely on the receiving side, or senders publishing models that can be checked remotely by the receiver. Our evaluation of a real data set drawn from 20 email users demonstrates that the approach effectively discriminates spearphishing attacks from legitimate email while providing significant ease-of-use benefits over traditional defenses.

I. Introduction

The goal of a social engineering attack is to trick the victim into performing an action in favor of the attacker but to the detriment of the victim. Commonly, this action consists of executing a piece of malicious software (malware) of the attacker's choosing. Spam and phishing messages frequently contain executable email attachments or links to malware that the victim should run. However, technical advances in spam and phishing protection significantly reduce the reach and success of large-scale attacks. To increase success rates, especially with high-value targets, adversaries have switched their strategies towards so-called *spearphishing* attacks.

In a spearphishing attack, the adversary leverages publicly available information about his victim to craft email messages that are custom-tailored to the victim and thus appear legitimate. This characteristic of custom-tailoring sets spearphishing attacks apart from regular phishing attacks, which commonly can be easily identified due to poor grammar and other obvious tell-tales [1]. Whaling attacks are specific variants of spearphishing where the attacker poses as his victim's superior. Spearphishing and whaling attacks have advanced to a major attack vector for advanced persistent threats (APTs), as usable technical defenses that allow potential victims to identify such attacks are scarce and limited in applicability.

Existing systems, such as IdentityMailer [2], identify spearphishing attacks before the offending email message reaches its intended victim. This is achieved by monitoring emails *sent* by legitimate, yet compromised, accounts within the same organizational entity (e.g., company). To this end, IdentityMailer observes the sending behavior of email accounts with respect to writing habits (e.g., character distributions), composition habits (e.g., the time when emails are commonly sent), and interaction habits (e.g., the list of recipients commonly included in outgoing email messages). Because IdentityMailer targets a single organizational unit, it can capture all these characteristics conveniently at a company's SMTP server.

Unfortunately, this reliance on observing all email traffic originating from a given email account implies that such defenses can only be effective within a given company. Recipients at other companies or regular email users cannot be protected by systems such as IdentityMailer. Thus, while potentially very useful, IdentityMailer and similar approaches only afford full protection if they are comprehensively deployed. As benefits from partial deployments are limited, incentives for initial deployments are limited too. This circular dependency implies that a comprehensive deployment of techniques such as IdentityMailer seems unlikely.

Instead of relying exclusively on sender-based deployment and cooperation, we present our approach – EMAILPROFILER– which identifies potential spearphishing attacks on the email recipient's side of the communication. Such a deployment has the immediate advantage of protecting the user from spearphishing attacks even in the absence of other cooperating entities in the ecosystem. Thus, a gradual deployment will benefit users as soon as they adopt such a protection scheme. Furthermore, EMAILPROFILER also supports cooperating email senders which allows for increased detection accuracy.

EMAILPROFILER identifies spearphishing attacks on the recipient's side by building a behavioral profile for each email sender. To this end, EMAILPROFILER extracts a set of 23 features from each email. While previous work relied heavily on meta-information accessible from email headers, EMAILPROFILER also leverages 199 stylometric features inspired by the field of natural language processing. The observed feature values are then aggregated by the sender into a behavioral profile that characterizes the behavior of the corresponding author. Incoming email messages undergo the same feature extraction process, but instead of integrating the feature values with the behavioral profile, the values are checked for consistency with the existing profile. If the difference of the newly extracted features with the behavioral profile is above the detection threshold, EMAILPROFILER will flag the incoming message as a potential spearphishing email.

Of course, bootstrapping is a significant challenge for systems such as EMAILPROFILER. To ameliorate this problem, EMAILPROFILER provides two mechanisms that ease the bootstrapping problem. First, users who wish to adopt EMAILPROFILER can train behavioral profiles based on previous correspondence with a given author. A second mechanism that EMAILPROFILER supports is the sharing of behavioral profiles via a privacy-preserving trusted entity. This approach allows EMAILPROFILER to query behavioral profiles of participating email users. This enables EMAILPROFILER to precisely classify incoming messages from authors with whom the receiving user did not have previous correspondence.

To summarize, this work features the following contributions:

- We propose and demonstrate that behavioral features drawn from email header information and stylometric properties of the email body text allow for a precise characterization of the email's author.
- We leverage this insight to implement EMAILPROFILER, an anti-spearphishing technique that accurately identifies spearphishing emails on the email-recipient's side of the communication.
- We design a privacy-preserving trusted infrastructure that allows users who opt into using this component to aid in overcoming the bootstrapping challenge that EMAILPROFILER and other approaches face.
- We evaluate EMAILPROFILER based on a real-world contemporary dataset obtained from participating volunteers at our research institution. During this evaluation EMAILPROFILER verifies authors with accuracy rates between 67% and 100%.

The remainder of this paper is structured as follows. We place our approach in the context of related work in Section II before describing EMAILPROFILER in Section III. We then describe a communication protocol for centralized privacy-preserving profile evaluation in Section IV. Our evaluation is presented in Section V, and Section VI concludes the paper.

II. RELATED WORK

EMAILPROFILER makes use of email metadata and stylometric features of email content in order to identify spearphishing emails. In the following, we discuss significant prior work in authorship detection using stylometric features, natural language processing, and spearphishing detection.

A. Stylometric Authorship Identification

Spearphishing and whaling emails use identity hiding and impersonation attacks. Since stylometry assists revealing the authorship it is reasonable to use stylometry while solving the identity problem of an email.

One of the main studies on authorship identification is Abbasi's work Writeprints [3]. In Writeprints, the authors used online texts from different sources such as Enron emails, eBay comments, Java forums, and CyberWatch chats as their datasets. They extracted a comprehensive feature set and experimented with their technique on the datasets for identification and similarity detections. They showed that when the number of authors in a dataset increases, the accuracy of the Support Vector Machine (SVM) for authorship identification decreases.

Afroz has applied stylometry to the problem of authorship identification in the area of cybercrime in their works. In one approach, they studied obfuscated writing where an author imitates someone else [4]. In another study, they studied doppleganger accounts in underground forums that included English, German and Russian texts [5]. They found that a hybrid method that combines stylometry with forum specific features were more successful in finding doppelgangers.

McDonald tested whether their author identification framework, called Anonymouth [6], could handle manually anonymized text. Where Anonymouth focuses on privacy of the author, Narayanan studied author identification using a large set of forum posts [7]. They showed a correlation between the number of features and the accuracy of different classification methods. Another common point of all these authorship identification works is that they constructed their corpus based on word counts greater than 500 or character counts greater than 7500. Ledger used single character frequencies to identify the author of acts in Two Noble Kinsmen between Shakespeare and Fletcher [8]. They stated that text samples of 500 words or fewer will not allow for accurate authorship identification.

Email authorship problems are investigated as well. Given the shortness of email documents, de Vel limited the topics in the emails to movies, food and travel and classified authors based on the structural characteristics and linguistic patterns of emails [9]. Similarly, in a feasibility study [10] raw keystrokes and emails have been used for author identification with a limited feature set. Nizamani used cluster based classification to identify authors in emails [11].

In a similar manner to Afroz's work, Iqbal studied the feasibility of forensic investigation of cyber crimes using stylometric features of emails [12]. In their study, they used most of the stylometric features listed in Writeprints [3] on the Enron email dataset. We used a different feature set in our work that includes a subset of Writeprints' stylometric features and header information features. Aside from the feature set, the main difference between their study and ours is the data used for testing.

Another email forgery detection method is Lin's work that uses both stylometry and geolocation during email analysis [13]. Their client-side plugin is tested using the Enron corpus. Brocardo also studied authorship verification [14], and proposed a method that uses n-gram analysis to verify the author of short messages.

One of the most recent papers on authorship verification of emails is IdentityMailer [15]. IdentityMailer uses header information of an email as well as stylometric features. More specifically, they extracted reply, forwarded, time of day, day of week, and recipient information from the metadata of the email to verify email authorship. The main difference between our work and IdentityMailer is the deployment model. IdentityMailer is designed to prevent transmission of spearphishing emails from compromised machines before they are forwarded to SMTP servers. In contrast, our profilers check for spearphishing emails after they are received by the user.

B. Natural Language Processing

Utilizing correct classifying methods while using Natural Language Processing (NLP) plays a key role. Classify, But Verify [16] examines this issue. Other NLP work includes NoCrack, which used Natural Language Encoders to crack passwords [17]. Crowston used NLP techniques to acquire qualitative data [18]. Recently, Palka and McKoy [19] combined NLP analysis with email filters. In that work, they tested the effectiveness of email filters using a fuzzing strategy based on n-gram analysis.

C. Spam and Phishing Email Filtering

Spam and phishing email filtering is pervasively deployed on today's Internet. Spam filters typically use both linguistic and sender reputation features for detection. Moreover, email clients incorporate feedback from users who mark suspicious emails as spam. Spam filter performance and efficiency has been studied and enhanced by many researchers [20], [21]. Over time, the term spam has come to include those that include links that direct users to unwanted, unsolicited domains as well. While studying spam links, Thomas compared the features of email spam with the features of tweet spam [22] and found that email spammers do not overlap with tweet spammers. Some of email spammers bypass filters by embedding spam into images. In his work [23], Wang describes a solution to image-based spam by clustering a new corpus of spam images and comparing them with non-spam image-based emails.

In the same way as spam filtering, linguistic features of an email are important for detecting phishing emails. Some characteristics of phishing emails are typos in the body, lack of knowledge of the receiver's name, asking for a monetary deposit, or asking users to click on a link. Aggarwal used NLP techniques to detect phishing emails [24] using these characteristics. Ramanathan employs multi-stage filtering on phishing emails, which combines NLP techniques with machine learning [25]. More recent work by Lottre [26] proposes a framework that marks emails as safe or non-safe; the aim of the framework is to educate users and increase their security awareness.

Spearphishing attacks are considered a subset of phishing email attacks; however, popular preventative methods for spam and phishing attacks are not well-suited to detecting spearphishing attacks. The main reason for this is that spearphishing emails are crafted to impersonate someone that the recipient would trust, and are therefore more likely to bypass mass-market spam filters.

Our work uses some of the methods described above for spam and phishing email detection, such as NLP and author identification. However, EMAILPROFILER uses an extended feature set and different classification methods. For example, the linguistic features EMAILPROFILER extracts are classified per person instead of into global safe and unsafe categories. Another distinguishing feature of our work is that we use header information to understand the writing habits of email authors.

III. EMAILPROFILER DESIGN

The main goal of EMAILPROFILER is to identify whether a received email originates from the author claimed in the email's metadata. To this end, EMAILPROFILER creates behavioral profiles for each sender and compares new incoming emails against these profiles. The behavioral profiles themselves consist of a combination of syntactic and stylometric features (§III-A). EMAILPROFILER supports two modes of operation: First, evaluating incoming emails based on receiver-trained profiles; second, generating profiles at the sender and making the profile available for querying at a trusted server. The feature sets used for these two variants slightly differ, as received emails contain additional information (e.g., header information pertaining to intermediary email servers) over emails that are about to be sent.

A. Feature Set and Extraction

Natural language processing techniques have identified a lower-bound threshold of 500 words to precisely identify the author of a given text [8]. Unfortunately, email messages frequently consist of less than 500 words. Fortunately, email messages provide ample opportunity beyond the email text itself to draw identifying information from. In particular, EMAILPROFILER leverages the structured information contained in email headers in combination with the information in the email's body text.

EMAILPROFILER analyzes the email body for three categories of features: *lexical*, *syntactic*, and *structural* features [3].

a) Lexical features: Lexical features comprise the total number of words, characters per word, characters in the whole text, characters per line, lines, sentences, single character frequencies in the text, and character frequencies that are used to end a sentence.

b) Syntactic features: Syntactic features include part-of-speech tags as defined and supported by the Stanford CoreNLP system [27]. These features include, for example, the number of adjectives, adverbs, coordinating conjunctions, and past participle verbs.

c) Structural features: Structural features consist of personal identifiers of the author, such as their signature, signature extras (contact information such as address, phone number, etc.), farewell, greeting, and sentence beginning and ending types. There are two features that consider the beginnings of sentences. The first feature is the number of sentences that start with an uppercase letter. The second feature gives the number of sentences that start with a lowercase letter. There are four features that characterize the sentence ending habits of an author. The first value corresponds to the number of sentences where an author uses spaces to delimit sentences. The second value indicates the number of sentences that the author terminates with a dot. The third value counts the number of sentences where the author uses spaces after ending a sentence. The fourth and final feature counts the number of sentence-ending punctuation characters except dots.

EMAILPROFILER also leverages information contained in the email headers. More precisely, the following header fields have corresponding features in the behavioral profiles: `return-path`, `x-mailman-version`, `x-originating-hostname`, `x-originating-ip`, `x-spam-flag`, `x-virus-scanned`, and `carbon copy`. Most of these features are defined only for received emails. For example, a `x-spam-flag` header is commonly injected by a mail server acting as a spam filter. Such filtering capabilities are most frequently employed at the receiving mail server and, thus, an outgoing email will lack such header information. EMAILPROFILER draws additional information from the timestamps recorded in the email headers. However, because the precise time is likely not significant to identify an author, we broadly classified the sending time into four categories: midnight, morning, afternoon, or night.

In total, the behavioral profiles extracted by EMAILPROFILER consist of 222 features. Most features values are captured as the normalized number of occurrences. However, features that contain words or phrases are not adequately captured by this representation. Instead, the feature value for such features is a binary value with 0 indicating that the corresponding feature has not been observed previously for a given author and 1 representing the case that the same value was already observed for the author.

B. Training Profilers

In the previous section we described the features EMAILPROFILER uses to characterize individual emails. Here we describe how the individual features are aggregated into a behavioral profile of the author of an email message. As stated previously, EMAILPROFILER has two different modes of extracting behavioral profiles. The *inbox profiler* will generate behavioral profiles based on the email messages previously received by the user of EMAILPROFILER. The *sentbox profiler*, however, is an optional component that users of EMAILPROFILER can use to generate behavioral profiles of their own email sending behavior which can then be shared on a trusted server. Behavioral profiles shared in that way can be leveraged by EMAILPROFILER to evaluate the veracity of incoming emails for senders with whom the recipient did not have sufficient prior communications to exercise the *inbox profiler*.

If the author of the email is a frequent contact of the receiver and has sent a sufficient number ($\geq S$)[1] of emails to the receiver, that author is labeled a *recognized* author. A recognized author sent sufficiently many email samples to be identified and distinguished from other authors in the receiver's inbox. An *unrecognized* author has not previously sent any emails to the receiver or the number of the emails is not enough to create a training set to identify the author. To assess whether an email claiming to originate from a recognized author is likely spearphishing, the user will compare the incoming message against the profile generated by the inbox profiler. Emails from unrecognized authors can be checked with the help of a trusted server that contains profiles generated from the sentbox profiler.

1) Inbox Profiler: For a recognized author, the receiver can compare an incoming email against the behavioral profile extracted for the sender. To establish this profile, EMAILPROFILER first aggregates all emails sent by the same sender. Subsequently, the features (as discussed in Section III-A) are extracted, and finally these features are used to train a classifier through a support vector machine (SVM). The classification task at hand is to detect whether an email is sent by the author who is listed as the sender. As the SVM performs best if it is provided with roughly equal amounts of positively and negatively labeled samples, we augment the list of emails sent by the same author (positive label) with the equal number of emails drawn at random from other authors (negative label). EMAILPROFILER then uses the SVM to determine the weights that indicate how characteristic each feature is for the style of the recognized author for whom the behavioral profile is built. These weights will then be used as coefficients in the decision function that separates legitimate from potential spearphishing emails.

2) Sentbox Profiler: As EMAILPROFILER does not generate behavioral profiles for unrecognized authors, it cannot assess the veracity of the sender's identity with the same mechanism

[1]S is a tunable parameter of EMAILPROFILER, and in our experiments presented in Section V we empirically determined that a value of $S = 50$ performs well.

for emails originating from unrecognized authors. Instead, Section IV illustrates how a recipient can query a trusted server to check the features extracted from a received email against a behavioral profile of the sender. Of course, this method is only applicable if the sender chose to make her behavioral profile available to the server.

An email sender who chooses to make her profile available to the trusted server will first run the *sentbox profiler*. This component operates quite similarly to the *inbox profiler*. However, instead of generating a behavioral profile for all senders in a recipient's inbox, the sentbox profiler generates a single profile for the cooperating user. To this end, EMAILPROFILER aggregates all sent emails of the user and extracts the features as described above. Subsequently, the same SVM classifier is trained based on the observed feature vectors and the resulting feature weights are combined into the behavioral profile of the user. In a final step, this behavioral profile is shared with the server to ease the challenge of bootstrapping.

We should note that the use of the sentbox profiler and the trusted server is only one way of overcoming the bootstrapping challenge. As the inbox profiler can start generating behavioral profiles based on only S messages from a given sender, the value of S directly affects the window of opportunity for potential attackers. That is, a spearphishing attack only circumvents EMAILPROFILER if it is one of the first S messages from a given sender. Attackers commonly try to leverage existing trust relationships, and spoofing sender email addresses and identities is a convenient way to misuse this trust. Thus, confining attackers to sending their spearphishing messages within the first S messages significantly reduces the exposure of potential victims to such attacks.

IV. SENTBOX MODEL

As described in the previous section, one deployment model supported by EMAILPROFILER is for users to publish authorship profiles trained on their own sentbox to a trusted third party. This model is primarily intended to address situations where recipients do not have enough emails from a given author to train an authorship profile on the receiving side (i.e., unrecognized users). In the following, we elaborate upon the details of this model.

The sentbox model is comprised of three steps:

1) Users train and upload their sentbox profile
2) Unrecognized user sends email
3) Receiver validates authorship against the sender's sentbox profile

In the first step, users who participate in this model train an authorship profile using their own sentbox. This profile is then uploaded to a trusted third party server, as shown in Figure 1. In the second step, a user receives an email from an unrecognized user – i.e., a user for which no receiving side profile exists or can be built due to insufficient training examples (Figure 2). This triggers the third step: validation of authorship for the received email by querying the sender's sentbox profile at the trusted third party, shown in Figure 3.

Notation	Definition
US	Public key of the server
E_{US}	Encryption of a message with the server's public key
Add_A	Email address of user A
Cer_A	Certificate of user A
N	Nonce
TS	Timestamp
RS	Private key of the server
D_{RS}	Decryption of the message with the server's private key
UA	Public key of user A
E_{UC}	Encryption of a message with user A's public key
S	Secret that will be used to generate a session key
$f(N)$	A function that takes a nonce N and returns a value N' that is based on N
N'	Result of $f(N)$
RA	Private key of user A
D_{RA}	Decryption of a message with user A's private key
$g(S)$	A function that takes secret S and returns another value that will be used as session key KS
KS	Session key
E_{KS}	Encryption of a message with a session key
$Prof_F$	Profile of user F
V_F	Feature vector of user F's email
D_{KS}	Decryption of a message with the session key
Res	Accept or reject result from a profile comparison

Table I
NOTATION AND DEFINITIONS.

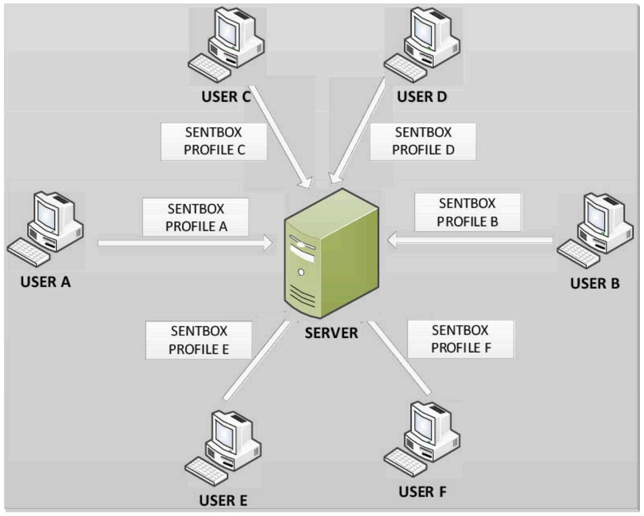

Figure 1. Users upload trained sentbox profiles to the trusted server.

In order to securely communicate with the trusted third party, a communication protocol was devised for the first and third steps. A summary of the notation and corresponding definitions is shown in Table I.

User A starts the validation protocol with the first message. This is encrypted with the server's public key US. The message contains the email address of user A Add_A, a certificate Cer_A belonging to A that contains the public key of A UA, a nonce N, and a timestamp TS. N is used to prevent replay attacks, whereas TS is used for freshness. As soon as the server receives Msg_1, it decrypts it as $D_{RS}\{Msg_1\}$ with its own private key RS. The server checks if this message is

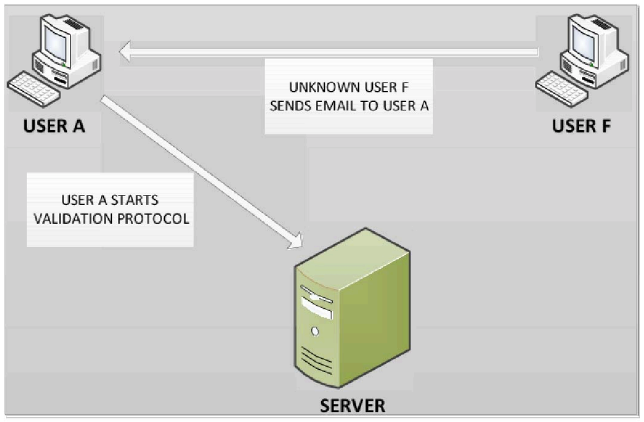
UNKNOWN USER F SENDS EMAIL TO USER A
USER A
USER F
USER A STARTS VALIDATION PROTOCOL
SERVER

Figure 2. Unknown user F sends email to user A, triggering the validation protocol.

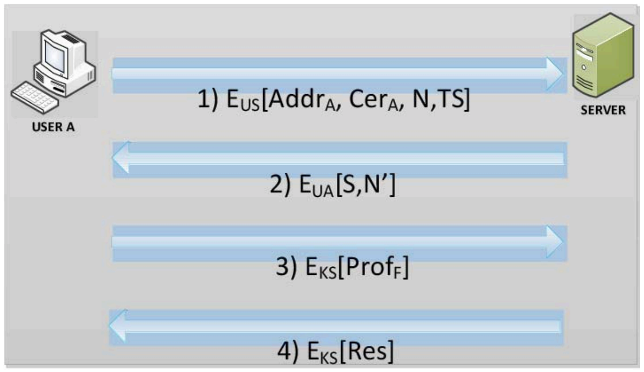

Figure 3. Validation protocol.

fresh using its timestamp. Then, the server generates the value $N' = f(N)$, where $f(N)$ is a shared function that is known by the server and user A. Nonce N' will be used to prove to user A that the next message actually originates from the server, because N can only be learned by decrypting the message with the server's private key. Therefore, N' can only be generated by the server and user A. The server also generates secret S which is used to generate a session key KS. KS is produced by function $g(S)$ which is a pre-agreed function.

In Msg_2, the server sends S and N' by encrypting with user A's public key UA. When user A receives Msg_2, he first decrypts it as $D_{RA}\{Msg_2\}$ with his own private key RA. Then, user A checks whether N' agrees with his local computation of $f(N)$. If they are equal, he continues the protocol, otherwise he terminates it. Afterwards, he computes $KS = g(S)$ in order to decrypt the following messages. He sends Msg_3, including a feature vector of user F's email V_F by encrypting with KS. When the server receives this message, it decrypts it by using KS and compares it with the sentbox profile of user F that was uploaded in step 1. If the query returns positive, it sends a validation message as a result, otherwise it sends a reject message. The resulting message Msg_4 that contains Res is encrypted with KS. When user A receives it, he decrypts it using KS and learns the result.

When users upload their sentbox profiles, a similar validation protocol is applied on uploading sentbox profile procedure with two small differences in Msg_3 and Msg_4. $Prof_F$ in Msg_3 is the sentbox profile of user F in the upload procedure, whereas in the validation protocol it is the feature vector V_F of an email received from user F. Also, in Msg_4, Res contains an accept or reject decision in the validation protocol, whereas in the upload procedure it contains a confirmation that the server has received the complete profile of user F.

V. Evaluation

In this section, we evaluate the efficacy of EMAILPROFILER in generating behavioral profiles that capture user behavior. Furthermore, we evaluate how well these profiles characterize the email writing and sending behavior of 20 volunteer users at our academic institution. Besides this empirical evaluation, we also analyze the robustness of the generated behavioral profiles from a theoretical standpoint against privacy attacks. Finally, we also discuss assumptions and limitations that underlie the current prototype implementation of EMAILPROFILER.

A. User Data Collection

Previous studies on email security and spam frequently used publicly accessible datasets such as Enron and Symantec for their evaluation. However, spearphishing is a precisely targeted and relatively new phenomenon that we believe is insufficiently reflected in these datasets. Thus, we chose to evaluate EMAILPROFILER on a contemporary dataset of emails provided by volunteers at our academic institution. To this end, we were able to solicit the help of 20 connected users on our campus. All the participants are PhD students and professors in the field of computer science and bioinformatics.

Extrapolating from our own mindset, we assumed that none of the participants would want to give us direct access to their email archives. Thus, we developed the profilers as stand-alone components that each volunteer could easily run for themselves. Collecting the data to evaluate EMAILPROFILER in this way had two significant advantages. First, users could inspect the data transmitted back for the evaluation and be convinced that no privacy sensitive information was contained in the transmitted results. Second, this forced us to engineer the profilers robustly to make them integrate smoothly with the email habits of the participating users.

To avoid privacy leaks in the output, we replaced email addresses with hashed versions thereof. Furthermore, all the participants were given explanations on how to use the profilers and what information is contained in the output data. We specifically asked the participants to run the profilers on their institution-associated email data. This allowed us to evaluate the functionality of the trusted server that contains the profiles extracted by the sentbox profiler. In our experiments, we set the threshold S at a value of 50. Furthermore, volunteers were selected in a way that each person had at least 50 emails from another person that participated in the experiment. As a result, each participating user has at least one recognized user associated with the user's profile in their inbox.

We acquired both inbox and sentbox generated profiles from participants. This way even if a user did not have enough sample emails from a participant but had at least one email, we were able to compare that received email with the sender's own profile which was obtained from the sender. This gave us the opportunity to design multiple testson the collected data.

A detailed breakdown of the data our volunteers processed with the profilers is presented in Table II.

We also computed statistics on the processed messages. For example, many email messages did not contain any or only very short text in their body. This observation is a clear indicator that existing stylistic-only author attribution schemes would likely result in poor accuracy. However, as real-world use of email features such characteristics prominently, a defense system such as EMAILPROFILER must be able to handle such email messages too. The excellent accuracy established in our evaluation (see Section V-B) thus demonstrates that drawing information from the structured email header is beneficial to perform author re-identification based on email messages.

UserID	# of Distinct Authors in Inbox	# of emails in inbox	# of emails in sentbox
U1	216	1,434	722
U2	41	670	8,915
U3	107	631	629
U4	320	1,168	4,188
U5	156	852	18
U6	84	433	220
U7	343	1,459	282
U8	235	3,127	1,416
U9	745	4,290	1,272
U10	246	3,032	509
U11	892	7,955	2,392
U12	759	6,346	2,395
U13	965	6,493	2,005
U14	311	2,607	709
U15	418	1,603	402
U16	526	3,423	488
U17	1,244	17,303	2,948
U18	846	11,450	2,274
U19	698	8,715	13,791
U20	1,279	16,440	1,094

Table II
DATASET STATISTICS.

B. Experiments

Based on the feature vectors extracted by the profilers, we trained a one-class SVM classifier as discussed in Section III.

To evaluate the accuracy of the behavioral profiles, we performed 10-fold cross validation for the profiles generated for each recognized author. To this end, we randomly separated the email messages from each recognized author into ten equal-sized non-overlapping subsets. We then used nine subsets or 90% of the messages for training and the remaining 10% for testing. This was repeated 10 times with a different subset used for testing in each iteration. In total, we performed this profile generation 215 times. The worst-performing profile resulted in 67% accuracy, whereas the best performing profile reached 100% accuracy. Averaging the accuracy of all 215 profiles results in a 93% accuracy value for EMAILPROFILER.

As the sentbox profiler generates one profile per participating user, we obtained 20 profiles using this method. Similar to the inbox testing case, each generated profile was tested using 10-fold cross validation.

Additionally, they have been put in tests where each user's whole profile data vs all other authors are used for training and tested against a profile generated with sent profile through using inbox data of a participant. This way, we created the case of a user receiving an email from a unrecognized user (Section IV). This case was repeated for each participant vs the rest of the participants. 10-fold cross validation results are between 80% and 100%, in 20 tests. Comparison with inbox data of another participant give the results in between 75% – 100%.

C. Theoretical Analysis

In the following, we perform a theoretical analysis of the robustness of the sentbox model described in Section IV against various attacks.

1) Denial of Service Attack: In our proposed architecture, the trusted server presents a single point of failure. While redundant operation and over-provisioning can reduce the risk of accidental failure, attackers might strive to launch denial of service attacks against this part of the infrastructure [2].

In order to prevent accidental brute force attacks, there are some common use cases that need to be considered. An unknown email sender, such as user F in Section IV, can send a collective email to multiple receivers. For instance, a professor can send an announcement to all university departments. In this case, he would be an unrecognized user for most of the recipients. The features of this email would be submitted as queries to the server multiple times in short succession. However, this simple case of an unintentional DoS can be mitigated with a caching layer at the server. That is, since the email is identical, the feature vector sent to the server during the validation protocol will be identical and, therefore, the validation result can be cached so long as the user profile itself remains valid.

2) Profile Reversing: If the number of times that a given profile is queried is not limited, a targeted profile reversing attack can be possible. A distributed attacker can query the server from a large set of disparate addresses with different feature vectors for the profile of a targeted user. The attacker can exhaustively search the feature space and thus reverse-engineer the target's profile.

In order to prevent this attack, a request limit L which de-

[2]Note that even if the attacker succeeds in this DoS attack, he is still limited to sending his spearphishing email as one of the first S messages. The client-side inbox profiler builds and maintains profiles for recognized authors independently of the server.

termines the maximum number of queries for an unrecognized user can be asked is defined. This limit is set to a reasonable default value (e.g., 1 query per second) and can be adapted to the email behavior of the user. For instance, for the above-mentioned case of the announcement email, a higher threshold might be advisable. While it is unlikely that a legitimate user consistently sends one email per second, such a limit would effectively prevent the exhaustive search of the feature space as we illustrate in Section V-C3.

3) Analysis of Regular and Smart Adversaries: In this section, regular and smart attacker cases are considered. The regular attacker is one who knows the features that are used in this model. He can only deduce the parameter types by considering the features. According to these types he determines the number of bits for each feature and applies a brute force profile reversing attack. All of these features, parameter types, and the parameter space in bits are illustrated in Table III.

Assuming that all the features are independent, in order to find the number of trials all values in the third column of the table are multiplied. According to this result, a regular attacker can expect to uncover the user's profile in approximately $2^{12,140}$ trials. If the attacker spoofs IP addresses and attacks in a distributed manner, then the number of years Y required to search the feature space will be

$$Y = \frac{2^{12,140}}{365L}$$

where L is the maximum allowable number of times an unrecognized user can be queried from the server. Due to the huge feature-value space, the linear dependency on the number of spoofed IP addresses and the query limit set at the server makes this unstructured attack an intractable challenge.

A smart attacker is one who knows the most popular range of values for each feature. He does not try all possibilities, but instead will only try the most probable cases which are deduced from empirical results. By multiplying all values in the fourth column of the table which represent conservative estimates of the ranges of likely values for each feature, the total number of trials needed to obtain a profile of a user is obtained. If the attacker spoofs IP addresses and attacks in a distributed manner, then the number of years to break the system will be

$$Y = \frac{9.673 \times 10^{120}}{365L} \approx \frac{2^{401}}{365L}.$$

While this is a significant reduction in difficult from a naïve brute-force feature space enumeration, given $L = 100$ it would nevertheless take an insurmountable number of years to acquire the correct profile.

VI. Conclusion

In this paper, we proposed EmailProfiler, a new approach to detecting spearphishing attacks using both features extracted from both email metadata and stylometric information. We evaluated this approach using an dataset comprising email from 20 volunteers at our academic institution. The results demonstrate that EmailProfiler's techniques can reach an average 98% accuracy when verifying email authorship through inbox and sentbox profiling. We also show that email profiles are robust to reverse-engineering in both a theoretic and empirical analysis. Finally, in contrast to prior work, EmailProfiler supports gradual adoption by users, leading to increased utility as a real-world, usable defense against spearphishing attacks.

Features	Th. Range	# Trials for Th. Analysis	Emp. Range	# Trials for Emp. Analysis
Email address	String	2^{32}	$[0, 2^{32}]$	2^{32}
Time of day	$[0, 3]$	4	$[0, 3]$	4
Letters in subject	Integer	2^{32}	$[0, 176]$	176
Caps in subject	Integer	2^{32}	$[0, 103]$	103
Words in subject	Integer	2^{32}	$[0, 29]$	29
Chars/word in subject	Double	2^{64}	$[0, 75]$	75
Total words	Integer	2^{32}	$[0, 23675]$	23675
Chars/word	Double	2^{64}	$[1, 217]$	216
Chars	Integer	2^{32}	$[0, 200259]$	200259
Line count	Integer	2^{32}	$[0, 8521]$	8521
Chars/line	Double	2^{64}	$[0, 40088]$	40088
Sentences	Integer	2^{32}	$[1, 4457]$	4456
Caps sentence start	Integer	2^{32}	$[0, 987]$	987
Small sentence start	Integer	2^{32}	$[0, 4433]$	4433
Sentence end spaces	Integer	2^{32}	$[0, 2701]$	2701
Sentence end dot	Integer	2^{32}	$[0, 431]$	431
Sentence end w/o space	Integer	2^{32}	$[0, 4256]$	4256
Sentence end punctuation	Integer	2^{32}	$[0, 4208]$	4208
Chars	Double	2^{64}	$[0, 1]$	$1024 * 2^{52}$
Non-chars	Double	2^{64}	$[0, 1]$	$1024 * 2^{52}$
POS	Double	2^{64}	$[0, 1]$	$1024 * 2^{52}$
BCC	Bool	2	$[0, 1]$	2
CC	Bool	2	$[0, 1]$	2
Farewell	Bool	2	$[0, 1]$	2
Greeting	Bool	2	$[0, 1]$	2
MIME version	Bool	2	$[0, 1]$	2
Sender	Bool	2	$[0, 1]$	2
Signature	Bool	2	$[0, 1]$	2
Extended signature	Bool	2	$[0, 1]$	2
X-Mailer	Bool	2	$[0, 1]$	2
X-Originating-IP	Bool	2	$[0, 1]$	2

Table III
Feature list.

Acknowledgment

This work was supported by the National Science Foundation (NSF) under grant CNS-1409738, and Secure Business Austria.

References

[1] C. Herley, "Why do nigerian scammers say they are from nigeria?" in *WEIS*, 2012. [Online]. Available: http://research.microsoft.com/apps/pubs/default.aspx?id=167719

[2] G. Stringhini and O. Thonnard, "That aint you: Blocking spearphishing through behavioral modelling," in *Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer International Publishing, 2015, pp. 78–97.

[3] A. Abbasi and H. Chen, "Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace," *ACM Trans. Inf. Syst.*, vol. 26, no. 2, pp. 7:1–7:29, Apr. 2008. [Online]. Available: http://doi.acm.org/10.1145/1344411.1344413

[4] S. Afroz, M. Brennan, and R. Greenstadt, "Detecting hoaxes, frauds, and deception in writing style online," in *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, ser. SP '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 461–475. [Online]. Available: http://dx.doi.org/10.1109/SP.2012.34

[5] S. Afroz, A. C. Islam, A. Stolerman, R. Greenstadt, and D. McCoy, "Doppelgänger finder: Taking stylometry to the underground," in *Proceedings of the 2014 IEEE Symposium on Security and Privacy*, ser. SP '14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 212–226. [Online]. Available: http://dx.doi.org/10.1109/SP.2014.21

[6] A. W. E. McDonald, S. Afroz, A. Caliskan, A. Stolerman, and R. Greenstadt, "Use fewer instances of the letter "i": Toward writing style anonymization," in *Proceedings of the 12th International Conference on Privacy Enhancing Technologies*, ser. PETS'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 299–318. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-31680-7_16

[7] A. Narayanan, H. Paskov, N. Z. Gong, J. Bethencourt, E. Stefanov, E. C. R. Shin, and D. Song, "On the feasibility of internet-scale Author identification," in *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, ser. SP '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 300–314. [Online]. Available: http://dx.doi.org/10.1109/SP.2012.46

[8] G. Ledger and T. Merriam, "Shakespeare, fletcher, and the two noble kinsmen," *Literary and Linguistic Computing*, vol. 9, no. 3, pp. 235–248, 1994. [Online]. Available: http://llc.oxfordjournals.org/content/9/3/235.abstract

[9] O. de Vel, A. Anderson, M. Corney, and G. Mohay, "Mining e-mail content for Author identification forensics," *SIGMOD Rec.*, vol. 30, no. 4, pp. 55–64, Dec. 2001. [Online]. Available: http://doi.acm.org/10.1145/604264.604272

[10] R. Goodman, M. Hahn, M. Marella, C. Ojar, and Y. Westcott, "The use of stylometry for email Author identification: A feasibility study," 2007.

[11] S. Nizamani and N. Memon, "Ceai: Ccm-based email authorship identification model," *Egyptian Informatics Journal*, vol. 14, no. 3, pp. 239 – 249, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S111086651300039X

[12] F. Iqbal, L. A. Khan, B. C. M. Fung, and M. Debbabi, "e-mail authorship verification for forensic investigation," in *Proceedings of the 2010 ACM Symposium on Applied Computing*, ser. SAC '10. New York, NY, USA: ACM, 2010, pp. 1591–1598. [Online]. Available: http://doi.acm.org/10.1145/1774088.1774428

[13] E. Lin, J. Aycock, and M. Mannan, "Lightweight client-side methods for detecting email forgery," pp. 254–269, 2012. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-35416-8_18

[14] M. L. Brocardo, I. Traore, S. Saad, and I. Woungang, *Authorship verification for short messages using stylometry*. IEEE, 2013.

[15] G. Stringhini and O. Thonnard, "That Ain't You: Detecting Spearphishing Emails Before They Are Sent." *CoRR abs/1410.6629*, 2014.

[16] A. Stolerman, R. Overdorf, S. Afroz, and R. Greenstadt, "Classify, but verify: Breaking the closed-world assumption in stylometric authorship attribution," 2011.

[17] R. Chatterjee, J. Bonneau, A. Juels, and T. Ristenpart, "Cracking-resistant password vaults using natural language encoders," in *Security and Privacy (SP), 2015 IEEE Symposium on*, May 2015, pp. 481–498.

[18] K. Crowston, E. E. Allen, and R. Heckman, "Using natural language processing technology for qualitative data analysis," *International Journal of Social Research Methodology*, vol. 15, no. 6, pp. 523–543, 2012. [Online]. Available: http://dx.doi.org/10.1080/13645579.2011.625764

[19] S. Palka and D. McCoy, "Fuzzing e-mail filters with generative grammars and n-gram analysis," in *9th USENIX Workshop on Offensive Technologies (WOOT 15)*. Washington, D.C.: USENIX Association, Aug. 2015. [Online]. Available: https://www.usenix.org/conference/woot15/workshop-program/presentation/palka

[20] V. P. Deshpande, R. F. Erbacher, and C. Harris, *An Evaluation of Naive Bayesian Anti-Spam Filtering Techniques*. IEEE, 2007.

[21] B. Zhou, Y. Yao, and J. Luo, "Cost-sensitive three-way email spam filtering," *Journal of Intelligent Information Systems*, vol. 42, no. 1, pp. 19–45, Feb. 2014.

[22] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song, "Design and Evaluation of a Real-Time URL Spam Filtering Service." *IEEE Symposium on Security and Privacy*, pp. 447–462, 2011.

[23] J. Wang and K. Katagishi, "Image Content-Based "Email Spam Image" Filtering," *Journal of Advances in Computer Networks*, vol. 2, no. 2, pp. 110–114, 2014.

[24] S. Aggarwal, V. Kumar, and S. D. Sudarsan, "Identification and Detection of Phishing Emails Using Natural Language Processing Techniques." *SIN*, pp. 217–222, 2014.

[25] V. Ramanathan and H. Wechsler, "Phishing detection and impersonated entity discovery using Conditional Random Field and Latent Dirichlet Allocation," *Computers & Security*, vol. 34, pp. 123–139, May 2013.

[26] A. Lötter and L. Futcher, "A framework to assist email users in the identification of phishing attacks." *Inf. & Comput. Security ()*, vol. 23, no. 4, pp. 370–381, 2015.

[27] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, ser. NAACL '03. Stroudsburg, PA, USA: Association for Computational Linguistics, 2003, pp. 173–180. [Online]. Available: http://dx.doi.org/10.3115/1073445.1073478