Applied Soft Computing 13 (2013) 4841–4848

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Obtaining the threat model for e-mail phishing

Cleber K. Olivo [a], Altair O. Santin [a,*], Luiz S. Oliveira [b]

[a] Pontifical Catholic University of Parana, Graduate Program in Computer Science, R. Imaculada Conceição, 1155, 80215-901 Curitiba, Parana, Brazil
[b] Federal University of Parana, Department of Informatics, R. Cel. Francisco H. dos Santos, 100, 81531-990, Curitiba, Parana, Brazil

ARTICLE INFO

Article history:
Received 31 December 2009
Received in revised form 20 June 2011
Accepted 26 June 2011
Available online 8 July 2011

Keywords:
Security
Threat model
E-mail phishing
Support Vector Machines

ABSTRACT

Phishing is a kind of embezzlement that uses social engineering in order to obtain personal information from its victims, aiming to cause losses. In the technical literature only the hit rate of the classifiers is mentioned to justify the effectiveness of the phishing detecting techniques. Aspects such as the accuracy of the classifier results (false positive rate), computational effort and the number of features used for phishing detection are rarely taken into account. In this work we propose a technique that yields the minimum set of relevant features providing reliability, good performance and flexibility to the phishing detection engine. The experimental results reported in this work show that the proposed technique could be used to optimize the detection engine of the anti-phishing scheme.

1. Introduction

Phishing is a way of embezzlement that uses social engineering to catch victims, deceiving them with the use of technological resources, usually with the goal of obtaining personal information (e.g. financial) and to cause them losses. In the Internet, phishing can reach the user in several ways, e.g., through a web browser pop-up, instant messaging or e-mail. Usually, the victim is persuaded to perform a mouse click to download and install malicious code or access a fraudulent web site without being aware of it.

It is known that the e-mail is the most used Internet service nowadays, so it has been the main resource used to practice phishing [1]. The SMTP (Simple Mail Transfer Protocol), which is used to send e-mails, allows anyone to forge the sender address [2]. In addition, most e-mail clients support HTML (HyperText Markup Language) natively, so all the resources of such a language may be used in a message. Once e-mail supports HTML with hyperlinks (hypertext link used to associate a visible text to an "invisible" URL), they have become a powerful tool for phishers (swindlers).

The techniques used to spread phishing through e-mails are very similar to Spam [3]. In light of this, phishing can be considered as a subcategory of Spam, or even be jumbled with it [4]. However, the negative effects of phishing are usually financial losses due to the identity theft, for example, while Spam – on its most common form, sends e-mail advertisements without the previous approval of the receiver. According to statistics from MessageLabs, about 1% of more than one billion of e-mails exchanged daily were Spam [5].

User security awareness against phishing e-mails is a very important issue since this threat not only applies technical subterfuges to make victims, but also tries to explore the victim's privacy information through social engineering. However, a recent study revealed that users keep themselves vulnerable to phishing even after attend a training program [6]. So, computational solutions are essential to help the user against the many forms of this threat.

Anti-Spam techniques as e-mail filtering may not be effective for the specific problem of phishing detection. Bayesian filters, which are often used to classify e-mail content based on the occurrence of certain keywords, may evaluate incorrectly words that appear in e-mails that were not previously classified as Spam. Moreover, its performance is restricted to the idiom of the e-mail database messages used in the training phase – a requirement to build the filter.

Many e-mail tools, as well as most of the browser tools, apply lists to classify "good" (whitelists) and "bad" (blacklists) sources/senders. Typically, the blacklists block the IP address of the e-mail (SMTP) server, the sender domain, or even the whole e-mail address domain of a sender. Blocking the IP address or domain can cause problems when the sender uses an SMTP server of any provider (e.g. Yahoo, Gmail, etc.), and blocking the whole sender's e-mail address domain can be inefficient because the source address could be forged [4]. Besides, Aaron [7] has shown that 50% of the

* Corresponding author.
E-mail addresses: cleber@ppgia.pucpr.br (C.K. Olivo), santin@ppgia.pucpr.br (A.O. Santin), lesoliveira@inf.ufpr.br (L.S. Oliveira).

1568-4946/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2011.06.016

phishing victims have their credentials stolen in the first 60 min after receiving a phishing e-mail. The problem is that a security vendor needs more than 60 min to identify a phishing campaign and update a blacklist to block it. In the case of phishing, the message origin, the IP address of the phisher and the fraudulent URL tend to change constantly to avoid the sender tracking or its identification. Moreover, the difficulty to deal with whitelists and blacklists can become very complex, because the flow of messages may be very high at the SMTP server – where the filtering is applied. Hence, this approach is usually ineffective for e-mail phishing.

Many studies in the technical literature have considered several e-mail features (properties) to detect phishing, but in general there is no evaluation of the relevance of them when put together (combined). Such an evaluation could reveal a phishing profile – a minimum set of relevant features denoting phishing at a given moment, defined as the threat model.

In many cases, a mere combination of features can result in a good classification hit rate for phishing detection [8–11], but the accuracy of the classifier results cannot be attested [12]. Moreover, the number of features [13] used in the detection engine has a direct impact in the processing time [14] and the phishing detection system may become the bottleneck of the e-mail system. Some works address issues such as classifier accuracy, search for the minimum set of distinct features and impact of inappropriate number of features in the detect engine performance, but in areas of knowledge different of e-mail phishing [13–15].

To the best of our knowledge, the related work published in the literature does not comment anything about the false positive rate – the accuracy of the results provided by the classifier [13] for phishing detection. The accuracy of a classifier is very important because we know that the users tend to ignore the system when they receive many false alerts. Thus, it is preferable a system that only sends accurate and reliable alerts than a system that sends many warnings, but with low credibility.

The technique proposed in this work takes into account a machine learning algorithm to evaluate the importance of each feature when combined with the others to detect e-mail phishing. We also assess the accuracy of the classification rate so that we can obtain the minimum number of features with reliability similar to all features together [15]. The byproduct of all this is the optimization of the processing time of the detection system. It is worthy of remark that our goal is not to create a new phishing detection system, but rather to show the possibility of an optimized engine for the detection systems used nowadays.

This paper is organized as follows: Section 2 considers the phishing features, detection systems based on machine learning and an introduction to the use of ROC curves and AUC. Section 3 presents related work and Section 4 introduces our proposal and results. In Section 5 we present our conclusions.

2. Phishing: features and detection techniques based on pattern recognition and machine learning

In this paper, the phishing problem is treated as a pattern recognition problem, i.e., different features are extracted from e-mails to obtain a model that discriminates phishing messages from non-phishing ones. Therefore, it is a two-class pattern recognition problem.

This section describes the main phishing features, the machine learning model, and also a brief introduction about the ROC curves and AUC, which were used to evaluate the classifiers.

2.1. Phishing features

The strategies used by phishers to trick the e-mail users are highly related to the use of computational resources that are generally unknown by the victims. The phishing detection techniques are based on identifying a set of features known as a technological strategy, usually involving the e-mail header and body. The main characteristics related to phishing detection are listed as follows.

- C1: Hyperlink with visible text like a URL, but pointing to a URL different from the visible text

  In this approach it is used a HTML hyperlink with visible (legible) texts intending to mimic a well-known URL. An example of HTML coding is "*http://www.paypal.com/login.php*". Thus, the visible text shows the *http://www.paypal.com/login.php* hyperlink, but the URL that will be loaded after the mouse click on the hyperlink is *http://playpal.com* [8–10,16].
- C2: Hyperlink with any visible text, but pointing directly to an IP-based URL

  Through this feature phishers do not need to expose their registration data in a DNS server, because queries to DNS will not need, given the malicious web site IP address is explicitly specified in the hyperlink [8–11,16].
- C3: E-mail body coded in HTML format

  The codification of the e-mail body in HTML, supported by almost all available e-mail clients, permits to hide the URL behind the visible text or even an image. The HTML language also allows the use of other technical subterfuges to trick the victims, such as including forms in the e-mail body [10,11].
- C4: Too extensive URL

  The visualization of a URL with a long text often confuses ordinary users because the real domain may be masked by the excessive use of subdomains (separated by dots) or subdirectories (separated by slashes) [10].
- C5: Sender domain different from some URL domain in the message body

  The Mail Sender forgery aims to reach some e-mail servers even they use filters based on blacklists (lists of URLs/domains well known as phishing/Spam origins). However, in the message body there is a malicious URL, which normally is not evaluated when the blacklist is checked. Once the mail sender can be easily replaced, phishers try to mimic the domain of a well-known or trustable sender for the recipient eyes. So the e-mail body has the URL that will deceive the victim [11].
- C6: Image with external domain different from the URLs in the message body

  In this approach, the e-mail body contains images (like logos) that are loaded from the authentic (original) web sites, but the target URL is fraudulent, and obviously different from the domain of the image. For instance, *<img src=http://www.fbi.gov/images/fbi.jpg>You might come to our office!Click here to find out why*. In such a case, the user is supposed to see the true FBI shield plus the message "*You might come to our office!*" and the visible anchor text "*Click here to find out why*", but after the hyperlink has been clicked, the user will download *malware.exe* from *badsite.com*.
- C7: Image origin as an IP address

  This is a typical case of using IP address instead of a registered URL in a DNS domain, but the technique is used to download images; for instance, *<img src="http://200.218.145.21/images_paypal/logo.gif">*.
- C8: Number of domains in the URL

  In this approach, phishers embed more than one true domain in the same URL. The idea is to confuse the user, who usually

thinks that the domain is always closer to the end of the URL. In an URL like "*http://www.badsite.com/login/citibank.com/login.php*" the phisher tries to induce the user to believe that he/she is accessing the legitimate Citibank.com web site, when he/she is actually taken to *badsite.com* [11].

- C9: Number of subdomains in the URL

  Similarly to the feature C8, some phishers add subdomains to make the URL appear like a well-known domain. For example, in "*http://update.citibank.badsite.com*", the phisher expects that the user will pay more attention at "*update*" and "*citibank*", instead of the domain "*badsite.com*" [11].
- C10: Hyperlink with image instead of visible text, and image URL based on IP address

  In this case the hyperlink is an image instead of a visible text, and the associated URL is an IP address. The phisher expects that the user will click on the image to take her to the fraudulent website [16].
- C11: Visible text of the hyperlink does not provide information about its destination

  The fraudulent URL can also be hidden behind any anchor text that is not an URL. Thus, differently of the feature C1, the visible text (anchor) is presented like any text and not like an URL. In other words, the visible text is any text different of an URL. For instance, *Click here to update* [10,16].

2.2. The machine learning algorithm

Several different machine learning algorithms could be used in this work, but one that is quite suitable in this context is the Support Vector Machine (SVM), which was originally designed to cope with two-class classifications problems [17]. The technical literature shows that SVM has been applied with considerable success in various application fields, including the phishing detection [11,18–20].

Consider a set of l samples distributed in a representation space R_n where n is the dimensionality of the sample space. For each sample x_i there is a label $y_i \in \{-1,+1\}$. In our specific case, '−1' represents phishing and '+1' represents non-phishing. According to Vapnik [17] the sample space can be described by a hyperplane separating them according to their labels $\{-1,+1\}$. This hyperplane can be modeled using few samples known as support vectors.

The SVM training phase can be summarized as the support vectors detection from the training database. After that, in test phase, the decision function (1) can be used to provide a class for a non-labeled sample.

$$f(x) = \sum_i \alpha_i \gamma_i K(x, x_i) + b \quad (1)$$

The parameters α_i and b are found by a quadratic programming algorithm, x is the not labeled sample and x_i is the support vector. The function $K(x, x_i)$ is known as the kernel function and maps the sample space to higher dimensions, where the samples become linearly separable.

There are different kernel types that can be used, including Linear, Polynomial, Gaussian and Hyperbolic Tangent. In this work we have tried several different kernels but the one that produced better results was the Gaussian Kernel. The use of a Gaussian kernel requires the user to set two parameters: γ and C. To define such parameters we have used the standard approach, a grid search with cross-validation [17].

2.3. ROC curves and AUC

A Receiver Operating Characteristic (ROC) is a graph that shows the relationship between sensibility and specificity of a classifier. The sensibility can be defined as the probability of correctly classify a sample labeled as positive; while the specificity can be defined as the probability of correctly classify an attribute – whose label is negative. In other words, a ROC curve shows the trade-off between True Positive and False Positive.

One of the main advantages for using ROC curves in the classifier evaluation is the fact that ROC is not sensitive to changes in the class distribution. If the ratio between positive and negative samples in the test database is different from the relationship found in the training database, the ROC curves remain the same [21].

The use of ROC curves may be useful for the visualization and selection of the best classifiers according to their results. The graphics are bi-dimensional and are represented in graph form where the axis Y represents the true positive rate (TPR) and the axis X represents false positive rate (FPR). This makes it possible to define a threshold that will result in the best TPR *vs.* FPR trade-off to the requirements of the system.

Another tool used to represent the efficiency of a classifier is the *Area Under the Curve* (AUC) graph [4]. Unlike the ROC curves, AUC provides a scalar value, which is basically the area under the ROC curve. Once it is a scalar value, the selection of the classifier becomes easier. The higher the AUC the best is the classifier.

3. Related work

Some of the related works that aimed to detect phishing e-mails will be briefly presented and discussed in the following.

Chen and Guo created a client approach based on five main features (including C1 and C2) [8]. The proposed technique reached 96% of detection rate. One advantage of their approach is that the learning phase for the classifier is not necessary. However, if the phishing features change, the formula used in detection can fail. Another negative aspect of their work is the lack of a test database with legitimate messages; therefore it is not possible to measure the false positive rates. Besides, the features are considered isolated and no combination of them was studied.

Cook and his colleagues used a similar technique to the previous one and reached 95.72% of detection rate, using a classifier with 11 features (including C1 and C2) [9]. Although the results reported a good detection rate, some features need clarification about its inclusion in the classification process, others do not provide a convincing explanation about its relevance regarding to phishing. The testing dataset was composed of 81 phishing messages and 36 legitimate e-mails, so the results may be questionable. Furthermore, there was not a separation of the dataset to implement the training and testing phases. Without such a separation the results may become unreliable, once the adjustment of the tool was made with the same messages that led to the reported percentage of accuracy. In addition, this approach needs to search for information on sources outside the e-mail system (e.g. *whois* service), which may increase too much the time needed to analyze each message in real cases.

Fette and his colleagues used a technique that involves machine learning with 10 features (including C1, C2, C3, C4 and C11) [10]. In this approach the detection rate achieved 99.5%, when used in cooperation with an anti-Spam tool. Despite of the high classification rate, this technique needs 10 features, anti-Spam tool and querying external sources (the *whois* service) to discover the "age of a domain" of the e-mail sender or some URL in the e-mail body. Such an approach may increase considerably the time to evaluate

each message, derailing the proposal in a real SMTP gateway that receives a great number of messages.

Basnet and his colleagues reported in their work that using 16 features (including C2, C3, C5, C8 and C9) and an SVM classifier reached the detection rate of 97.99% [11]. The authors' proposal, although using SVM, depends on many features and the identification of keywords in the body of the e-mail message, which can make the approach very slow during the detection phase in real detection systems. The chosen keywords are searched considering the financial sector. Although e-mail phishing historically had been associated with such a sector, statistics showed that this kind of behavior is changing recently, leading to targets as trading, government, etc. [22,23].

The main limitation of the proposals reported in the technical literature is that in general too many features are evaluated without taking into account whether they really are essential to identify phishing. Therefore, it could lead to unnecessary computational cost in phishing detection in real environments with high flow of e-mails. Moreover, some approaches are concerned to report only the detection rate without addressing their accuracy (false positive rates).

4. Proposal

In the beginning of our work we decided to concentrate our efforts to obtain the features that best define phishing. That is, a minimum set of features that combined should define unequivocally the properties (profile) of phishing, which denote the strategy of the attacker – the threat model [24]. So, after an exhaustive search on the technical literature, we identified the consensus features, cited by the majority of authors and adequate explained from the perspective of phishing (described in Section 2.1). Moreover, we evaluated our phishing database to identify those e-mails that represent at least one of the 11 resulting features. Otherwise, we could not evaluate a feature for which there was not at least one phishing e-mail in the database.

During the feature evaluation we avoid querying any external (not native) service (resource) to the e-mail system. In our evaluation those approaches would cause considerable e-mail queue when many messages need to be assessed and their use may not be as effective as expected. In other words, the *whois* service is only efficient whether the domain is recent, but in general the phishers 'hide' themselves under consolidated domains to avoid the detection by this type of query.

When evaluating an e-mail message each set of features will result in a true or false value for phishing. Thus, the classifier has only two classes (non-phishing and phishing). This strategy facilitates the evaluation of the accuracy of the classifier results and the interpretations of the ROC curves. The accuracy is used to assess the false positive rate that a given combination of features provides.

The threat model considers the essential features (phishing properties) combined. It is very common for an IT security professional to judge that a new technique of social engineering is a new phishing feature. Actually, without the use of a method to evaluate the appropriate feature set, such statements are imprecise, as it can be a subset of another feature set, or when evaluated together with the others it may not be distinguished. In more complex cases, the new considered feature may even confuse the detection technique in use, creating a point of uncertainty for the classifier, which will increase the false positive rate.

The performance impact caused by the use of an unnecessary number of features is also object of evaluation of this paper. An unnecessary number of features may burdensome the phishing detection system, because if the feature is correlated to other features, as commented before, it will not help the detection engine. Therefore, it will require spending CPU time to extract a feature that is not useful for phishing detection.

Unfortunately, at each new improvement published to mitigate phishing, a new technique to circumvent it is created by the phishers. One of our goals is to create a systematic procedure to obtain the threat model, providing an easy way to reevaluate and adjust the model being used at any given time. Therefore, the phishing detection engine will be always up to date.

The threat model presents the advantage of considering all relevant features combined, making it difficult for phishers to create attacks that individually are distinct enough to not be included in the set of properties defined by such a model.

Our proposal considers the following stages: (i) preparation of training and testing databases, (ii) execution of the search algorithm, (iii) evaluation of the combination of features providing the best results in the search, (iv) generation of the ROC curves and AUC to evaluate the accuracy of the classifier results, and (v) achievement of the threat model and performance evaluation. The following subsections describe with details each step of the proposed approach.

4.1. Preparation of training and testing databases

The phishing database was built selecting and labeling manually messages provided by the university SMTP server (the messages were filtered from January 2007 to December 2009). The phishing database has 450 unique messages. The non-phishing database has 450 unique legitimate e-mails and was built with authentic messages, including messages like sign up confirmation and true online shopping. Thus there are many phishing messages that are very similar to them. Both databases were divided into two equal parts for training and testing.

Table 1 shows the databases message instances for the phishing and non-phishing features, which are binary (feature present or absent). Table 2 shows a range of discrete values for features C4, C8 and C9 that can assume floating-point values.

One can notice that in some cases the phishing and non-phishing percentage sums up more than 100%. It happens because a message usually has more than one phishing or non-phishing feature.

4.2. Evaluation of combination of features using search algorithms

In order to identify the best combination of features regarding to phishing, we performed tests using the simple Hill Climbing search algorithm [25]. As the search space is small – the number of features provides at most 2^{11} combinations, the algorithm was good enough to provide satisfactory results. The execution of the Hill Climbing algorithm provides the best combination of features used to train the classifiers, their detection rate and accuracy, which were recorded for further analysis. For a greater number of features such task should be performed using a more complex search algorithm, such as Genetic Algorithm [26]. The best detection rate found for each dimensionality of features is shown in Table 3.

One can notice that the detection rate increases until eight features, so it makes no sense to use more than this for the database considered in this work. Details about the best combination of features will be further explained in Section 4.4.

4.3. ROC curves and AUC analysis

The ROC curve is very important for the evaluation of the classifiers since it shows the relationship between false positive rates (FPR) and true positive rates (TPR). Fig. 1 compares the ROC curves of the three best classifiers and the worst one. The best classifiers

Table 1
Databases messages stratification for binary feature instances.

Features		C1	C2	C3	C5	C6	C7	C10	C11
Instance (%)	Phishing	29.1	20.6	96.6	10	68.4	3.5	6.0	64.4
	Non-phishing	0	0	68.8	45.3	3.55	0	0	16.8

Table 2
Databases messages stratification for some discrete values to non-binary feature instances.

Features		C4					C8					C9		
Discrete value		<2	2	3	4	>4	1	2	3	4	5	0	1	>1
Instance (%)	Phishing	6.4	27.7	24.2	21.1	20.4	82.6	14.8	1.78	0.44	0.22	71.3	21.5	7.11
	Non-phishing	33.5	28.6	31.5	6.2	0	100	0	0	0	0	74.2	12	13.7

Table 3
Best classifier detection rate according to the number of features.

Number of features	2	3	4	5	6	7	8	9	10	11
Best hit rate (%)	77.78	77.78	86.66	90.66	92.22	94.44	94.89	94.22	94.22	93.78

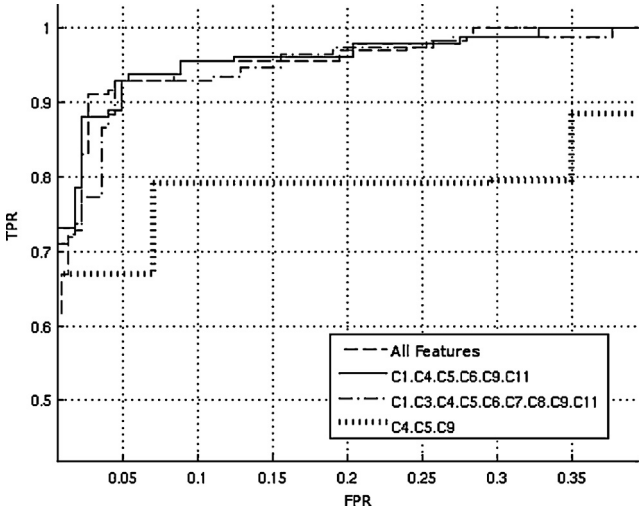

Fig. 1. ROC curves showing the three best and the worst classifier.

6C = {C1.C4.C5.C6.C9.C11}, 9C = {C1.C3.C4.C5.C6.C7.C8.C9.C11} and 11C (all features together) reached similar detection rates.

From Table 3 the accuracy rates of the classifiers shown in Fig. 1 are 77.78%, 92.22%, 94.11%, and 93.78% for 3C, 6C, 9C, and 11C, respectively. The best FPR stays below 2%, except in the case of all the features together (FPR around 3%).

One can notice that if the chosen operating point for the classifiers 6C, 9C, and 11C is near a hit rate of 100%, the ROC (Fig. 1) shows the FPR would be around 30% [12]. So, we argue that the related works should report the FPR together with the classifier hit rate to give a complete report about their experimental results.

Observing the values of the AUC (*Area Under the Curve*) from Table 4, it is clear that the classifiers trained with 6, 9, and 11 features are the best ones since they have the AUC close to one. It is also important to notice that the classifier trained with 11 features, almost the double of six, offers no significant improvement in terms of accuracy.

Table 4
AUCs for the best and worst combination of features.

Number of features	3	6	9	11
AUC value	0.890289	0.980760	0.976770	0.980247

4.4. The threat model

The threat model represents the phishing profile at a given time. It should take place in the phishing detection engine, but it should have flexibility to adapt to the feature set used by the anti-phishing system. In this section the proposed procedure to obtain the threat model is described in details.

Initially, it was analyzed the best combinations of three features (3C), which reached a detection rate of 77.78% in 20 out of 165 possible classifier combinations. Among the 20 results we observed that the most frequent features were C4, C5, C6 and C9. There are four combinations with three of them, 3C = {C4,C5,C6}, {C4,C5,C9}, {C4,C6,C9} and {C5,C6,C9}. This behavior called our attention to these four features, which we named "Threat Model Candidate" (TMC). To confirm the TMC importance we hypothesized that they should appear also in all best combinations for four features or more. Indeed, they really are present in all the best results. Table 5 shows the best combination of features and their classifier detection rates. Highlighted in bold are the features selected as TMC.

The best combination of features was formed by all the TMC four features (4C) and at least three of them were present in the slightly lower classifier detection rates reported in Table 5. Although the best detection rate for 5C does not contain all the TMC features, we considered this case an exception, because all other combinations with best detection rate were composed of the TMC feature subset. Thus, this analysis gave us important evidences that these four features, chosen as TMC, could be a threat model of four features.

4.5. Evaluation of the threat model

In order to evaluate the efficiency of the threat model, we created a procedure to investigate the impacts of the absence of TMC features C4, C5, C6 and C9. The procedure consists of identifying the best detection rate for the classifiers that do not use some combination of TMC features.

Table 6 shows the impacts in the detection rate when removing some of TMC features. The *Random Result* (RR) in Table 6 means that the classifier failed. For instance, considering the combination of two features (2C), it is not possible to classify the messages if those TMC features (C4 and C5) are not present in the detection engine.

It also can be observed from Table 6 that in many cases there were important impacts on the detection rates when some TMC features are removed, but the more interesting case is the simultaneous absence of the features C4 and C5. In such a case, the classifier

Table 5
Best classifier hit rates for each combination of features.

Features	Best hit rate	Features combinations
2C	77.78%	{**C4**, **C5**} and {**C5**, C8}
3C	77.78%	{**C4**, **C5**, Cx}, Cx = {C1}, {C2}, {**C6**}, {C7}, {C8}, {**C9**}, {C10}, {C11} and {**C6**, **C9**, **Cy**}, Cy = {**C4**}, {**C5**}
4C	86.66%	{**C4**, **C5**, **C6**, **C9**}
5C	90.66%	{C3, **C5**, **C6**, C10, C11}
	87.33%	{C3, **C4**, **C5**, **C6**, **C9**}
6C	92.22%	{C1, **C4**, **C5**, **C6**, **C9**, C11}
7C	94.33%	{C1, **C4**, **C5**, **C6**, **C9**, Cx}, Cx = {C7}, {C8}
8C	94.89%	{C1, C3, **C4**, **C5**, **C6**, C8, **C9**, C11}
9C	94.11%	{C1, **C4**, **C5**, **C6**, **C9**, C11, Cx}, Cx = {C2, C3, C8}, {C3, C7, C8}, {C3, C8, C10}, {C7, C8, C10}, {C2, C7, C10}
11C	93.78%	{C1, C2, C3, **C4**, **C5**, **C6**, C7, C8, **C9**, C10, C11}

An underlined element Cx can be replaced by any subset from the list Cx = { }.

reached Random Result for all possible combinations of features, except for 8C. Such results gave us evidences that the classifiers depend totally on these two combined features, hence, they could be considered as a threat model of two features. A similar behavior happens for the sets 3C = {C4, C5, C6} and {C4, C5, C9}. So, it is possible to argue that there are two threat models for three combined features.

One can also notice that no conclusive results were possible in the absence of a TMC features. This shows the importance of the Threat Model Candidate as a phishing profile.

Beyond the evaluation above, we executed a test with the Spam Assassin [27]. The goal of this test was to evaluate if the threat model could be confirmed in real world Spam/phishing detection system, expecting that the detection engine could be simplified.

In the first test the Spam Assassin was trained with the same training database used to obtain the classifiers (Section 4.1). After many adjustments in the Spam Assassin thresholds, the best detection rate achieved was 96.44% (threshold $sa_t = 2.9$).

In the second test, we kept the Spam Assassin threshold optimization ($sa_t = 2.9$) and performed a training only with the phishing database messages containing the TMC = {C4, C5, C6 and C9}. After that we performed the classification in the test phishing database (Section 4.1). The detection rate was 89.78%.

In the second test, the training phase was done using only the messages that contained all the four TMC features, i.e., only 33.7% of the messages of the original database. But, even using only 36% of the features (four out of 11), we obtained a detection rate that is only 6.68% lower than the first test, which have used the whole training database and 11 features. This experiment shows that Spam Assassin could detect the phishing profile/properties of 11 features applying only the four features of the TMC. Inferring that in general for each feature there is a detection rule, we conclude that the detection engine has been significantly reduced, because the feature set was reduced from 11 to 4, what means that 64% of the features were eliminated.

To show that the decrease in the detection rate is not an important issue, we have performed the well-known one-way ANOVA (Analysis of Variance) statistical test. To do that we have executed different experiments varying the threshold sa_t from 2.5 to 3.5. The *F*-ratio value for the recognition rate produced by the ANOVA test was 11.13 ($p < 0.0103$). From the upper critical values of the F distribution at 1% of significance level, we have $F(1,8) = 11.16$, which lead us to conclude that such a decrease in the detection rate is insignificant at 1% of significance level.

We have chosen a threat model composed of four features for the sake of simplicity, but it could be found for any other number of combinations of features. For instance, in the real world application it would be better to work with the 6C classifier to detect phishing since it is more accurate. From Table 3 it is possible to observe that the classifier trained with 6 features reached a detection rate of 92.22%, the best FPR was 2% (Fig. 1) and AUC was 0.9807 (Table 4) – the best AUC in our experiments. Also, the performance of 6C was 21.3% (100–78.70%) better than working with 11C (Table 7). It is worth of remark that it is possible to work with any classifier presented in Table 5, because all of them contain the threat model of 4C – the phishing profile.

4.6. Performance evaluation

Performance is one aspect that should be taken into account in the e-mail detection/test phase, which requires the e-mail fea-

Table 6
Threat model evaluation.

TMC features removed	Number of combined features and best detection rate (%)									
	2C	3C	4C	5C	6C	7C	8C	9C	10C	11C
{C4}	RR	77.78	77.78	90.66	77.78	77.78	77.78	77.78	77.78	–
{C5}	RR	77.78	79.11	84.00	88.66	88.88	88.88	89.77	88.88	–
{C6}	RR	77.78	77.78	82.22	87.55	87.55	92.44	92.66	77.78	–
{C9}	RR	77.78	77.78	90.66	82.00	82.00	82.00	78.00	77.78	–
{C4, C5}	RR	RR	RR	RR	RR	RR	82.00	RR	–	–
{C4, C6}	RR	77.78	77.78	77.78	77.78	77.78	77.78	77.78	–	–
{C4, C9}	RR	77.78	77.78	90.66	77.78	77.78	77.78	77.78	–	–
{C5, C6}	RR	70.88	70.44	82.22	87.55	71.77	87.33	87.33	–	–
{C5, C9}	RR	70.88	70.44	82.00	82.00	82.00	82.00	67.33	–	–
{C6, C9}	RR	77.78	77.78	77.78	77.78	77.78	77.78	78.00	–	–
{C4, C5, C6}	RR	RR	RR	RR	RR	RR	RR	–	–	–
{C4, C5, C9}	RR	RR	RR	RR	RR	RR	RR	–	–	–
{C4, C6, C9}	RR	77.78	77.78	77.78	77.78	77.78	77.78	–	–	–
{C5, C6, C9}	RR	70.88	70.44	RR	66.44	64.22	63.77	–	–	–
{C4, C5, C6, C9}	RR	RR	RR	RR	RR	RR	–	–	–	–
Best Classifier from Table 5 (%)	77.78	77.78	86.66	90.66	92.22	94.44	94.88	94.22	94.22	93.77

Table 7
Average time to extract the best combinations of features.

Features	Combinations	Relative CPU time
4C	C4.C5.C6.C9	43.22%
5C	C3.C5.C6.C10.C11	46.97%
6C	C1.C4.C5.C6.C9.C11	78.70%
7C	C1.C4.C5.C6.C8.C9.C11	84.85%
8C	C1.C3.C4.C5.C6.C8.C9.C11	88.77%
9C	C1.C3.C4.C5.C6.C7.C8.C9.C11	90.82%
10C	C1.C3.C4.C5.C6.C7.C8.C9.C10.C11	97.95%
11C	C1.C2.C3.C4.C5.C6.C7.C8.C9.C10.C11	100.00%

tures extraction in real time. If a large number of features should be extracted, it can become a complex activity and increase substantially the processing time in SMTP Gateways with great flow of e-mails [4,28]. Thus, it was conducted a performance evaluation to understand how some features may increase the CPU time spent in the feature extraction. Table 7 shows the relative CPU time spent to extract the best combination of features. The time required to extract 11 features was taken as reference to calculate the relative CPU time in Table 7. For this test it was not done any scripts optimization.

We observed that CPU time for SVM training and detection/testing phases for 11 and 4 features reduce only 10% for TMC. However, a greater reduction in CPU time takes place during the features extraction. Note that extracting only the TMC (4C) reduces the CPU time in about 56% (from 100% to 43.2%), so the result of this work also contributes towards reducing the processing time.

5. Conclusions

This paper presented a proposal to identify the essential features, which combined define the threat model – e-mail phishing attacker strategy. Threat model prevented the use of irrelevant features in the detection engine and consequent impact on its efficiency. Assisted by the ROC curves and AUC we evaluated the false positive rate to identify efficiently the more accurate classifier to compound the detection engine.

We did not limit the classifier evaluation to the detection hit rate as reported in the technical literature since the accuracy of the classifier is very important for phishing matters. As mentioned before, for a detection system its reliability is more important than its hit rate, because if the alerts are issued without accuracy the e-mail administrator may consider the system unreliable and will tend to ignore any further alert.

As the threat model describes the e-mail phishing attacker techniques in a consolidated way, the phisher shall create new e-mail phishing approaches to succeed. So, our proposal embarrasses the creation of well-known variation of phishing to deceive the traditional e-mail phishing filters.

Experiments using Spam Assassin, an off-the-shelf product, with the threat model 4C improved the classification of e-mails, reducing the set of features in 64% (from 11C to 4C), at cost of only 6.68% in the detection hit rate, which has been demonstrated to be insignificant at 1% of significance level. One can notice that if more accurate classifier is required, it can be chosen from one of the best classifiers presented in this work, because all of them are based on the threat model.

The proposed threat model considers the possibility of changes in the essential features. So, if one of the threat model features reduces its incidence, the model can lose its efficiency. Therefore, a periodic reassessment of the model and a possible reconfiguration of features combinations are very important.

Once phishing can occur in different ways at different times, the administrator can choose between the combinations that provide the best results at each time. Moreover, this task can be easily automated.

Through the performance evaluation test was observed a reduction of about 56% in detection processing time for extracting only the TMC (4C) against 11C. Therefore, the proposal reduced significantly the computational effort, mainly if considered a large flow of e-mail messages like in an STMP gateway.

We also concluded that in some cases, depending on the desirable detection rate and accuracy, the increase in CPU time does not justify the computational cost, i.e., each SMTP administrator can choose the threat model more suitable for her needs. Moreover, the proposed technique for evaluating the effectiveness of the threat model can be used to discover whether certain features no longer exist or decrease its incidence. When this happens, the detection system will be compromised, so this is an important tool to identify when the classifier will fail. To the best of our knowledge, there is no other approach in technical literature that presented this facility so far.

As future works we will provide a database for online queering to the training phase. This motivation arose from the difficulties found to create the e-mail databases used for the development of this work.

Acknowledgments

This research has been supported by The National Council for Scientific and Technological Development (CNPq), Grants 310319/2009-9 and 306358/2008-5, and by State of Paraná Research Foundation (Araucária Foundation), Grant 7374. Cleber K. Olivo wishes to thanks the Coordination for the Improvement of Higher Level Personnel(CAPES)for the scholarship granting.

References

[1] Symantec, Spam decreasing, but social media phishing soaring, says Symantec , Infosecurity 7 (6) (2010) 6.
[2] A. Herzberg, DNS-based e-mail sender authentication mechanisms: a critical review , Computers & Security 28 (8) (2009) 731–742.
[3] C. Kanich, C. Kreibich, K. Levchenko, B. Enright, G. Voelker, V. Paxson, S. Savage, Spamalytics: an empirical analysis of Spam marketing conversion , in: Proceedings of Conference on Computer and Communications Security, 2008, pp. 3–14.
[4] E. El-Alfy, R. Abdel-Aal, Using GMDH-based networks for improved Spam detection and e-mail feature analysis , Applied Soft Computing 11 (1) (2011) 477–488.
[5] MessageLabs, MessageLabs Intelligence. http://www.messagelabs.com/resources/mlireports (retrieved: July 2010).
[6] R. Dodge, C. Carver, A. Ferguson, Phishing for user security awareness , Computers & Security 26 (1) (2007) 73–80.
[7] G. Aaron, The state of phishing , Computer Fraud & Security 2010 (6) (2010) 5–8.
[8] J. Chen, C. Guo, Online detection and prevention of phishing attacks , Communications and Networking in China (2006) 19–21.
[9] D. Cook, V. Gurbani, M. Daniluk, Phishwish: a stateless phishing filter using minimal rules , Lecture Notes in Computer Science (2008) 182–186.
[10] I. Fette, N. Sadeh, A. Tomasic, Learning to detect phishing e-mails , in: International World Wide Web Conference, 2007, pp. 649–656.
[11] R. Basnet, S. Mukkamala, A. Sung, Detection of phishing attacks: a machine learning approach , Soft Computing Applications in Industry (2008) 373–383.
[12] P. Paclik, C. Lai, J. Novovicova, R.P.W. Duin, Variance estimation for two-class and multi-class ROC analysis using operating point averaging , International Conference on Pattern Recognition (2008) 1–4.
[13] S. Shivaji, E.J. Whitehead, R. Akella, K. Sunghun, Reducing features to improve bug prediction , IEEE/ACM International Conference on Automated Software Engineering (2009) 600–604.
[14] K. El-Khatib, Impact of feature reduction on the efficiency of wireless intrusion detection systems , IEEE Transactions on Parallel and Distributed Systems 21 (8) (2010) 1143–1149.
[15] Y. Bazi, F. Melgani, Toward an optimal SVM classification system for hyperspectral remote sensing images , IEEE Transactions on Geoscience and Remote Sensing 44 (11) (2006) 3374–3385.
[16] J. Yearwood, M. Mammadov, A. Banerjee, Profiling phishing e-mails based on hyperlink information , International Conference on Advances in Social Networks Analysis and Mining (2010) 120–127.

[17] V. Vapnik, The nature of statistical learning theory, 2nd ed., Springer Verlag, ISBN: 0387987800, 1999.
[18] H.-S. Kim, S.-D. Cha, Empirical evaluation of SVM-based masquerade detection using UNIX commands , Computers & Security 24 (2) (2005) 160–168.
[19] Y. Pan, X. Ding, Anomaly based web phishing page detection , Annual Computer Security Applications Conference (2006) 381–392.
[20] Y.-G. Kim, M.-S. Jang, K.-S. Cho, G.-T. Park, Performance comparison between backpropagation, neuro-fuzzy network, and SVM , Lecture Notes in Computer Science (2008) 438–446.
[21] T. Fawcett, An introduction to ROC analysis , Pattern Recognition Letters 27 (8) (2006) 861–874.
[22] Anti-Phishing Working Group, Phishing Activity Trends: Report for the month of September, 2007. http://www.antiphishing.org/reports/apwg_report_sept_2007.pdf (retrieved: December 2010).
[23] Anti-Phishing Working Group, Phishing Activity Trends: 2nd Half 2008. http://www.antiphishing.org/reports/apwg_report_H2_2008.pdf (retrieved: December 2010).
[24] J. King, K. Lakkaraju, A. Slagell, A taxonomy and adversarial model for attacks against network log anonymization , ACM Symposium on Applied Computing (2009) 1286–1293.
[25] R. Greiner, PALO: a probabilistic hill-climbing algorithm , Artificial Intelligence (1996) 177–208.
[26] M. Kudo, J. Sklansky, Comparison of algorithms that select features for pattern classifiers , Pattern Recognition (2006) 25–41.
[27] The Spam Assassin Project. http://spamassassin.apache.org/ (retrieved: December 20100.
[28] C. Huang, J. Dun, A distributed PSO–SVM hybrid system with feature selection and parameter optimization , Applied Soft Computing 8 (4) (2008) 1381–1391.