2010 International Conference on Advances in Social Networks Analysis and Mining

Profiling Phishing Emails Based on Hyperlink Information

John Yearwood, Musa Mammadov and Arunava Banerjee
Graduate School of ITMS, University of Ballarat, Vic, Australia
Email: j.yearwood@ballarat.edu.au, m.mammadov@ballarat.edu.au, mrarunbanerjee@gmail.com

***Abstract*—In this paper, a novel method for profiling phishing activity from an analysis of phishing emails is proposed. Profiling is useful in determining the activity of an individual or a particular group of phishers. Work in the area of phishing is usually aimed at detection of phishing emails. In this paper, we concentrate on profiling as distinct from detection of phishing emails. We formulate the profiling problem as a multi-label classification problem using the hyperlinks in the phishing emails as features and structural properties of emails along with whois (i.e.DNS) information on hyperlinks as profile classes. Further, we generate profiles based on classifier predictions. Thus, classes become elements of profiles. We employ a boosting algorithm (AdaBoost) as well as SVM to generate multi-label class predictions on three different datasets created from hyperlink information in phishing emails. These predictions are further utilized to generate complete profiles of these emails. Results show that profiling can be done with quite high accuracy using hyperlink information.**

I. Introduction

'Phishing' can be defined as a scam by which an email user is duped into surrendering private information that will be used for identity theft. Phishing attacks use both social engineering and technical subterfuge to steal personal identity data and financial account credentials. It is one of the fastest growing scams on the Internet. The exclusive motivation of phishers is financial gain. Phishers employ a variety of different techniques from spoofed links to malware (keyloggers) to DNS Cache Poisoning [1] (which is also known as 'Pharming') to lure the unsuspected user into divulging their personal information [2]. Spoofed emails would contain phishing deception methods like hidden addresses that are spoofed like *http://www.commbank.com.au.stpr.ru/* instead of the original address as *http://www.commbank.com.au/*. They also exploit different vulnerabilities in the browser like hiding the address of the actual site in the status bar. Also malicious software redirects users to spoofed sites.

Usually, a spoofed email is sent to a large group of people from an address that appears to be from their bank or some other legitimate institution. The email is typically worded to instill a sense of urgency and to elicit an immediate response from the recipient. For example, 'verify your account details or your account will be closed'. The hoax email also contains a link to an online form that is branded to look exactly like the organization's website. The form has to be filled in using sensitive information like passwords, user account details, credit card details. Until recently most phishers used the names of financial institutions to deceive people into giving away their account information. They now use the names of other organizations like eBay and Apple.

There have been many approaches to detect and prevent phishing attacks like anti-phishing toolbars, and scam website blockers [3], [4], [5]. Further machine learning approaches have also been devised for this purpose [6], [3]. Also another approach to develop an architecture for detecting phishing is proposed in [7], [8]. For example, the eBay Toolbar is a browser plugin that eBay offers to its customers, primarily to help them keep track of auction sites. The toolbar has a feature called 'Account Guard' that monitors the domain names that users visit and provide warning in the form of a coloured tab on the toolbar. The tab is usually grey but it turns green if the user is on eBay or a PayPal site. It turns red if the user is on a site that is detected as spoofed by eBay. Similarly spoofguard is a Internet Explorer browser plugin that warns users when webpages have a high probability of being spoofed.

The phishing problem has been and still is very important, and the detection and warning approach taken to the problem is not enough. The existing literature mainly deals with phishing detection problems. The main problem addressed in the literature is the detection of phishing emails based on some significant features that they possess. In this work a different aspect of phishing is investigated, namely the profiling of phishing emails. Phishers usually follow a variety of techniques, so a profile can be expected to show a conglomeration of different activities. Profiles can be understood as metadata on phishers, in particular, information on activities of a related individual or a group involved in the activity. Profiles can be ascertained to provide information on different phishers involved in the activity. By generating profiles, phishing activities can be better understood as well as monitored. In this paper we describe an approach based on representing a profile as a set of labels (classes) identified in the phishing emails that align with characteristics useful for profiling. Multi-label classification is used on the links within the emails to predict a set of labels that form a part profile of the phishing activity.

The paper is organized as follows. Section 1 provides an introduction to phishing and some background on the literature surrounding the problem. Section 2 focuses on profiling. Sections 3 and 4 describe our formulation of the problem and the data sets that are used and generated to form a basis for this approach. Section 5 presents the classification algorithms used and the evaluation measures.Sections 6 and 7 present the results.

978-0-7695-4138-9/10 $26.00 © 2010 Crown Copyright
DOI 10.1109/ASONAM.2010.56

II. PROFILING

'Profiling is a data surveillance technique which is little understood and ill-documented, but increasingly used. It involves generating suspects or prospects from within a large population, and inferring a set of characteristics of a particular class of person from past experience' [9]. In [9], different data surveillance techniques such as front-end verification and data matching have been surveyed. It has been found that profiling data requires different sets of measures and there are different problems that need to be tackled in this area. We take the definition of profiling as in [9]: 'Profiling is a technique whereby a set of characteristics of a particular class of person is inferred from past experience, and data-holdings are then searched for individuals for close fit to that set of characteristics.' Furthermore numerous potential areas for the use of profiling have been identified as well, such as patients who have a likelihood of suffering from certain diseases or disorders, students having potential artistic talents and many others. However the potential use of profiling has been to identify customers buying patterns and market products accordingly.

Certainly profiling has been in vogue, particularly in areas like 'Market Basket Analysis' [10], [11] that profiles customers based on their buying patterns which can further be used by companies to ascertain the nature of competitive markets. Also there have been studies in 'Investor Profiling' [12], [13], wherein an individual's investment decisions are taken into accoiunt and used to underline the policies and marketing strategies of investment companies. More recently 'Offender Profiling'[14], [15] in Forensic Psychology [16] is used to identify perpetrator(s) of a crime, based on nature of the offence committed and its mode of operation [17], [18]. This leads to determination of various aspects of criminal psychology before, during and after the crime is committed.

Further 'Customer Profiling' which deals with gathering non-sensitive data about customers (like age, buying patterns and others) is a very important tool in customer relationship management (CRM) activities of companies as can be found in the survey in [19]. Furthermore as is mentioned in the above survey 'the more information on customers, the better equipped an organization will be to cater to the needs of their customers'.

In this paper, we follow the same trend set up by these studies, to profile phishing emails based on the structural characteristics of the emails received by persons and the information derived on hyperlinks from 'Whois Database'[20]. Since different domain names from different countries were present in the hyperlinks, whois information had to be generated after querying different whois databases, such as the Asia Pacific Network Information Centre (APNIC) and the Rseaux IP Europens Network Coordination Centre (RIPE NCC) [21], [22]. In our work on understanding phishing activity from a social engineering and social networking point of view we are interested in categorizing the activities of phishing groups and devising techniques for automatically obtaining parts of the group profile.

III. THE PROBLEM FORMULATION

In this paper, we use data mining to help profile phishing emails. Usually a phisher contacts a victim through emails; hence we take the most significant part of the email - the hyperlink information as features. For our experiments, we develop three different datasets from hyperlink information for generating profiles. We use characteristics like the structure of the emails sent by the phishers to their potential victims (which we call *structural information*) and metadata on the hyperlinks - the *whois information*. We consider these characteristics as classes that will correspond to labels in a multi-label classification problem.

Utilizing a data classification technique will provide the relationships between the hyperlinks in the phishing emails and their pre-specified categories/classes. Further we can use the multi-label classifier to assign unknown emails to their categories or classes and therefore to particular attrubutes in their profiles. Familiarity with the data provides confirmation that most examples would provide multiple labels that would be informative in terms of profiling.

The approach suggested considers:

- Accessing features from the emails that are simple and effective.
- The particular characteristics of the emails that can be considered as attributes in profiles.

Our view is that profiles should be able to distinguish between different groups. For example, an email may have the following characteristics, it has, a table, an image and so on. Another group may have different subsets of these characteristics. Phishers have different modus operandi or ways of working. In one case, phishers have different ways of handling phishing activity. Some phishers may embed scripts and images in the form which can safely pass detectors and when clicked by the user takes them to a site that is not the original one. In other instances, another group might insert a fake link in the form and when clicked will take the user to a phishing site. Hence the modus operandi is different for different groups. Based on this fact, we would want to identify groups using the different forms of structures embedded inside emails. If we define the feature set as consisting of these characteristics then data clustering would provide different groups having similar profiles. This problem that has been considered in [23]. Preliminary analysis shows that there are many difficult problems in clustering. Different algorithms give different cluster results. In this paper we follow a different approach. We choose these characteristics as classes and try to predict a set of classes or labels of new emails. The feature set used in this case, is essentially the hyperlink information from emails.

IV. CHARACTERISTICS OF THE DATA

Based on two different types of information (that can be readily obtained), classes were selected for generating profiles. They are: (1) Structural Properties of the emails sent to victims, which present salient characters of the emails (2) Whois

properties of the hyperlinks, which gives detailed information about a domain hosted on the internet.
The structural properties that could be used as classes are:

(a) textcontent - binary value specifying if the email had a text part or was solely an html email. It has been observed that most phishing emails have multiparts attached to them such as text and html parts. This would be '1' if the email had a text part and '0' otherwise.
(b) vlinks - specification of the number of visible links in the email. The value for this class is '1' if the number of visible links is greater than zero and '0' otherwise. Visible links are mainly used in a phishing email as a disguise for the actual hyperlink.
(c) htmlcontent - binary value specifying if the email had a html part or was solely a text email. This would be '1' if the email had html part, '0' otherwise In case of both parts being present in the email, both *textcontent* and *htmlcontent* would have the value of unity.
(d) script - binary value specifying if the email has an embedded script. '1' if email had scripts, '0' otherwise. It has been noted that scripts are an important part of phishing emails as they are usually not picked up by the anti-phishing toolbars. Scripts can perform myriad of activities - like opening hoax site in another window or storing the username and password. Presence of certain scripts might be a good way to generate a profile.
(e) table - determines the number of tables in the email. Value for this class is '1' if the number of tables is greater than zero,'0' otherwise. Tables are useful in profile generation as the data in each row of the table can be made to form a hyperlink to some hoax sites. Hence presence of the tables can be used in profile generation.
(f) image/logos - determines embedded images in the email. Value for this class is '1' if the number of images are greater than zero,'0' otherwise. Images are an useful tool for profile generation since some emails sent by phishers come in multipart format containing image and text part. Images in particular act as hidden link in transfering the unsuspected user to a phishing site. Hence the presence of images in an email can be used in profiling.
(g) hyperlinks - determines the number of hyperlinks in the email. Value for this class is '1' if the number of hyperlinks is greater than zero,'0' otherwise. As was discussed earlier, presence of hyperlinks are an important part of an email and phishers take great care in hiding these links.
(h) formtag - binary value, '1' if the email had a form embedded, '0' otherwise. Presence of forms in an email would probably open up a data entry window and ask the user to enter their information. On submission, the data would be transferred to the hoax site that is set as the action. Hence this is useful for profiling.
(i) faketags - number of faketags in the email. Value for this class is '1' if the number of faketags are greater than zero, '0' otherwise. The faketags are important because they are thrown into the emails to confuse the phishing email detector.

From these structural properties, the email characteristics sent by an individual or a group of phishers can be identified. We use these structural classes for generating the profiles in all the above-mentioned datasets.

Another set of classes are generated from the whois properties of the hyperlinks themselves. Since the hyperlinks are from different countries and were hosted on different domains, information from a number of whois databases were used to generate the classes mentioned here. In recent work [3] for detecting phishing emails, the authors had also used 'whois' information to select appropriate features for their learning algorithm. In our case we use whois information as classes. Based on the available information retrieved from the databases the whois classes were assigned manually. It was realized that three different types of whois classes could be determined from the information found on the embedded hyperlinks. The classes that were generated are as:

(a) Hacked_Site - if a legitimate site was hacked and used to send emails to customers then the value of this would be '1', '0' otherwise.
(b) Hosted_Site - if a site was hosted on a server and was used to send emails and receive responses then the value of this would be '1', '0' otherwise.
(c) Legitimate_Site_Addition - This denotes a hosted site with addition to a legitimate domain. If a site was hosted on a server and its name was just an extension to a legimitate domain address, then this value would be '1', '0' otherwise.

Whois classes are of great significance since they help us to profile the activity of the phishers, whether an individual host their own site or hack a site or host a site very similar to the original, just an addition to it. Particularly the latter could be hosted in different domains and on different servers. Hence identification of these classes are crucial to profiling phishing emails.

We select combination of different characteristics as classes to generate the different datasets. The aim being to identify the prominent characteristics that can be used for effective identification of emails. The choice of these classes is based on the rarer characteristics that are prominent in emails but are not so prominent as to be present in most emails. We describe the selection of these classes in Section IV-D.

A. Information on data

The phishing emails in this paper are 2048 emails which are obtained from a major Australian Bank. These are emails gathered by their information security group and have been

identified/ detected as phishing emails. Most emails have been collected over a span of 5 months. Most of the emails are of 1026 characters in length and have text as well as hyperlink content embedded in them. Some of them contain html structures like script, tables, images and other structures that can be useful in identifying the structures of the emails and hence the modus operandi of the phishing group or activity. In this paper, we create different datasets from hyperlink information in phishing emails. We utilize 2048 emails which were previously detected as phishing emails. While extracting hyperlinks some emails that did not have any hyperlink information were removed. The final set of documents containing hyperlinks were ascertained to be 2038. The datasets generated are listed hereunder. In all these datasets the classes defined are from structural characteristics of emails and from whois information as described in Section IV-D.

B. Generation of datasets

Hyperlink Based ($\mathcal{L}$)

In this dataset, a complete hyperlink present in an email is taken as a feature. Hyperlinks specify links to a resource usually on the web. In a phishing email a hyperlink is usually kept hidden from the user. To generate datasets these hyperlinks were extracted from the emails. Hyperlinks can usually be found as values of *href* attribute of an anchor $< a >$ tag within an email. Emails can have one or more than one feature based on whether one or multiple hyperlinks are present. Hyperlink extraction in phishing emails is particularly more troublesome, because of the presence of spurious tags (similar to the anchor tag) like $< acf >$ to confuse the parsers. Phishers do this to ensure that their hidden links are not picked up by the anti-phishing toolbars and the like. Also junk text deliberately included in the emails makes it more difficult to determine the content. From this dataset we aim to find out whether extracting an unseen hyperlink can provide useful information on the profile of the phisher.

Hyperlink Suspected Component Based ($\mathcal{L}_{sus}$)

In this dataset, the extracted hyperlink is broken down and only the 'suspected part' is taken. By 'suspected part' we mean that part of the hyperlink which contains information about the directory structure of the link. Usually, a phisher lures an unsuspecting victim to a site which is usually located at a convenient location within a personal directory created by the phisher. So this directory holds all the related files that phishers use to achieve the objective of fetching sensitive information from victims. Hence, a link from any hosted server to this particular directory can be regarded as suspected link, wherein the suspected part is the link to this directory. Moreover, it has been observed, that in some hyperlinks although the hosted server remains the same, the directory structure changes as victims from different financial institutions are attacked. Hence, we call this particular dataset *Hyperlink Suspected Component Based* taking into consideration these facts. An example of suspected part in a hyperlink would be *phishing/html/index2.files* assuming that the given hyperlink is *http://www.domainname.com/phishing/html/index2.files*. Our aim in generating this dataset is to identify whether the unseen directory structure of a hyperlink, can provide profile information.

Hyperlink Template Based ($\mathcal{L}_{temp}$)

An extracted hyperlink is broken down further into its template format in this dataset. By 'template format' the constituent parts of the hyperlink is meant. In this dataset we break a hyperlink down into its constituent elements. For example, given a hyperlink:
http://www.domainname.com/phishing/html/index2.files, the template format would be *www.domainname.com*, *phishing*, *html*, *index2.files*. Hence an email in this dataset would usually have multiple features. In generating this dataset, the aim is to study if given an unseen template can we predict a profile.

The idea behind generating these datasets is that phishers employ different varieties of email links to hide their destination link from the victim. Essentially, these datasets are designed to pick up these different formats. Another point worth mentioning here is that we would also be interested in observing how these three profiles generated correspond to each other.

C. Choice of hyperlinks as features

A hyperlink in an html page signifies a link to a resource on the web that can be loaded in the browser when some event occurs - for example, mouse click on the hyperlink. Phishers usually utilize this technique to transfer an unsuspecting user to a hoax site. Usually emails are the *modus operandi* of phishers trying to contact their potential victims. Emails would not have been as useful for phishing activity if hyperlinks could not be embedded in them. Hence, to a phisher, an embedded hyperlink in an email is the most important feature. Certainly, a lot of care is taken to disguise an embedded link in various ways, for example an email might consist of a link embedded in a picture that the user sees on opening the email. Accidentally clicking on this picture would send the user to a phishing site. There are also other techniques of making the embedded link invisible to the user by not letting it appear in the status bar. Since this is such an important feature to a phisher, we take hyperlinks as features for generating profiles.

D. Selection of Classes

When generating profiles we need to use those characteristics that would be best at distinguishing between the different phishing groups. For this reason, we will consider the occurrence frequencies of the characterisitcs considered above. We have, out of all structural characteristics the following frequencies: From Table I, it can be seen that some classes have very high frequencies, hence these classes are not taken into account as they are likely to be poor discriminators between the emails. From these classes, we further generate

TABLE I
CLASSES AND THEIR FREQUENCIES

Structural	Classes	Frequency
	vlinks	2037
	htmlcontent	1962
	link	1938
	image	1080
	table	868
	faketags	645
	textcontent	529
	script	79
	form	48
Whois	Classes	Frequency
	Hosted_Site	1252
	Legitimate_Site_Addition	807
	Hacked_Site	146

two different sets of classes for classification purposes as mentioned below. In all these cases we remove the most frequently occurring classes from the list of classes.

```
Class Set 1 - Remove maximum frequency
classes:
(There are 9 classes in this case)
link, image, table, faketags,
textcontent, script, form,
Hosted_Site, Legitimate_Site_Addition,
Hacked_Site

Class Set 2: Remove maximum frequency
classes and whois classes:
(There are 6 classes in this case)
link, image, table, faketags,
textcontent, script, form

Class Set 3: Only Whois classes:
(There are 3 classes in this case)
Hosted_Site, Legitimate_Site_Addition,
Hacked_Site
```

We are left with multiple classes and our idea of treating the problem of generating a profile as a multi-label classification problem is based on these classes as labels which will constitute elements of the profile.

V. CLASSIFICATION ALGORITHMS AND EVALUATION MEASURES USED

A. Algorithms

In the experiments, we use two different algorithms. *BoosTexter*, proposed in [24], is a well-known classification algorithm developed for multi-label classification problems. It is based on boosting concept in machine learning [25]. Boosting increases classifier accuracy by combining rules generated at each round by a weak learning algorithm. *BoosTexter* uses two algorithms to solve multi-label classification problems, namely *AdaBoost.MH* and *AdaBoost.MR* [24]. It generates more accurate classification rules after sequentially calling the weak learner in a series of rounds. In our experiments, we run *BoosTexter* for 300 rounds. Another classification algorithm that we use to generate profiles is *SVM_light* - an implementation version of Support Vector Machines [26].

B. Evaluation Measures

To determine classifier accuracy for multi-label classification, we use the following measures for performance analysis. These measures proposed in [24] are specially designed for multi-label classification problems. They are namely, *One-Error*, *Coverage* and *Average Precision*. We use the modified versions of these measures given below.

Let $\mathcal{X}$ be the set of all documents. The classification algorithm generates a prediction vector $\mathcal{H}(x) = (\mathcal{H}_1(x), \cdots, \mathcal{H}_c(x))$ where c is the number of classes for each document $x \in \mathcal{X}$. The maximal value of $\mathcal{H}_i(x), i = 1 \cdots c$ indicates that the document x is more likely to belong to class i. In the following, the notation $|S|$ represents the cardinality of the set S.

1) One-Error:

   This measure evaluates how many times a 'maximal' class predicted has not occured in expert vector for the class. Let, as before, $\mathcal{H}(x) = (\mathcal{H}_1(x), \cdots, \mathcal{H}_c(x))$ be a set of prediction classes, where c is the number of classes. In cases, where there are more than one class, having the same maximal weight in the predicted vector, this measure needs to be defined. Consider $\mathcal{H}^*(x) = \{i \in \{1, \cdots, c\} : \mathcal{H}_i(x) = max\{\mathcal{H}_1(x) \cdots \mathcal{H}_c(x)\}$, and $\mathcal{Y}^*(x) = \{i \in 1, \cdots, c\} : i \in \mathcal{H}^*(x)$ and $\mathcal{Y}_i(x) = 1$ $\}$. Then one-error is defined as:

$$E_{one-error} = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} (1 - \frac{|\mathcal{Y}^*(x)|}{|\mathcal{H}^*(x)|}) \quad (1)$$

2) Coverage :
   This measure evaluates the performance of a classifier for all classes that have been observed. Given $x \in \mathcal{X}$, let $\Gamma(x)$ be the set of all ordered classes $\tau = \{i_1, \cdots, i_c\} \subset 1, \cdots .c$ statisfying $\mathcal{H}_{i1}(x) \geq \cdots \geq \mathcal{H}_{ic}(x)$. Then according to the class vector $(\mathcal{Y}_1(x), \cdots, \mathcal{Y}_c(x))$, the rank and error is defined as:

$$rank_\tau(x) = max\{n : \mathcal{Y}_i(x) = 1, \; n = 1, \cdots, c\}; \quad (2)$$

$$error_\tau(x) = \frac{rank_\tau(x)}{||\mathcal{Y}(x)||} - 1 \quad (3)$$

   Obviously the terms $rank_\tau$ and $error_\tau$ depend on the order of τ. One way to avoid the dependence on ordering is to take the middle value of maximal and minimal ranks. In this work, this value is used as the measure. This can be defined as:

$$rank(x) = \frac{1}{2}(rank_{max}(x) + rank_{min}(x)); \quad (4)$$

   where

$$rank_{max}(x) = max_{\tau \in \Gamma(x)} rank_\Gamma(x)$$
$$rank_{min}(x) = min_{\tau \in \Gamma(x)} rank_\Gamma(x)$$

The numbers $rank_{max}(x)$ and $rank_{min}(x)$ are associated to the worst and best ordering respectively. To define coverage the following formula will be used.

$$E_{cov} = \quad = \quad \frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}}\left(\frac{rank(x)}{||\mathcal{Y}(x)||}-1\right) \qquad (5)$$

It must be noted that $E_{cov} = 0$ if a classifier makes predictions such that for all $x \in \mathcal{X}$, the observed reactions are placed on the top of the ordering list of weights $\mathcal{H}_i(x)$.

3) Average Precision :
Let $Y(x) = \{l \in \{1,\ldots,c\} : \mathcal{Y}_l(x) = 1\}$ be the set of classes that have been observed for an example x and $\mathcal{H}(x) = \{\mathcal{H}_1(x),\cdots,\mathcal{H}_c(x)\}$ be predicted classes calculated. $\mathcal{T}(x)$ denotes the set of all ordered classes $\tau = \{i_1,\ldots,i_c\}$ satisfying the condition

$$\mathcal{H}_{i_1}(x) \;\geq\; \ldots \;\geq\; \mathcal{H}_{i_c}(x);$$

where $i_k \in \{1,\ldots,c\}$ and $i_k \neq i_m$ if $k \neq m$. In the case, when the numbers $\mathcal{H}_i(x), \quad i = 1,\cdots,c,$ are different, there is just one order satisfying this condition. But if there are classes having the same weights then predicted classes can be ordered in different ways; that is, in this case the set $\mathcal{T}(x)$ contains more than one order. Given order $\tau = \{\tau_1,\ldots,\tau_c\} \in \mathcal{T}(x),$ the rank for each class $l \in Y(x)$ as $rank_\tau(x;l) = k$, where the number k satisfies $\tau_k = l$. Then *Precision* is defined as:

$$P_\tau(x) = \frac{1}{|Y(x)|}\times$$
$$\sum_{l\in Y(x)}\frac{|\{k\in Y(x) : rank_\tau(x;k) \leq rank_\tau(x;l)\}|}{rank_\tau(x;l)}.$$

This measure has the following meaning. For instance, if all observed classes $Y(x)$ have occurred on the top of ordering τ then $P_\tau(x) = 1$. Clearly the number $P_\tau(x)$ depends on the order τ. This is defined as

$$P_{best}(x) = \max_{\tau\in\mathcal{T}(x)} P_\tau(x)$$

and

$$P_{worst}(x) = \min_{\tau\in\mathcal{T}(x)} P_\tau(x)$$

which are related to the 'best' and 'worst' ordering. Therefore, it is sensible to define the *Precision* as the midpoint of these two versions: $P(x) = (P_{best}(x) + P_{worst}(x))/2$. *Average Precision* over all records $\mathcal{X}$ will be defined as:

$$P_{av} = \frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}} P(x). \qquad (6)$$

For all experiments conducted in this work, the above-mentioned measures are used as the performance measures for the determination of classifier accuracy. From the above, it can be seen that *Average Precision* is more suitable for multi-label evaluation problems.

VI. Classification Results

In generating predictions from *BoosTexter* described in Section V, we use a bag-of-words approach in which hyperlinks from emails are the features and structural and whois information are the classes. Hence, we would have an input feature vector and an input class vector being provided to the algorithm to generate a prediction vector. To evaluate the classifier's accuracy we perform four-fold cross-validation on all the datsets mentioned in Section IV-B. Further we evaluate classifier performance using the performance measures described in Section V. *BoosTexter* achieves quite high accuracy on these datasets which means that the profiles generated by *BoosTexter* are quite accurate. Results of *One-Error*, *Coverage* and *Average Precision* from Boostexter are presented in Table II. The results are averaged over four folds. Further results from SVM using the linear kernel have been presented .

TABLE II
Boostexter results on the test set of different phishing hyperlink based datasets. nDocs denotes the number of documents present in the dataset. *One-Error*, *Coverage* and *Average Precision* is denoted by One-Err, Cov and Avg-Pr respectively.

DatasetName	nDocs	One-Err	Cov	Avg-Pr
$\mathcal{L}$	2038	0.001	0.05	99.05
$\mathcal{L}_{sus}$	1805	0.001	0.043	99.16
$\mathcal{L}_{temp}$	2038	0.001	0.03	99.30

TABLE III
SVM linear kernel results on the test set of different phishing hyperlink based datasets. nDocs denotes the number of documents present in the dataset. *One-Error*, *Coverage* and *Average Precision* is denoted by One-Err, Cov and Avg-Pr respectively.

DatasetName	nDocs	One-Err	Cov	Avg-Pr
$\mathcal{L}$	2038	0.012	0.096	98.56
$\mathcal{L}_{sus}$	1805	0.085	0.122	95.49
$\mathcal{L}_{temp}$	2038	0.009	0.079	96.23

Hence for these types of datasets boosting algorithms can be a suitable choice for generating profiles. The above results also show that, *Average Precision* is higher on *Hyperlink Template Based* dataset which could be expected, since breaking a hyperlink into separate parts will generate more features for the algorithm to learn. The important fact is that accuracy on the test set also increases. Furthermore, number of examples in the *Hyperlink Suspected Component Based* dataset is less than the others, since not all hyperlinks do have a link to the directory structure.

VII. Profile Generation Results

To generate profiles, the results generated by the classifier are used. *BoosTexter* generates predictions in which the most related class has the highest weight and the least related class have the least weight. Following the notation mentioned in Section V, a prediction vector generated by the classification algorithm is given by $\mathcal{H}(x) = (\mathcal{H}_1(x),\cdots,\mathcal{H}_c(x))$ where c is the number of classes for each document $x \in \mathcal{X}$; $\mathcal{X}$ being the set of all

documents. In the prediction vector, $\mathcal{H}_i(x) > 0, (i = 1, \cdots, c)$ will mean that the example belongs to class i. Further, $\mathcal{H}_j(x) > \mathcal{H}_i(x) > 0, (i = 1, \cdots, c), (j = 1, \cdots, c), (i \neq j)$ will mean that the example is more related to class j than to class i. Futher, classes that do not correspond to this particular example have negative weights.

Our method of profile generation from predictions constitute the following steps:

Step 1:Choose all the positive coordinates in $\mathcal{H}(x)$, that is, all $\mathcal{H}_i(x) > 0, (i = 1, \cdots, c)$.

Step 2:Arrange them in a descending order.

Step 3:Generate complete profile involving the classes related to these positive coordinates.

We present below some results from profile generation experiments. In *Profile 1*, we present an example on ***Hyperlink Based (*$\mathcal{L}$*)*** dataset. Similar profiles can be generated from ***Hyperlink Suspected Component Based (*$\mathcal{L}_{sus}$*)*** and ***Hyperlink Template Based (*$\mathcal{L}_{temp}$*)*** datasets. Weights for different classes as generated by the classifier are also presented. Moreover, we also provide our interprerations of this profile.

```
Profile 1: An Example of Profiling
           on Hyperlink Based Dataset
Example ID: 1146556342.16183_1
Features (Hyperlink):
   http://www3.netbank.commbank.
      common-site.net/netbank/bankmain/
Classes (structure and whois):
  textcontent=0, vlinks=1, htmlcontent=1,
  script=0, table=0, image=0, hyperlink=1,
  form=0, faketags=0, Hosted_Site=1,
  Legitimate_Site_Addition=1
Profile Generated:
   vlinks(0.030) htmlcontent(0.018)
   hyperlink(0.016) Hosted_Site(0.014)
   Legitimate_Site_Addition(0.012)
```

In the above profile, the hyperlink present in an email is taken as the feature. The classes supplied are the structural classes of the emails as well as the whois classes. The profile is generated as prediction from the algorithm. *Profile 1* shows us that the presence of hyperlink

```
http://www3.netbank.commbank.common-site.net/
          netbank/bankmain/,
```

the directory structure

```
netbank/bankmain/
```

or template

```
www3.netbank.commbank.common-site.net,
netbank, bankmain
```

has a correlation with the presence of visible links in emails. The visible links are used to disguise hyperlinks, which might exist as a hidden link within the html file. This is also shown by the positive value of 'hyperlink' class. Further, the email is directed to html compliant browsers, since only 'htmlcontent' is present. Moreover, hosting of the site is on a different domain and is superimposed on a legitimate site *http://www3.netbank. commbank.com* as is evident from 'Hosted_Site' and 'Legitimate_Site_Addition' classes. Further, presence of 'vlinks' that is visible links in the email has the highest weight which means that the hosted site has a link that is a spoofed link that is usually linked to another site.

Moreover, using weights generated by the classifer, it can be said that 'vlinks' and 'htmlcontent' are the most important classes enhancing the fact that this hyperlink correlates with presence of visible links in emails (for a disguise) and with html based emails only. Further, classes like 'Hosted_Site' and 'Legitimate_Site_Addition' having similar weights in most datasets bears evidence to the fact that the phishing site is a hosted site and to fool the user it is generated as being an addition to a legitimate site. Further some specific details can also be derived. It can be seen from *Profile 1* above, that this directory structure within a hosted server can be regarded as suspected one since the phisher uses items from this directory for phishing operation. It is also possible that files in this directory take the user to another location, but safely this can be regarded as the primary destination. There is no presence of other structures in the email for retrieving information like forms or scripts. The phisher(s) rely on tranferring the unsuspected user to the hoax site and extract information.

To summarize, the above results it can be stated that an email containing the hyperlink:

```
http://www3.netbank.commbank.common-site.net/
          netbank/bankmain/
```

is a phsihing email which has its site hosted on a domain that is an addition to a legitimate domain to fool an unsuspected user into thinking that it is from the legitimate domain. Further, the directory structure within the hosted server would be of

```
netbank/bankmain/
```

type. This directory structure will further try to convince the unsuspected user into thinking that it is definitely the legitimate site. The phishers who hosted this site targets html-compliant browsers. They do not seem to be have pages for text-based browsers. Their mode of operation is not using scripting or entry of details into an embedded form within the email but to lure the unsuspected victim to a fake site using the hyperlink. Thus, more knowledge is obtained from a hyperlink by using different datasets and by summarizing all profiles obtained.

VIII. Conclusion

In this paper, we have presented a novel method for obtaining profiles from phishing emails using hyperlink information as features and structural and whois information as classes. We have transformed the problem of profiling into a multilabel classification problem in which profiles are generated based on the predictions of the classifier. We have used a well-known classification algorithm (*BoosTexter* and *SVM*) for our experiments. Further, we create three different datasets from the hyperlink information in emails and use four-fold cross-validation to generate our predictions. The results from *BoosTexter* provided very high classification accuracy, hence more accurate profiling was obtained. We have also provided prediction weights generated by the classifier that show the relative importance of the classes used in profile generation. In future, we would enhance this technique to bring in more prominent features and develop more representative classes for profiling. Also we would like to experiment with different classifiers and compare the profiles generated in the process. Further, we aim to achieve a valid criterion for measuring the importance of the classes present in profiling.

References

[1] J. Stewart, "DNS cache poisoning - the next generation," Secure Works, Tech. Rep., 2003, http://www.secureworks.com/research/articles/.

[2] A. Emigh, "Online identity theft: Phishing technilogy, chokepoints and countermeasures," Radix Labs, Tech. Rep., 2005, retrieved from Anti-Phishing Working Group: http://www.antiphishing.org/resources.html.

[3] I. Fette, N. Sadeh, and A. Tomasic, "Learning to detect phishing emails," in *WWW '07: Proceedings of the 16th international conference on the World Wide Web*. New York, NY, USA: ACM Press, 2007, pp. 649–656.

[4] M. Wu, R. Miller, and G. Little, "Preventing phishing attacks by revealing user intentions," *Symposium on Usable Privacy and Security (SOUPS)*, 2006.

[5] A. Juels, M. Jakobsson, and T. N. Jagatic, "Cache cookies for browser authentication (extended abstract)," in *SP '06: Proceedings of the 2006 IEEE Symposium on Security and Privacy (S&P'06)*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 301–305.

[6] M. Chandrasekaran, K. Karayanan, and S. Upadhyaya, "Towards phishing e-mail detection based on their structural properties," in *Proceedings of the New York State Cyber Security Conference*, 2006.

[7] D. Chau, "Prototyping a lightweight trust architecture to fight phishing," MIT ComputerScience And Artificial Intelligence Laboratory, Tech. Rep., May 2005, final Report, http://groups.csail.mit.edu/cis/crypto/projects/ antiphishing/.

[8] M. Jakobsson and A. Young, "Distributed phishing attacks," Cryptology ePrint Archive, Report 2005/091, 2005, http://eprint.iacr.org/.

[9] R. Clark, "Profiling: A hidden challenge to the regulation of data surveillance," *Journal of Law and Information Science*, vol. 4, no. 2, 1993.

[10] "Market basket analysis," Wikipedia, January 2007, retrieved July 15, 2007 from http://en.wikipedia.org/wiki/Market-Basket-Analysis.html.

[11] D. Petrovic, "Analysis of consumer behaviour online," Analogik.com, Tech. Rep., 2007, retrieved July 15, 2007 from http://analogik.com/article-analysis-of-consumer-behaviour-online.asp.

[12] "Investor profiles," The National Mutual Life Association of Australasia Limited, retrieved July 20, 2007 from http://www.axafreedom.com.au/freedom/freedom.nsf/content/InvestorProfiles.

[13] "Interactive investor profile tool," Southside Bank: Trust & Investment Services Group, retrieved July 20, 2007 from http://www.southsidetrust.com/tool.htm.

[14] "Offender profiling," Wikipedia, November 2006, retrieved July 26, 2007 from http://en.wikipedia.org/wiki/Offender-profiling.html.

[15] "FBI method of profiling," Wikipedia, January 2006, retrieved July 26, 2007 from http://en.wikipedia.org/wiki/FBI-Method-of-Profiling.html.

[16] D. Webb, "A free and comprehensive guide to the world of forensic psychology," All About Forensic Psychology, retrieved July 28, 2007 from http://www.all-about-forensic-psychology.com/criminal-profiling.html.

[17] L. Alison, M. Smith, O. Eastman, and L. Rainbow, "Toulmin's philosophy of argument and its relevance to offender profiling," *Psychology, Crime and Law*, vol. 9, no. 2, pp. 173–183, June 2003.

[18] T. Castle and C. Hensley, "Serial killers with military experience: Applying learning theory to serial murder," *International Journal of Offender Therapy and Comparative Criminology*, pp. 453–465, 2002.

[19] "Customer profiling survey solution enabling cross and up selling," Confirmit: Customer Survey, 2007, retrieved July 29, 2007 from http://www.confirmit.com/solutions/survey/ customer%5Fprofiling/.

[20] "InterNIC : Whois search," InterNIC - Public information Regarding Internel Domain Name Registration Services, http://www.internic.net/whois.html.

[21] "The APNIC whois," Asia Pacific Network Information Center, http://wq.apnic.net/apnic-bin/whois.pl.

[22] "Ripe database," RIPE Network Coordination Centre, http://www.ripe.net/index.html.

[23] D. Webb, J. Yearwood, P. Vamplew, M. Liping, B. Ofoghi, and A. Kelarev, "Applying clustering and ensemble clustering approaches to phishing profiling," in *Proceedings of AusDM 2009, The Australasian Data Mining Conference 2009*. CRPIT, 2009.

[24] R. Schapire and Y. Singer, "BoosTexter: A boosting-based system for text categorization," *Machine Learning*, vol. 39(2/3), pp. 135–168, 2000.

[25] Y. Freund and R. Schapire, "A short introduction to boosting," *Journal of Japanese Society for Artificial Intelligence*, vol. 14(5), September 1999.

[26] T. J. Joachims, *Learning to classify text using support vector machines: methods, theory and algorithms*. Kluwer Academic Publishers, 2002.