UDC 004.056

Analysis of the core research for vendor email compromise filtering model using machine learning

Dmytro Zibarov[1], Oleh Kozlenko[1]

[1] *National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, 03056, Ukraine*

Abstract

Vendor email compromise became one of most sophisticated types of social engineering attacks. Strengths of this malicious activity rely on basis of impersonating vendor that company working with. Thus, it is easy for attacker to exploit this trust for doing different type of data exfiltration or ransom. To mitigate risks, that come with these challenges, information security specialist should consider using different types of approaches, including machine learning, to identify anomalies in email, so further damages can be prevented. The purpose of this work lies in the identification of optimal approach for VEC-style attacks detection and optimizing these approaches with least amount of false-positive (FP) parameters. The object of this research is different methods of text processing algorithms, including machine learning methods for detecting VEC emails. The subject of research in this paper mainly considers impact of mentioned text processing algorithms and its relation with efficiency of VEC email classification, identifying most effective approach and, also, how to improve results of such detections.

Results of this paper consists of details for VEC-email attacks detection, challenges that comes with different approaches and proposed solution, that lies in using text processing techniques and agent-related approach with main sphere of implication – machine-learning systems, that are used for identifying social-engineering attacks through email.

Keywords: VEC, email, machine learning, malicious activity

Introduction

Vendor email compromise (VEC), is a targeted type of Business Email Compromise (BEC) attack in which an attacker impersonates a third-party vendor in order to steal information or assets from that vendor's customers [1]. This form of targeted social engineering attack exploits trust for suppliers as its basic concept. Malicious scenario may include vendor email spoofing, credential theft, etc. VEC detection and prevention systems can use different methods for information security, for example, Agari offers a solution that includes email behavior analysis and IronScales uses a comprehensive approach that consists in analyzing all metadata using machine learning. Main goal of this research is to identify and examine detection methods for VEC-style attacks, challenges that involve these types of attacks and best ways to process data for machine learning algorithms for detection. Of course, any automated approach does not guarantee a 100 percent detection probability, so organizations use the services of analysts for additional control over the system.

1. VEC detection methods

Primary detection methods are:

1. Security training programs for employees. It is very important for a person to be able to independently determine the danger if an attacker can bypass the security edge.
2. Absorption spectroscopy – verification of e-mail authentication standards (SPF, DKIM, DMARC) and detection of fake domain names.

3. Red flags identification: unexpected and unusual changes in typical content, headings, grammatic errors in email
4. Use of proven solutions against phishing attacks. One of the most effective methods of detecting VEC attacks is a comprehensive analysis of email headers, analysis of the author's content and writing style using machine learning algorithms. [2]

Potential VEC indicators may include:

- what appear to be genuine email addresses or usernames;
- suspicious (unusual) mail server or IP address
- suspicious email text (unexpected payment requests, limited response options); [3]

Five hundred emails that were already classified by the commercial system as dangerous with the BEC indicator were collected as malicious emails from the company's database. Since there are no ideal system, it became necessary to additionally review all letters and false positives.

Since the use of real "clean" traffic violates the terms of confidentiality, a dataset with the texts of advertising mailings from Kaggle was used to contrast malicious emails, after which it was artificially expanded by adding the address of the sender and the address of the recipient.

Eventually about 500 .json files containing all email headers were collected. To ease work with them, a small application was written in the Python programming language, which does the following:

- Finds all files with a .json extension in the current directory and stores them in the json_files list.
- Creates a new output.csv file in write mode, with UTF-8 encoding and without adding a new line after each record.
- Creates a writer object of the csv.writer class that allows you to write data to a CSV file.
- Writes the first line to a CSV file with column headers: 'body', 'fullname' and 'mail_address'.

For each JSON file in the json_files list:

- Opens a JSON file in read mode with UTF-8 encoding.
- Loads data from a JSON file into the data variable.
- If data is a dictionary type, then:
  - Gets the value for the 'body' key, or an empty string if there is no such key.
  - Gets the value for the key 'fullname' from the nested dictionary by the key 'sender_email_address', or an empty string if there is no such key.
  - Gets the value for the key 'mail_address' from the nested dictionary by the key 'sender_email_address', or an empty string if there is no such key.
  - Writes the received values to the next line of the CSV file.

Of course, many e-mails use HTML and CSS in the body of the letter, so aadditional function was written to remove all tags belonging to these markup languages, just using regular expressions to remove everything between angle brackets (“<”, “ >").

Text preprocessing is the process of converting unstructured text into a single, structured form suitable for analysis and study. Since most texts are written by a person, it is possible that there will be extra spaces, punctuation marks, letters of a case that does not fit the sentence, etc. Text preprocessing helps:

- To reduce noise and data redundancy. Textual data often contains information that is not useful for analysis, such as punctuation marks, stop words, extra spaces, etc. Preprocessing allows you to remove and/or replace such information and thereby reduce the data set or its complexity. [4]
- To improve the quality and speed of machine learning algorithms. Textual data almost always has a heterogeneous format, such as writing style, cases, word forms, etc. Pre-processing allows you to bring the words to the same format, remove everything unnecessary and simplify the test, and thus the speed and accuracy of machine learning algorithms will be increased. [4]
- To extract useful information and context from textual data. Textual data usually contains implicit or non-obvious information, such as semantics, context, emotion, etc. Pre-processing allows you to convert information into a numerical vector, and thus connect words to each other (represent the features of the text in numbers). [4]

To bring the set of safe letters to the appearance of the set of malicious ones, about 700 more addresses and names were generated, which were added to the body of the blank letter.

Accordingly, to merge the two datasets, only the text of the letter (mail-body), the address of the recipient and the address of the sender were used from the malicious emails.

After conducting this study, researchers found that very rarely did preprocessing improve model estimation, and applying all of them at once improved only the random forest algorithm. It can also be seen that the multilayer perceptron is almost unaffected by pre-processing.

Research results show that it depends on several factors:

- the model used;
- targets that are pursued (for example, if the user is still on FP, then all preprocessing methods and the k-nearest neighbor method can be used, since it exhibits an aggressive policy against positive classifications);
- it is worth noting that I did not focus on the general picture that emerges, but, for example, when using the removal of stop words, most models showed a fairly high and stable result (stable in terms of the ratio).

The F1-score metric, which demonstrates the balance of the model, is also very important. In this context, the multi-layer perceptron during systematization showed the best result.

So, in the end multilayer perceptron model showed the best results when using only stemmatization (the influence of this method is insignificant, but it will be used in the next stages). Also, the random forest with all pre-processing performed well, but worse than MLP. Popular multi-layer perceptron optimization methods can also be applied, for example:

- Adam is a stochastic gradient descent based optimizer that works well for large datasets and deep architectures. It combines the best properties of the AdaGrad and RMSProp algorithms to provide an optimizer that can handle sparse gradients on noisy problems [5].
- L-BFGS (Limited-memory BFGS) is a quasi-Newtonian style optimizer that approximates the Broyden-Fletcher-Goldfarb-Shannot (BFGS) algorithm using a limited amount of computer memory. This is a popular algorithm for parameter estimation in machine learning [5]. It works well for small datasets and provides good gradient estimation accuracy. Because L-BFGS stores only a few vectors that implicitly represent the approximation, the memory requirements are linear, making it particularly suitable for multivariable optimization problems.

The settings for these optimizers (hyperparameters) were chosen:

- adam:
    - lr (learning rate): This is an optimization step that determines how quickly the model changes its parameters during training.
    - betas (moments): These are two hyperparameters that control the exponential smoothing of gradients and quadratic gradients.
    - eps (stabilizing parameter): This is a very small number that is added to the denominator when updating the parameters to avoid division by zero. [6]
- L-BFGS:
    - max_iter (maximum number of iterations): This is the maximum number of iterations of the optimizer.
    - max_eval (maximum number of evaluations): This is the maximum number of evaluations of the loss function. [6]

Different pretreatment methods have been found to have different effects on individual methods. For example, removing stop words had a very negative effect on k-nearest neighbors, but a positive effect on the support vector method. In turn, no pre-processing method had a significant effect on the multilayer perceptron, that is, this method works perfectly with texts of any format.

Having chosen the most successful method of machine learning (multilayer perceptron), the analysis was followed by an attempt to optimize it and to improve its performance, however, regardless of the purpose of the optimizers, they only made it worse (or simply did not give significant results to conclude that they helped), therefore, researchers conclude that (based on researched methods) it would be to use a multilayer perceptron with pre-processing of the text in the form of stopword removal, followed by control from an analyst who will be able to correct the model's errors, after which it will feed the updated data.

Against the background of classic methods, for example, red flags, this approach is better adapted to new types of attacks and will allow to reduce costs for the control team, as a few people are enough to train the model on unknown patterns.

2. VEC detection challenges

In 2022, BEC and VEC attacks changed from trivial mail hacking and bypassing multi-factor authentication to impersonating a law firm and social engineering attacks. Typical BEC/VEC emails usually aim to steal money (change payment information, buy gift cards, pay fake bills, etc.) or steal sensitive company data. [5] According to the FBI, BEC is currently the most expensive digital crime [6]. To face these challenges, detection system must be constantly updated to prevent FP (false positive) and FN (false negative) reactions. In order to draw correct conclusions about the impact of an individual malicious email, it's necessary to apply a comprehensive approach, which includes a complete analysis of email headers and content of its body [4]. It is also necessary to correctly place the indicator weights (for example, a grammatical error in the text is not as critical as a "look-a-like" address), so after the initial analysis specialists will have a list of detected indicators with a conditional sum of weights and if this sum is greater than a threshold - we can say that email is harmful, and if it is not high enough – email is suspicious and user should pay attention to its authenticity.

3. Acknowledgements

Presented research was conducted by its participants with no additional funding and support. Zibarov Dmytro – main researcher, that conducted all practical realization of aforementioned concepts about VEC detection methods and results. Kozlenko Oleh – support researcher, that conducted theoretical evaluation of given results and proposer of main concept of the research.

Conclusions

Business email compromise, including vendor email compromise, is a serious problem for modern business, which can lead to not only financial, but also reputational losses. To prevent information leaks, companies usually use third-party services that provide ready-made solutions to prevent these types of attacks. Typically, security service providers analyze email content, sender details, and email headers to categorize an email and identify potential threats in it.

References

[1] Business Email Compromise — FBI – URL: https://www.fbi.gov/how-we-can-help-you/safety-resources/scams-and-safety/common-scams-and-crimes/business-email-compromise

[2] Detection of Business Email Compromise Attacks with Writing Style Analysis | Request PDF (researchgate.net) – URL: https://www.researchgate.net/publication/358062347_Detection_of_Business_Email_Compromise_Attacks_with_Writing_Style_Analysis

[3] How to Prevent Vendor Email Compromise (VEC) Attacks (xorlab.com) – URL: https://www.xorlab.com/en/blog/how-to-prevent-vendor-email -compromise-vec-attacks

[4] 2022 Wrap Up - The Evolution of Business Email Compromise (BEC) (dacbeachcroft.com) – URL: https://www.dacbeachcroft.com/en/gb/articles/2022/december/ 2022-wrap-up-the-evolution-of-business-email-compromise-bec/

[5] What is Business Email Compromise (BEC)? - Microsoft Security – URL: https://www.microsoft.com/en-us/security/business/security-101/what-is-business-email-compromise-bec

[6] - What is Business Email Compromise (BEC) And How To Prevent It | UpGuard – URL: https://www.upguard.com/blog/business-email-compromise