Re-Implementation of That Ain't You!
Spear-Phishing Prevention through behavioral modelling

Extended Abstract

Research Implementation

Anand Yadav
SUNY Albany
ayadav@albany.edu

Sahil Bhasin
SUNY Albany
sbhasin@albany.edu

ABSTRACT

Spear phishing is an email or electronic communications scam targeted towards a specific individual, organization or business. Although often intended to steal data for malicious purposes, cybercriminals may also intend to install malware on a targeted user's computer.

One of the ways in which attackers steal sensitive information from corporations is by sending spear phishing emails. A typical spear phishing email appears to be sent by one of the victim's coworkers or business partners, but has instead been crafted by the attacker. A particularly insidious type of spear phishing emails are the ones that do not only claim to be written by a certain person, but are also sent by that person's email account, which has been compromised. Spear phishing emails are very dangerous for companies, because they can be the starting point to a more sophisticated attack or cause intellectual property theft, and lead to high financial losses. Currently, there are no effective systems to protect users against such threats. Existing systems leverage adaptations of anti-spam techniques. However, these techniques are often inadequate to detect spear phishing attacks. The reason is that spear phishing has very different characteristics from spam and even traditional phishing. To fight the spear phishing threat, we propose a change of focus in the techniques that we use for detecting malicious emails: instead of looking for features that are indicative of attack emails, we look for emails that claim to have been written by a certain person within a company, but were actually authored by an attacker. We do this by modelling the email-sending behavior of users over time, and comparing any subsequent email sent by their accounts against this model. Our approach can block advanced email attacks that traditional protection systems are unable to detect, and is an important step towards detecting advanced spear phishing attacks.

CONCEPTS

Computer Science → Data Mining → Support Vector Classifier, Information Security → Spear-phishing

KEYWORDS

SVM- Support Vector Machines, feature extraction, spear-phishing

1 INTRODUCTION

This is how it works: An email arrives, apparently from a trustworthy source, but instead it leads the unknowing recipient to a bogus website full of malware. These emails often use clever tactics to get victims' attention. For example, the FBI has warned of spear phishing scams where the emails appeared to be from the National Center for Missing and Exploited Children.

Many times, government-sponsored hackers and hacktivists are behind these attacks. Cybercriminals do the same with the intention to resell confidential data to governments and private companies. These cybercriminals employ individually designed approaches and social engineering techniques to effectively personalize messages and websites. As a result, even high-ranking targets within organizations, like top executives, can find themselves opening emails they thought were safe. That slip-up enables cybercriminals to steal the data they need, in order to attack their networks.

1.1 BACKGROUNDS

Traditional security often doesn't stop these attacks because they are so cleverly customized. As a result, they're becoming more difficult to detect. One employee mistake can have serious consequences for businesses, governments and even nonprofit organizations. With stolen data, fraudsters can reveal commercially sensitive information, manipulate stock prices or commit various acts of espionage. In addition, spear phishing attacks can deploy malware to hijack computers, organizing them into enormous networks called botnets that can be used for denial of service attacks.

To fight spear phishing scams, employees need to be aware of the threats, such as the possibility of bogus emails landing in their inbox. Besides education, technology that focuses on email security is necessary.

The traditional security mechanisms implemented on most of the systems only check for phishing. They usually check the email id of the sender, if the email has some executable file attached to it, if it has some links which seem shady.

This kind of mechanism only prevents us from the threats that are immediate. If you open the mail and click on the link or run the program, some hidden code can execute and do some damage. For advanced attacks, this method will fail. The advance attacks will have accurate sender's email, and will seem unharmful to any human. Hence, the system will not be able to detect those.

An attacker can have access to the email address of the user and might be capable of sending an advance level of phishing email, but fortunately, it is highly improbable that an attacker can copy the writing habits of the user.

Every user will have a different writing style. Our model is using this to our advantage.

2 EXPERIMENTAL AND COMPUTATIONAL DETAILS

Our model is built on the idea of classification. In the end, we want to be able to classify our incoming email into one of the two classes, "spear phishing" or not "spear phishing".

We are using LinearSVC as our classifier here and 10-Fold Kmeans cross-validation technique. The loss computation for our SVM has been set to Hinged loss computation. The emails sent by each user are taken as training data to train our classifier. We will get into the concept of the SVM and training later in as our work progresses. For now, to put into simple words, we will be able to build a model which will have most probable/mean value of the feature for each of the features and if there is a 'significant amount' of difference between the incoming email feature and the feature that the model has, that will raise and alarm that this user might not be the actual user.

To do that, we have to implement an algorithm which extracts the features of an email.

Fortunately, we already have different kind of algorithms which are used to calculate the readability of a text, sentence formations etc.

The value of margins to be set for training our classifier model has been set equal to 1 (checked for 5 different (0.01, 0.1, 1, 10, 100) most probable values of margins and found out that setting margins to 1 gives us the most optimum results)

2.1 Features

We are using enron corpus dataset which has around 500,000 emails and around 100,000 sent emails. This is the same dataset used in the original work. Link to download the dataset - https://www.cs.cmu.edu/~enron/enron_mail_20150507.tar.gz.

Currently, we are extracting eleven features from a given email content:

1. Flesch Reading Ease
2. Flesch-Kinkaid grade level
3. Day
4. Time of day
5. Number of sentences
6. Number of words
7. Total length of legit words
8. Number of paragraphs
9. Average words per sentence
10. Average Characters per word
11. Average Sentences per paragraph

Flesch Reading Ease

This test rates text on a 100-point scale. The higher the score, the easier it is to understand the document. For most standard files, you want the score to be between 60 and 70.

The formula for the Flesch Reading Ease score is:

206.835 – (1.015 x ASL) – (84.6 x ASW)

where:

ASL = average sentence length (the number of words divided by the number of sentences)

ASW = average number of syllables per word (the number of syllables divided by the number of words)

Flesch-Kincaid Grade Level test

This test rates text on a U.S. school grade level. For example, a score of 8.0 means that an eighth grader can understand the document. For most documents, aim for a score of approximately 7.0 to 8.0.

The formula for the Flesch-Kincaid Grade Level score is:

(.39 x ASL) + (11.8 x ASW) – 15.59

where:

ASL = average sentence length (the number of words divided by the number of sentences)

ASW = average number of syllables per word (the number of syllables divided by the number of words)

These formulas are already being used by companies such as Microsoft for the analysis of textual contents.

We will try to extract more relevant features from email which can help us uniquely classify a user.

2.2 Implementation Details

Now, let’s dive into how are we using the features to check if an incoming email could be a spear phishing email:

- First, we extract the features from the sent emails of every user.
- Next we set the classifier of every email that is sent by the user (for whom the model is getting trained)as 1 and for every other user as 0.
- Feed all the features along with respective classifier to the SVM(Linear SVC) classifier.
- Now for testing our model we extract feature of the new incoming email.
- Then we supply these features to SVM(LinearSVC) to get a classification result.

Conceptually, we are trying to compare the features of the incoming email with the feature values that we expect the user to have in his/her emails.

This way, spear phishing email will get classified as a non-spear phishing email or a spear-phishing email based on the prediction of SVM Classifier as it is trained according to the habits one particular user.

Here is a comparison between this implementation and the implementation proposed in the research paper given:

1. We are only taking 11 prominent features from the email, while in the research paper the number of features is much higher.
2. We are using LinearSVC while in the paper they used SVM. LinearSVC is more efficient it also trains much faster than the SVM. Look at the link https://stackoverflow.com/questions/35076586/linearsvc-vs-svckernel-linear-conflicting-arguments for more details.

3 Results and Discussion

A system was proposed that protects the identity of corporate users, by checking if an email has been written by the owner of an email account. This work is the first step towards the protection of individuals and companies.

In the original implementation, enron corpus(the same dataset that we are using) dataset was used for training their model. Support Vector Machines (SVC) are used for training, which have around 90% accuracy for at least 1000 emails of users.

From the following dataset, for user Allen (who has almost 1000 mails) we are getting the following prediction scores.

Recall Score :- 0.989746808956

F-1 Score :- 0.984646630847

Accuracy Score :- 0.989746808956

Some other related works include:

Spam Filtering
Origin Analysis
Content Analysis
Forged Message Detection on corpus

4 CONCLUSIONS

Presenting a system that protects the identity of corporate users, by checking if an email has been written by the owner of an email account.

This work is the first step towards the protection of individuals and companies against advanced email attacks, such as spear-phishing. able to detect attacks that state-of-the-art systems are unable to detect.

A HEADINGS IN APPENDICES

A.1 Introduction

A.1.1 Background

A.2 Experimental and Computational Details

A.2.1 Feature

A.2.2 Implementation Details

A.3 Results and Discussion

A.4 Conclusions

A.5 References

ACKNOWLEDGMENTS

This work was originally proposed in a research paper titled ‘That ain’t you! Blocking spear-phishing through behavioral modelling’. Gianluca Stringhini and Olivier Thonnard: Authors of the That Ain’t you!.

We would also like to show our gratitude to the Prof. Amir Masoumzadeh-Tork for sharing his pearls of wisdom and very valuable insight with us during the entire course of this project. We are also immensely grateful for his comments on an earlier version of the manuscript. Although If any errors are discovered they are our own and should not tarnish the reputation of these esteemed persons.

REFERENCES

[1] https://link.springer.com/chapter/10.1007/978-3-319-20550-2_5

[2] https://support.office.com/en-us/article/Test-your-document-s-readability-85b4969e-e80a-4777-8dd3-f7fc3c8b3fd2

[3] https://link.springer.com/chapter/10.1007/978-3-319-20550-2_5

[4] https://support.office.com/en-us/article/Test-your-document-s-readability-85b4969e-e80a-4777-8dd3-f7fc3c8b3fd2

[5] https://link.springer.com/chapter/10.1007/978-3-319-20550-2_5

[6] https://support.office.com/en-us/article/Test-your-document-s-readability-85b4969e-e80a-4777-8dd3-f7fc3c8b3fd2

[7] https://link.springer.com/chapter/10.1007/978-3-319-20550-2_5

[8] https://support.office.com/en-us/article/Test-your-document-s-readability-85b4969e-e80a-4777-8dd3-f7fc3c8b3fd2