Comprehensive Method for Detecting Phishing Emails Using Correlation-based Analysis and User Participation

Rakesh Verma
University of Houston
4800 Calhoun Rd, Houston
Houston, TX 77004
rmverma@cs.uh.edu

Ayman El Aassal
ENSIAS
Avenue Mohamed Ben Abdellah Regragui
Rabat, Morocco
elaassal.ayman@gmail.com

ABSTRACT

Phishing email has become a popular solution among attackers to steal all kinds of data from people and easily breach organizations' security system. Hackers use multiple techniques and tricks to raise the chances of success of their attacks, like using information found on social networking websites to tailor their emails to the target's interests, or targeting employees of an organization who probably can't spot a phishing email or malicious websites and avoid sending emails to IT people or employees from Security department. In this paper we focus on analyzing the coherence of information contained in the different parts of the email: Header, Body, and URLs. After analyzing multiple phishing emails we discovered that there is always incoherence between these different parts. We created a comprehensive method which uses a set of rules that correlates the information collected from analyzing the header, body and URLs of the email and can even include the user in the detection process. We take into account that there is no such thing called perfection, so even if an email is classified as legitimate, our system will still send a warning to the user if the email is suspicious enough. This way even if a phishing email manages to escape our system, the user can still be protected.

Keywords

Phishing email; email format; Headers; Body; URL; Comprehensive method; Correlation

1. BRIEF PROBLEM DESCRIPTION

Phishing emails are one of the most dangerous forms of attacks used by attackers worldwide. It is considered an easy way to avoid multiple security layers and it is directed to the weakest link of the security chain, which is the end user. It takes on average 229 days to discover a breach [3]. The user receives an email with a link to a malicious website, or a malware embedded attachment, which is made by the attacker himself. The email is constructed in such a way as to appear as legitimate as possible, which raises the success probability of the attack. It has become easy now for attackers to mimic the email format of an organization or a company and they can either mass-mail it or be more selective. In the latter case it is called *spear phishing* and the email is tailored to each specific recipient, using personal information available in social networking websites (SNW), to appeal to his interests.

CODASPY'17 March 22-24, 2017, Scottsdale, AZ, USA

ACM ISBN 978-1-4503-4523-1/17/03.

DOI: http://dx.doi.org/10.1145/3029806.3029842

2. CONTRIBUTION AND HYPOTHESIS

2.1 Hypothesis

After analyzing a number of emails, we noticed that there always is some inconsistency between different parts of a phishing email. The most suspicious one is where an email is supposedly from a company asking the target to send personal information, and the sender is using a fake name and the email address is not from that company's domain. The better constructed a phishing email is, the more consistent it appears. The best phishing emails are hard to detect and tend to avoid detection measures, and smart attackers send these emails to people who are easy targets (e.g., non IT people). In the literature [16, 11, 14, 7, 9, 13, 15], nobody could detect 100% of phishing emails, this shows the limit of the current automatic detection systems. Hence, it is better to include the user either in the detection process, or at least by just sending him a warning to make him pay more attention [10, 5]. For this purpose, user training is necessary.

2.2 Contribution

We design a comprehensive method, which relies on a set of rules that determine if an email is a phish based on information extracted from all parts of the email. We correlate the information collected from the header, body and URLs contained in the email, and check if it make sense. We verify in our method, if the information extracted from the header relates to the information contained in the body and even the URLs (same company or same domain name for example). We also include the user in a simple and intuitive way, so that the user becomes an asset to the security process rather than being the weakest link. At the end of the detection process, the system sends the user a warning if the email is suspicious enough but was not classified as phish. This will count as a final countermeasure against the imperfection issue of the system, and at the same time sensitize the user and make him more aware of this threat

3. METHODOLOGY

3.1 Header analysis Algorithm

Algorithm 1 Header Analysis Algorithm

1: given an email *msg* (eml or txt format)
2: // Get the necessary information from the header.
3: *address* = email address in "From" field
4: *domain* = domain of email address in "From" field
5: *display* = displayed sender name in the "From" field
6: *message-ID* = right hand side of "Message-ID" field
7: *received1* = a list containing the "from" and "by" tokens in last "Received" field of email header
8: *received2* = a list containing the "from" and "by" tokens in the second last "Received" field of email header
9: *IPaddr1* = Ip address of sending host in last "Received" field of email header
10: *IPaddr2* = Ip address of sending host in the second last "Received" field of email header
11: *match* = initiated at 0 and incremented by one each time the Matching Algorithm returns 'Yes' in Line 22.
12: // analyze the header fields:
13: // Verify if the email passes the "DKIM" or "SPF" authentication
14: **if** $(dkim \parallel spf) = pass$ **then**
15: $auth \leftarrow positive$
16: **else**
17: $auth \leftarrow negative$
18: **if** $auth = positive$ **then**
19: Body Analysis (Semantic and URL analysis)
20: **if** $auth = negative$ **then**
21: // Use reverse DNS Lookup to verify if the IP addresses in the "Received" field belong to the domain name of the claimed hosts.
22: //Call the Matching Algorithm on *domain*, *message-ID*, *received1* two by two.
23: // Decision is then based on the reverse DNS Lookup and the value of the variable *match*
24: **if** $match = 0 \parallel [(IPaddr1 \notin Host$ **in** $received1)$ && $(IPaddr2 \notin Host$ **in** $received2)]$ **then**
25: $email \leftarrow Phish$
26: **if** $match = 1$ && $[(IPaddr1 \notin Host$ **in** $received1) \parallel (IPaddr2 \notin Host$ **in** $received2)]$ **then**
27: $email \leftarrow Phish$
28: **if** $(match = 2)$ && $(nameMatch = No)$ **then**
29: $email \leftarrow Phish$
30: **if** $(match = 3) \parallel (match = 2$ && $nameMatch = Yes)$ **then**
31: Body Analysis (Semantic and URL analysis)
32: **if** *URL Analysis* **is** *Positive* **then**
33: $email \leftarrow Phish$
34: **if** *URL Analysis* **is** *Negative* **then**
35: **if** *Semantic Analysis* **is** *Positive* **then**
36: Send recipient a warning
37: //If nameMatch=No then it is added to the warning.
38: **else**
39: $email \leftarrow Legit$

3.2 Matching Algorithm

DKIM: Domain-Keys Identified Mail is a mechanism that uses digital signature to authenticate multiple email header fields and the sender identity as well [8, 2].

Algorithm 2 Matching Algorithm

1: Given two header fields *head1* and *head2*
2: // Step1: extract domain names
3: *addr1* = *head1*.split("@").lower()
4: *addr2* = *head2*.split("@").lower()
5: // Step2: build list of bigrams for each address (bigrams are separated by ".")
6: // Step2.a: split the domain names using "."
7: *spltAddr1* = *addr1*.split(.)
8: *spltAddr2* = *addr2*.split(.)
9: // Step2.b: build the bigram list for each domain name
10: *BigramtAddr1*=[*spltAddr1*[i:i+2] **for** *i* in range(len(*spltAddr1*-1))]
11: *BigramtAddr2*=[*spltAddr2*[i:i+2] **for** *i* in range(len(*spltAddr2*-1))]
12: // Step3: Compare the two list of bigrams
13: // Step3.a: Define variables
14: *ratio* = 0.0
15: *threshold* = 0.5
16: // threshold can be modified to answer our expectations
17: // Step3.b: Comparison
18: **for** *bigram1* **in** *BigramAddr1* **do**
19: **for** *bigram2* **in** *BigramAddr2* **do**
20: *seqratio*= Levenshtein.ratio(*bigram1*,*bigram2*)
21: **if** *seqratio* > *ratio* **then**
22: *ratio* ← *seqratio*
23: **if** *ratio* >= *threshold* **then**
24: *similar* ← "*Yes*"
25: **else**
26: *similar* ← "*No*"
27: **return** *similar*

SPF: Sender Policy Framework enables to verify that the sending mail server is an authorized sender of the domain that appears in the "mail from" address [8].

The "From" header field shows the email address of the sender, and optionally the name of the sender. The "Received" header field contains two token: *from* which indicate the server sending the email as well as it's IP address; *by* that shows the machine receiving it and the protocol it's using. Each server that receives the emails add it's own "Received" header before transferring it.

URL Analysis: Verify if the URLs in the body of the email link to a malware or a phishing website.

Semantic Analysis: Analyze the body semantically to verify if it urges the recipient to either open an URL and fill in personal information or send that information by email to another address

In the Header Analysis Algorithm: In Line 29, URL Analysis is positive if the URL links to a malicious website or malware. In Line 34, Semantic Analysis is positive if the recipient is asked to send information and/or there is a sense of urgency in the text.

Note that our decision criteria are based on trial and error. Considering the randomness of the header fields' values in emails, we tried to choose thresholds that gave us the best compromise between true positives in phishing emails and true negative in legitimate emails. For the rest of the paper, the results we show are obtained by only the header analysis. In the Header Analysis Algorithm, instead of doing the body analysis, we will consider the email legitimate since it will have passed the rules set for the header.

4. PILOT EXPERIMENT

Dataset: We used 300 Legitimate emails: 100 from leaked DNC[1] and 200 from SpamAssassin's[12] easy ham. For phishing emails we used 400 emails in total; 100 collected from personal mailbox and 300 from Nazario's dataset[6].

Results: Phishing emails: 349 True positives, 51 False negatives. Legitimate emails: 164 True negatives, 136 False positives.

Analysis: 51 FNs phishing emails had similar and therefore consistent header fields. 136 FPs in legitimate emails were a result of inconsistent header values, and in most of them the IP addresses analyzed couldn't be resolved or were not matching the claimed hosts. This further proves the randomness of the content of header fields.

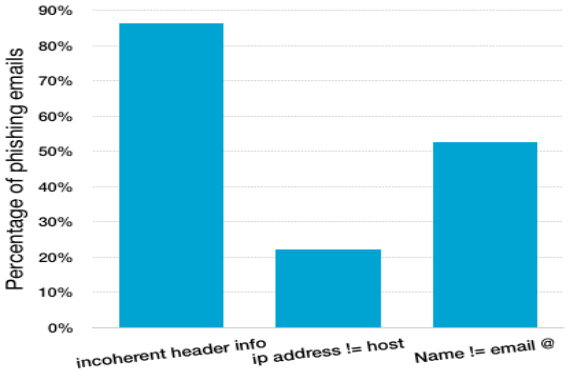

Figure 1: Percentage of phishing emails containing each factor

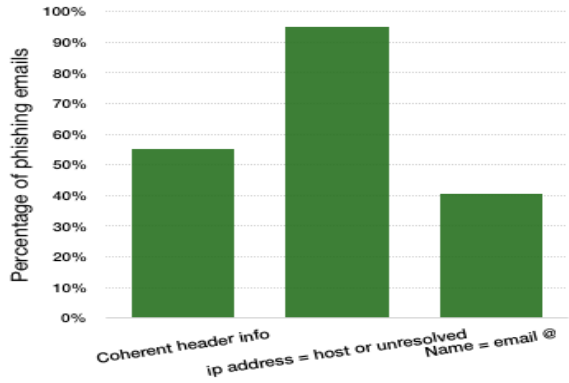

Figure 2: Percentage of legitimate emails containing each factor

5. CONCLUSION & FUTURE WORK

Header fields are an important source of information when it comes to emails. It is useful to analyze the header because we can correlate several pieces of information and verify their authenticity and coherence. However,header fields can not be trusted completely because they can be modified by the attacker and measures can be taken to avoid the authentication mechanisms (like changing the "whois" [4] information of the sending domain). This algorithm is meant to be the first step of the detection process. Higher detection rate and lower false positives may be obtained after implementing the body analysis method. The testing dataset is small but varied, and it gives a general idea about the randomness of the content of the header fields. Future work will be to incorporate into the system the body analysis, and then do a lab experiment. This will help us determine the best way to include the user in the detection process and the best kinds of notifications to use. User training programs must be considered, if we want to rely on user judgment in the detection process.

Acknowledgments

Research supported in part by NSF grants DUE 1241772, CNS 1319212 and DGE 1433817.

6. REFERENCES

[1] Dnc email database. https://wikileaks.org/dnc-emails/.

[2] Domainkeys identified mail (dkim) signatures, rfc4871. https://tools.ietf.org/pdf/rfc4871.pdf.

[3] Fireeye annual threat report 2014. http://investors.fireeye.com/releasedetail.cfm?releaseid=839454.

[4] Official icann whois search. https://whois.icann.org/en.

[5] D. D. Caputo, S. L. Pfleeger, J. D. Freeman, and M. E. Johnson. Going spear phishing/ exploring embedded training and awareness. volume 12, pages 28–38. IEEE, 2014.

[6] J. N. P. Corpus. http://monkey.org/jose/phishing/.

[7] H. Guo, B. Jin, and W. Qian. Analysis of email header for forensics purpose. In *Communication Systems and Network Technologies (CSNT), 2013 International Conference on*, pages 340–344. IEEE, 2013.

[8] A. Herzberg. Dns-based email sender authentication mechanisms: a critical review. volume 28, pages 731–742. Elsevier, 2009.

[9] M. C. Kotson and A. Schulz. Characterizing phishing threats with natural language processing. In *Communications and Network Security (CNS), 2015 IEEE Conference on*, pages 308–316. IEEE, 2015.

[10] A. Neupane, M. L. Rahman, N. Saxena, and L. Hirshfield. A multi-modal neuro-physiological study of phishing detection and malware warnings. In *ACM CCS*, 2015.

[11] F. Sanchez and Z. Duan. A sender-centric approach to detecting phishing emails. In *Cyber Security (CyberSecurity), 2012 International Conference on*, pages 32–39. IEEE, 2012.

[12] SpamAssassin. The apache spamassassin project. http://spamassassin. apache.org/.

[13] R. Verma, N. Shashidhar, and N. Hossain. Phishing email detection the natural language way. In *ESORICS*, 2012.

[14] R. M. Verma and K. Dyer. On the character of phishing urls- accurate and robust statistical learning classifiers. In *ACM CODASPY*, 2015.

[15] R. M. Verma and N. Hossain. Semantic feature selection for text with application to phishing email detection.

[16] R. M. Verma and N. Rai. Phish-idetector: Message-id based automatic phishing detection. In *SECRYPT, IEEE Xplore*, 2015.