



An RDF Based Anti-Phishing Framework

Vamsee Krishna Kiran M¹, Gowtham R², Archanaa R³
Computer Science and Engineering, Amrita University,
Coimbatore, Tamil Nadu-641112, INDIA

Abstract: Phishing is a fraudulent scheme to acquire sensitive information from users by masking as a trustworthy entity in an electronic communication. These attacks are increasing in frequency and have become more sophisticated and inevitable in the present day web. Though there are many existing anti-phishing solutions, attackers succeed to lure victims. This gives us a scope to develop a novel Anti-Phishing approach that overcomes the difficulty of detecting phishing websites. The Anti-Phishing method explained here explores the Resource Description Framework (RDF) data model to annotate the context sensitive features of the webpage. The semantic features of the page are collected from its HTML source and external information repositories in the internet. Two intact RDFs are generated for the suspicious webpage as well as webpage which is retrieved from search engine results; those are compared to check its equality to determine the legitimacy of the suspicious webpage and this comparison results are analysed to predict its target page when the suspicious page is being a phishing webpage.

Keywords: Anti-phishing, target identification, RDF, meta data, semantic web, RDF schema, suspicious, phishing

I. Introduction

Victims of phishing scams always find their personal or financial information stolen. Credit card numbers, health information, e-mail address login credentials, answers to security questions and other sensitive data are the targets of attackers. Once this information is attained, these details are used to create fake accounts on victim's name, ruin their credits or even prevent the users from accessing their own accounts. In the first half of 2012, the RSA (Anti-Fraud Command Centre) [1] identified 195,487 distinct phishing attacks which is an increase of 19% as compared to the second half of 2011. The total number of phishing attacks launched in 2012 was 59% higher than 2011. As of January, 2013, the RSA [2] has shut down more than 770,000 phishing attacks in more than 180 countries. According to research firm Gartner, number of phishing attacks will grow to over 185 billion by 2015 [4].

Any phishing attack usually involves two stages. First, a fake website which looks similar to a legitimate website is created. Then the users are motivated to visit this website instead of the original one where the authentication details can be extracted from the user without his knowledge. To counter these phishing attacks, possible counter measures have been developed by the researchers, most of the countermeasures would use any of the categories such as heuristic approaches, blacklist based methods, white list based methods or hybrid approaches. However these approaches alone are not sufficient to defeat phishing attacks, it also requires multifaceted or a novel mechanism to effectively combat phishing attacks [3].

Unfortunately, none of the aforesaid technical mechanisms completely solve this problem. This clearly indicates the need for a robust anti-phishing technique which needs to identify the legitimacy of the webpage by understanding its semantics, from its metadata and the way in which it is related to other web pages. This gives us an opportunity to develop a novel method that uses semantic web concept to detect phishing web pages. Resource Description Framework (RDF) [5] is intended for representing metadata about web resources. It provides a framework for expressing the information about the structural features and identities of the web pages. Semantic features are the characteristics of a webpage which are represented in RDF without losing its meaning. This annotated semantic mark-up representation helps the programs to understand and differentiate a webpage from other web pages.

In this paper, we proposed a novel approach which makes use of semantic features of a webpage to detect the phishing web pages and its phishing targets. Phishing target is the legitimate webpage whose look and feel is being imitated by the phisher. In this process, we take the webpage under scrutiny and extract metadata which are then represented in the form of RDF. In the similar way RDFs are constructed for web pages that are obtained as the search results, when given keywords from the suspicious page as input. On comparing both the RDFs, we find the authenticity of the suspicious webpage, and this comparison results are analysed to identify the possible target site(s). As our approach depends only on content of the suspicious webpage it requires neither a prior knowledge about the site nor requires the training data. Moreover our method also detects zero-day phishing attacks.

The rest of the paper is organized as follows: Section 2 presents an overview of literature review and related work. Section 3 illustrates the system overview. In section 4 we have explained metadata extraction from a suspicious webpage, RDF construction, phishing detection and target identification procedure. In section 5 we have given the implementation details. Evaluation metrics and experimental results are discussed in Section 6. Finally, conclusion is presented in Section 7 followed by references.

II. Literature Survey

Blacklisting and white listing approaches are most pre-dominant and widely used solutions in the modern day browsers. The white list approach maintains a list of all safe web sites and its associated information. Any website that does not appear in the list is recognized as malicious website. The current white list tools usually use a universal white list where all legitimate web sites are required to be included in the white list. Whereas, Han et al. [6] developed an approach to maintain individual white list which records the well-known legitimate web sites of the user rather than maintaining universal legitimate sites list. But AIWL (Automated Individual White-List) warns the user when the submitted account information does not match with the entry in the white list. In our approach, there are no login problems and we are using third-party DNS lookup to mitigate Pharming attacks.

The blacklist approach maintains a list of known phishing sites to check currently visiting websites against the list. This blacklist is usually gathered from multiple data sources like spam traps or identified by spam filters, user posts (e.g. PhishTank [28]), or verified phish compiled by other parties such as takedown vendors or financial institutions. For example, Prakash et al. [7] used approximate matching algorithm that divides a URL into multiple components that are matched individually against entries in the blacklist. Zhang et al. [8] proposed a system where customized blacklists are provided for the individuals who choose to contribute data to a centralized log-sharing infrastructure. This individual blacklist is generated by combining relevance ranking score and the severity score generated for each contributor. But blacklist needs frequent updates from their sources, when the blacklist isn't updated properly, it is impossible for the tools to identify phishing websites. Moreover the exponential growth of list needs great deal of system resources. In our approach we are not depending on any predefined data sources to check the legitimacy of the suspicious web pages. As our approach checks the legitimacy of the webpage by analysing its characteristics which in turn helps us to identify zero day phishing attacks.

Heuristic based approaches inspect the characteristics of a web site to predict its legitimacy. Chou et al. [9] developed the browser plug-in Spoof Guard that identify phishing web pages based on series of heuristics. These heuristics are grouped into stateless evaluations that identify suspiciousness of a web page which is downloaded in the web browser and stateful evaluations that evaluate the web page credibility through its recent visits by the user (user's history file). This approach mainly suffers from false alarm when the user visits any login page with same username and password entry in the database, whereas in our method we neither record private details of user nor suffer from false alarms. Fette et al. [10] proposed a method for detecting phishing emails by incorporating features specific to phishing. They propose ten different features to identify phishing email. Eight of these features can be extracted from the email itself, while the other two features, the age of linked-to domain names has to be obtained by a WHOIS query at the time the email is received and spam-filter output feature incorporate the class assigned to the email. The features used in this method are not adequate to detect pharming attack and pages hosted on compromised domain.

Zhang et al. [11] proposed a content-based approach CANTINA, based on the TF-IDF (term frequency and inverse document frequency) algorithm to identify top ranking keywords from the page content and meta keywords/description tags. These keywords are given as input to trusted search engine such as Google. Here, a webpage is considered legitimate if the page domain appears in the top N search results. The heuristics used in this method are primarily adopted from Chou et al. and Fette et al. work. CANTINA+ [12] is an upgraded version of CANTINA proposed by Xiang et al., where new components are included to achieve better results. Particularly, they have included ten other features along with four of the CANTINA features and one extended feature. Both of these methods are falling short in detecting phishing webpage containing images and scripts, as they have no text in the page to analyse. And also, they fail to detect page consisting of bad forms, bad action fields or non-matching URLs.

Joshi et al. [13] developed PhishGuard tool that identifies phishing websites by submitting actual credentials after the bogus credentials during the login process of a website. They also proposed architecture for analysing the responses from server against the submission of all those credentials to determine if the website is legitimate or a phish. Chuan et al. [14] designed a component BogusBiter that submits a large number of bogus credentials along with the actual credential of users to nullify the attack. A similar approach has been applied by Joshi et al. but BogusBiter is triggered only when a login page is classified as a phishing page by a browser's built-in detection component. These two approaches fail to detect legitimacy of the pages having self-signed certificate or pages with TLS or SSL certificate.

Fu et al. [15] proposed an approach which uses Earth Mover's Distance (EMD) to measure Web page visual similarity. In this approach, they first convert the involved web pages into low resolution images and then use colour and coordinate features to represent the image signatures. EMD is used to calculate the signature distances of the images of the Web pages. They used trained EMD threshold vector for classifying a webpage as a phish or legitimate. Medvet et al. [16] proposed an approach which identifies phishing web pages, by considering text pieces and their style, images embedded in the page and the overall visual appearance features of the web page. Chen et al. [17] present an image based anti-phishing system, which is built on discriminative key point

features in Web pages. Their invariant content descriptor and the Contrast Context Histogram (CCH) compute the similarity degree between suspicious and legitimate pages. Chen et al. [18] also proposed an approach which uses Gestalt theory for detecting visual similarity between two web pages. They used the concept of super signals to treat the web pages as indivisible units; these indivisible super signals are compared using algorithmic complexity theory. But these techniques results in false positive when legitimate page cross the similarity threshold value.

Hossain Shahriar and Mohammad Zulkernine [19] proposed a model to test a suspected phishing websites based on trustworthiness testing approach. In trustworthiness testing, they check behaviour (response) of websites matches with known behaviour of a phishing or legitimate website to decide whether a website is phishing or legitimate. This model is explained using the notion of Finite State Machine (FSM) that captures the submission of forms with random inputs and the corresponding responses to describe the website's behaviour. This approach can detect advanced XSS-based attacks that many contemporary tools currently fail to detect, but it fails when there are embedded objects like flash as this approach is based on responses due to form submissions. Our approach works fine even if there are embedded objects in the webpage as we identify phishing page from the hyperlinks and source of objects present in the websites and do not rely only on the contents of the webpage.

Wenyin et al. [20] proposed an approach to identify legitimacy of a given suspicious webpage and discovering its phishing target by calculating and reasoning defined association relations on its Semantic Link Network (SLN). Wenyin et al. [21] also proposed an anti-phishing approach that identifies phishing targets using a suspicious webpage's associated relationships. A Web graph is constructed from the associated pages and further partition of the graph results in a Web community (the parasitic community) for the given webpage. Parasitic coefficient is used to measure the parasitic relationship's strength from the given page to each page in the community. The page with the strongest parasitic relationship to the given suspicious webpage is regarded as the phishing target. If such target is found, they identify the given suspicious webpage as a phishing webpage. Otherwise, they consider it as legitimate. Here, the computational cost of building web community, partitioning the web graph and identifying the phishing target is expensive.

Our work is also motivated by several researches that developed methods, which help us to check webpage's legitimacy by understanding its content. We discuss these approaches in brief now.

Debajyothi et al. [22] proposed an approach that automatically translates a natural language text in a webpage to RDF, and stores it in a manner that is understandable by the system for the semantic web that are usually large ontologies. This paper implements some pre-processing steps like tokenizing, POS tagging, normalizing the text etc. for extracting the structured information from the web page. This structured information is then represented in RDF using the algorithm that is defined in the paper.

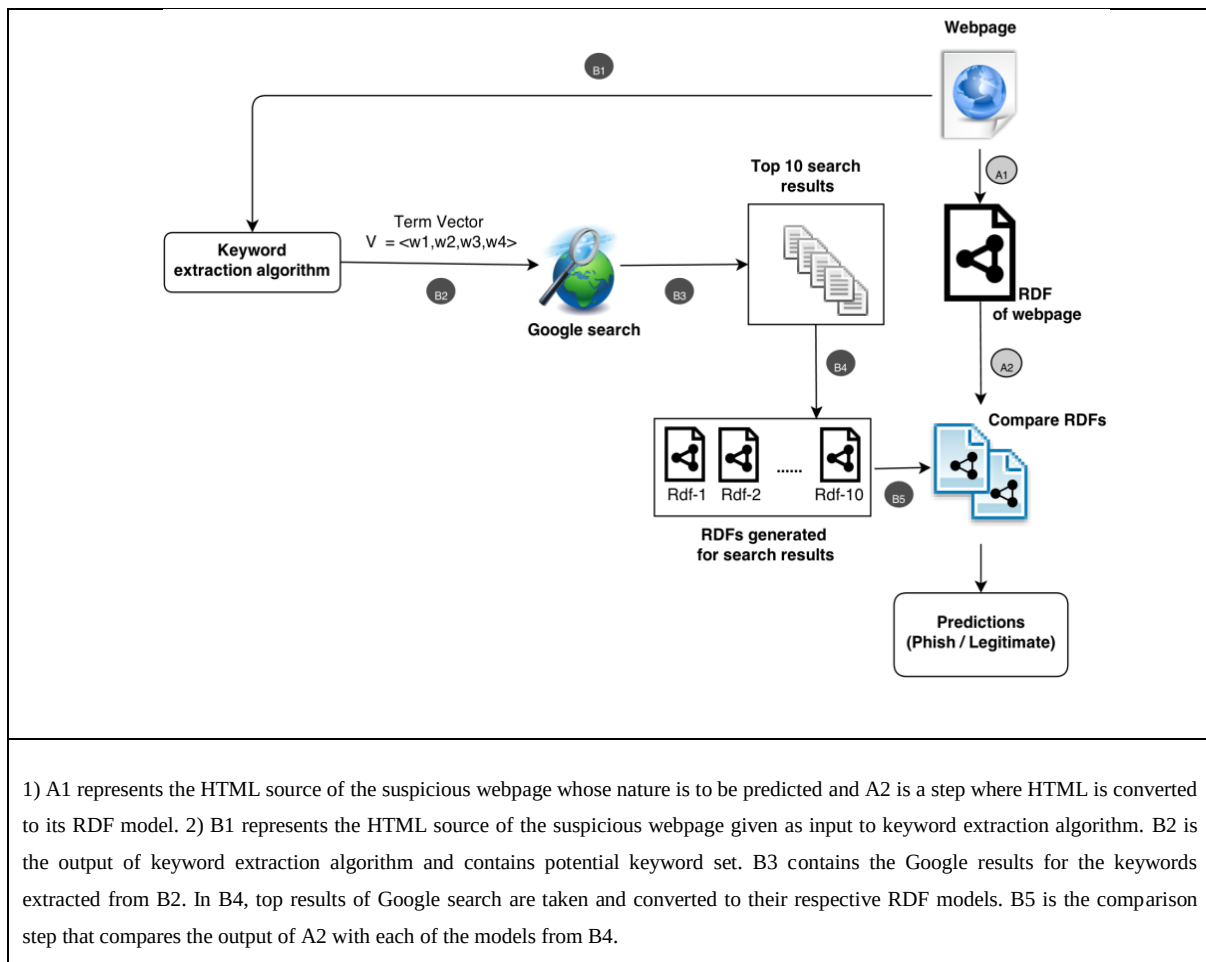
Charlotte et al. [23] proposed a method that automatically classifies HTML document according to Dewey Decimal Classification (DDC). This automatic classifier extracts context sensitive information from the HTML document which is used to analyse and assign the appropriate DDC class marks. The classifier outputs the element set in RDF syntax as defined by the RDF schema which is adopted from WC (Wolverhampton Core) and DC (Dublin Core metadata) vocabularies. This paper mainly focuses in classifying the documents based on the domain they belong to. This paper gives the fundamental knowledge for extracting the features and representing them in RDF model.

Faisal et al. [24] developed a RDF based phishing detection method, it inspects legitimacy of the suspicious webpage by extracting RDF features from the given bank's web page and checks it with predefined RDF knowledge base maintained at the centralized server. To do this, each bank's profile is maintained at the centralized server. Profile information contains details like the bank's URL, bank name, branch name, allowed ports, allowed IP addresses etc. This basic information about the bank is maintained as ontology in the database for future comparisons. But, the drawback of this approach is, it detects phishing websites that aims at only banks. Also, this method possesses practical issues.

III. Architecture

Our system identifies phishing webpage and its target by comparing the metadata of the suspicious page and the pages indirectly associated with suspicious webpage. Metadata that we consider here are represented in the form of RDF document. The metadata element set is selected in such a way that no two RDF documents are same for two different web pages. This RDF representation of web pages eases the process of phishing identification and also helps the system to better understand the web document in phishing target detection.

The entire system can be viewed as a three step process as shown in Figure-1. For a given suspicious webpage, the system first extracts the metadata from its source code and external information repositories, and constructs an RDF document by considering the data model. Then, in the similar way RDF documents are generated for web

Figure 1: System design

pages that are indirectly associated with the suspicious webpage. These indirectly associated web pages are retrieved from search engine result, by inputting the keywords those are extracted from the suspicious page. Here, we consider first 'n' results of search engine in the same order in which it has returned. The RDFs generated in this step are compared with the RDF of suspicious page to check its legitimacy. If any of this comparison results is isomorphic then the suspicious webpage would be considered as legitimate, otherwise it is considered as phishing and the differences are recorded. Finally, the differences of the results compared are inputted to Target Identification Algorithm (TIA) to detect its phishing target, this step is carried out only when the suspicious webpage is identified as phishing.

IV. RDF Construction and Phishing Detection

Construction of RDF documents for the web pages involves extracting some set of features and representing these features in a RDF model using the RDF schema defined for our data model. We have chosen 19 properties of webpage which can be used to differentiate between the phishing and legitimate web pages and also used to identify its phishing target.

A. Metadata element set

Metadata element set contains the vocabulary of 19 properties to describe the characteristics of the webpage as shown in Table-1. These elements are classified into three types: structural, descriptive and behavioural.

HTML to RDF depends on a pre-processing step. This pre-processing step involves identifying the feature set that is to be extracted from the web page. Once we have a defined feature set that can identify a web page uniquely, we construct the RDF model of the web page that imbibes those features.

We already have some defined RDF vocabularies like Dublin Core, XHTML and HTTP vocabularies. We combine these vocabularies and use to define the RDF models. RDF gives the user a scope for defining his own vocabulary under a defined namespace. We will have a new defined vocabulary for some features and we combine those with the existing ones. All the features are represented as RDF properties each represented using a triplet (subject, predicate and object). The feature set chosen is given as follows:

Table 1: Shows the Element set for RDF construction of the web page.

S.No	Properties	Description	Source
1	Title	Name or Title of the given web page.	Dublin Core
2	Format	Give the character set format of the web page.	Dublin Core
3	Creator	An entity primarily responsible in creation of the web page. Examples of a Creator include a person or an organization.	Dublin Core
4	Subject	Typically, the subject will be represented using keywords, key phrases of the web page.	Dublin Core
5	Description	A brief description of the web page.	Dublin Core
6	Date	Gives the last modified date of the web page.	Dublin Core
7	Identifier	URI of the web page.	Dublin Core
8	Status code	Gives the status code number that is returned from HTTP response.	HTTP
9	Form method	Method of the form i.e. GET or POST	XHTML
10	Form name	Name given to the form element	XHTML
11	Form action	Action part of the form element	(OWN)
12	Imgsrc	Gets the Source of the images	XHTML
13	Script src	Gets the source of the Script, if the script is written explicitly	(OWN)
14	Copyright	Gives the copyright information of the web page.	(OWN)
15	Bytes	Gives the web page's source file size in bytes.	(OWN)
16	Captcha	Check for captcha in the web page	(OWN)
17	Frame src	Gets the source of the frames in web page	(OWN)
18	IP address	Gets the domains IP addresses of the web page	(OWN)
19	Parent Domain	Gets the parent domain of the web page	(OWN)

The above mentioned 19 elements are considered after doing a comparison study on different web sites from different domains. These elements are considered by performing observations on the source codes of nearly 620 web sites and a manual study on the source codes. In a web page most of the content remains static and is made up of static HTML elements. There are some elements that are dynamic and whose sources can be modified by the attacker. Observations prove that identifying these set of elements will be sufficient. The element set provided above will have some standard elements of the web page and also the elements that can be easily modified by the attacker. We have also evaluated other set of elements from the vocabularies mentioned. Observations found that other elements are not suitable to our problem statement and cannot help with phishing detection.

In the feature set given above, 7 elements are taken from the Dublin Core [25] where a set of 15 elements are mentioned. We have observed that the other 8 elements of the DC are not in wide use, therefore unnecessary to consider. Few elements like form, image sources, status code are taken from XHTML and HTTP vocabularies [29][30]. Elements like Identifier, Form action, IP address, Domain, Bytes, Java script and Image sources etc. will be unique. So, we consider the combination of these features which will help us in verifying the legitimacy of a webpage. The features extracted from webpage are represented in the form of RDF model. An RDF model is a triple containing subject, predicate and objects. Each and every statement is represented in the form of a triple making the logical assertions to be simple.

B. RDF Schema Generation

Once the required properties are identified, a schema must be created. We can also make use of existing schemas where these properties are already defined. RDF schema defines the structure by which RDF statements are interpreted. It is possible that we can define or reference as many schemas as we want. These schemas are referenced using the namespace mechanism. Defining new schemas enables the developers to define their own properties that are suitable to their respective applications. Existing schemas can be directly referenced from the RDF/XML document. If developers need a new schema for their application, then there are several steps involved in constructing the user defined schemas.

First, the required elements are to be identified. Next, check for any existing vocabularies where the elements are already defined. If few of your elements are already defined, then either you can reference to that vocabulary

directly from your RDF/XML document or you can use a property-sub property relationship between your element and the element that is already defined. Identify what elements have to be defined as classes, subclasses, properties and sub properties etc. Define the elements accordingly and write the RDF schema for all the elements. RDF schema written has to follow some standards mentioned in the RDF schema specification [26]. The RDF schema for our application is given in the Appendix A and the respective RDF generated for a web page is given in Appendix B.

C. Keyword Extraction

In the proposed approach, we need to extract a keyword vector from the given suspicious page and input that vector to Google search engine. As the main area of focus is on defining the mechanism that can identify the zero day phishing attacks, we will not discuss much on the keyword extraction algorithm as it is a separate domain. However, we discuss the methodology we have used in extracting the keywords from the web page. The keyword extraction done in this paper mainly extracts the words from the title and meta tags. Some websites contain the meta tag for keywords. We used this tag for extraction. On the other hand, some web sites may not contain meta tags as this is purely optional. In such scenario, we extract the words from the body of the web page. As we get huge number of words from the body of the page, we implement stop word removal mechanism, so as to cut down the list of words and to have only potential keywords. We use the standard stop word list given in the github [32] to process our data. Using better keyword extraction algorithms like Google similarity distance [33] can improve our approach by retrieving the most potential words from the web page. This approach will get the expected results in the top search results. The evaluation of this keyword extraction algorithm is given in their paper. We will not discuss much on this keyword extraction as it is beyond the scope of this paper and a separate research area.

D. Phishing Detection

After constructing the RDF model for the suspicious page, we compare this model with the RDF models constructed for each web page that is obtained as the Google top results. We define a comparison strategy that employs few functionalities of the Apache Jena framework [27].

The prediction strategy is divided into two steps. In the first step, we compare the RDF's using the Isomorphic function of the Apache Jena. Isomorphic function takes two models as the input and compares. Two models are isomorphic when each statement in one can be matched with a statement in the other. This case helps us to reduce the running time of our approach. Isomorphic and Equals methods of Apache Jena library are clearly explained in a document called "Matching RDF Graphs" by HP [31]. This document explores the equality of two RDF graphs in light of the graph isomorphism literature. This document shows how the standard graph isomorphism algorithms can be used effectively for comparing RDF graphs. We considered the theory mentioned in this document and used the same for identifying and comparing the anonymous nodes. The comparison methodology makes use of this theory.

This method can be explained using an example as follows: Let our suspicious page be a legitimate page and we extract the keywords from that page and give it to Google search to obtain the top results. It is more likely that the same page will be there in the top 10 results obtained from the Google. In this case, if we construct the RDF's for the pages retrieved from the Google, both the RDF's i.e. RDF of our suspicious page and RDF of the same page resulting from the Google results will be same. Therefore when we compare these RDF's with the isomorphic method it results in a perfect match. If we can find the match, then we can directly predict the status of the page. But, there are some cases where a page will be a legitimate page but then will not match with any of the results. This is a possible case when our keywords are not so efficient and couldn't retrieve the intended page in the top results. Here, to avoid the false positives we go for the second step that takes care of such cases where a legitimate page can be predicted as a phishing page when the isomorphic case fails.

In the second step, we perform comparison on a subset that is taken from the main element set defined above in this paper. We take this subset in such a way that it is not dependent on purely the content of the web page. Therefore, even if the isomorphic case fails, these subset comparison methods should be able to reduce the false positives. The subset elements considered are IP address, Status code and Copyright.

The reason for choosing these three elements in the step 2 comparison is, imagine a case where our keyword extraction algorithm may not retrieve the exact page that we are looking for, but we get a page that is from the same website. The three features that we have taken will remain same for all the pages from the same website. This means all the pages from the website are launched from the same IP address, and will have same copyright and status code information. This will help us to identify the pages as possibly legitimate instead of identifying them as phishing.

In order to explain the importance of step two comparisons, let us consider an example where a legitimate page is having two identifiers. Ex: <http://www.paypal.com> and <https://www.paypal.com/in/webapps/mpp/home> both refers to the same page. If we depend only on the isomorphic case it may identify this URL as a phishing one. So, we compare using the second step. Here second step comparison gives the match, since IP addresses will match as both are from the same parent domain i.e. paypal.com. Also, status code will match as both the pages are from same domain and online (available). So, they both return status code 200. Also, we observed that the copyright information will be same for pages within the same domain. This comparison of the subset gives a match; therefore we can possibly consider the given URL as legitimate instead of predicting the page as phishing.

Given a suspicious web page, system behaviour is given as follows:

Let the suspicious page be Phishing page:

In the first case, Isomorphic verification will not match any of the results from the Google search since, Google relies on page ranking algorithm and, no phishing page will have a better rank to come up in the top 10 results of Google search. In the second step, comparing resources (IP address, Status code and copyright) for a phishing page and a legitimate page will result in change of at least one property. Since we say that the complete set has to satisfy, even if one property doesn't satisfy we conclude that we didn't find a match.

Since, both the cases fails, the page is predicted as a phishing page.

On the other hand, there may be some phishing pages implementing DNS spoofing technique. In such scenarios, we may get some false negatives as both the step 1 and step 2 comparisons may fail. But, false negatives will be very few in number for this approach which are almost negligible as shown in the results.

Let the suspicious page be legitimate page:

After supplying the keywords, if the Google search is getting the exact page in the top results, then Isomorphic method finds an exact match with the suspicious page. If an exact match is found, we say that page is legitimate. If not, we adopt second step. Here even if the intended page is not shown in the top 10 results of Google, most of the results will be from the same domain. So, we compare the IP address, Status code and copyright of the pages. If all the three matches with any of the result, we can say that the page is possibly legitimate.

Hence, the page is predicted as a legitimate page.

E. Target Discovery

In this paper, we also identify the possible target domain. We also believe that if the keyword extraction algorithm is a better one, then our approach can even identify the target page which none of the existing approaches are doing. Our identification of target is dependent on our feature set.

In phishing, the attacker tries to imitate the original page to provide the same look and feel as a legitimate page. For this, the attacker takes most of the content from the legitimate page and changes only few areas of the page which are important for capturing the confidential information of the users and send it to the attackers database. For example an attacker may copy the same content of a legitimate page into his site changing only the form action tag such that the confidential details of the form are saved into attackers false database. In some cases the attacker may keep everything same changing only the script source, where a false script gets executed instead of the intended one. This kind of behaviour can be detected using our element set.

V. Target Identification Algorithm (TIA)

- (1) Construct the RDF model of the suspicious web page
- (2) Extract the keywords from the suspicious page and give it to Google
- (3) Since the keywords are site specific, most of the results obtained from Google will be from the same domain. This gives us a chance to identify the targeted domain.
- (4) In order to identify the targeted page we employ comparison that is based on the Equals method obtained from the Apache Jena framework
- (5) Compare this RDF model of suspicious page with the RDF models of the Google search results.
- (6) Equals method compares each and every resource of the suspicious page with the pages from the Google search.
- (7) While comparing we maintain a count value that indicates how many resources have mismatched between two pages.
- (8) The page from the Google results that has the minimum number of mismatch value will be our possible target page.

Here, in this approach we maintain a count value and check for the minimum of it. We choose minimum as our criteria since, the attacker will imitate a legitimate page. This means most of the content will be the same and only few properties like action tags, script tags etc. will be changed. Therefore, the page that has minimum number of mismatches will be our target page.

VI. Implementation

We have implemented our method on a standalone system which can be deployed either as a service or we can integrate into web browsers as a light weight plug-in to alert users to phishing attacks. We have used Jsoup parser for extracting information from the web page. We also used Google search API for getting the top most results for a given keyword vector. We have done some pre-processing steps like stop word removal and POS tagging etc.

so as to identify proper set of keywords. RDF schema generation, RDF model construction, comparing two RDF models for phishing detection and target discovery, all these are done using Apache Jena package. We have integrated all the packages and developed the system using Eclipse IDE.

VII. Evaluation

A. Metrics used in evaluation

The basic metrics used for the evaluation are as follows:

True Positives (TP): The number of correctly classified phishing pages, i.e. identifying the phishing page as phishing.

False Negatives (FN): The number of falsely classified phishing pages, i.e. identifying the phishing page as legitimate.

True Negatives (TN): The number of correctly classified legitimate pages, i.e. identifying the legitimate page as legitimate.

False Positives (FP): The number of falsely classified legitimate pages, i.e. identifying the legitimate pages as phishing.

We have also used three metrics to evaluate the performance of system, which are True Positive Rate (TPR), False Positive Rate (FPR) and Accuracy (ACC).

True Positive Rate (TPR) measures the percentage of correctly classified phishing sites. TPR is computed using equation 6.1.

$$TPR = \frac{TP}{P} = \frac{TP}{(TP + FN)} \quad (Eq. 6.1)$$

Where TP is the number of correctly classified phishing pages. P is the number of phishing pages, which is equivalent to sum of correctly classified phishes (TP) and falsely classified phishes (FN).

False Positive Rate (FPR) measures the percentage of legitimate sites wrongly classified as phishing. FPR is computed using equation 6.2.

$$FPR = \frac{FP}{L} = \frac{FP}{(FP + TN)} \quad (Eq. 6.2)$$

Here FP is the number of legitimate pages which are wrongly classified as phishing, L is the number of legitimate pages which is equivalent to sum of falsely classified legitimate pages (FP) and correctly classified legitimate pages (TN).

Accuracy (ACC) measures the degree of closeness between measurements of classified sites and sum of actual phishing sites and legitimate sites. ACC is computed using equation 6.3.

$$ACC = \frac{(TP + TN)}{(P + L)} \quad (Eq. 6.3)$$

Here accuracy value will be close to 1 for any ideal anti phishing system. Accuracy of the system can be improved by having higher TP value and lower FP value.

B. Description of data

We have collected a real world dataset of 4554 live phishing and legitimate websites over a period of 3 months from November 2012 to January 2013. Specifically, our dataset consists of 1200 legitimate pages and 3354 unique phishing pages. Legitimate pages in our dataset are obtained from three sources as shown in Table-2. Legitimate pages in our data set mainly focus on the popular and most targeted websites published in the sources.

Table 2: Legitimate data source

Source	Sites	Link
Google's top 1000 most-visited sites	650	http://www.google.com/adplanner/static/top1000/
Alexa's Top sites	340	http://www.alexa.com/topsites
Netcraft's Most Visited Sites	150	http://toolbar.netcraft.com/stats/topsites/
Millersmile's of top targeted sites	60	http://www.millersmiles.co.uk

Each entry in our phishing dataset is unique which are downloaded from two sources as shown in Table-3. Most of the phishing URL were not alive at the time of testing very few of them are active. Since, phishing web sites are very transient and short-lived. They remain active only for few hours to an average of 3 days. Some phishing URL's are moved through different hosts, with multiple take down and reactivations. Because of these factors, we downloaded the pages up to level one when they were alive.

Table 3: Phishing data source

Source	Sites	Link
Phishtank's open database [28]	2579	http://www.phishtank.com/developer_info.php
Reasonable-Phishing Web pages List	775	http://antiphishing.reasonables.com/BlackList.aspx

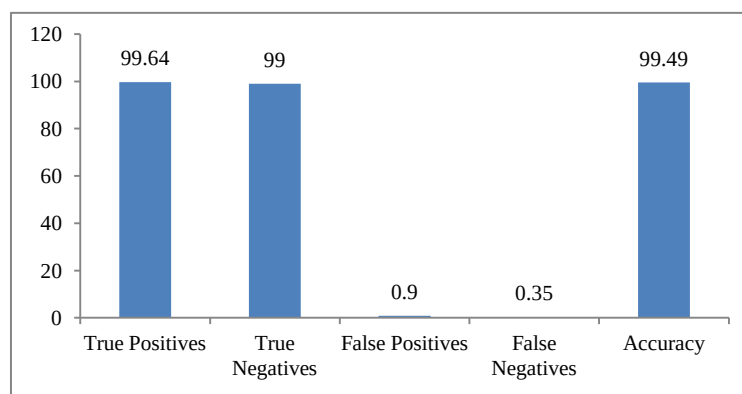
C. Detection Accuracy

The experiment results are shown in Table-4. The true positive rate of this method is 99.64%, false positive rate is 0.5% and accuracy is 99.49% as shown Figure-2. This statistics clearly shows that this system detects phishes with less false positives and high accuracy rate. Moreover, for all the successfully classified pages we have identified its target also.

Table 4: Experiment results: N is the total number of pages, n is the number of correctly classified pages

	Phishing pages	Legitimate Pages	Total
N	3354	1200	4554
n	3342	1189	4531

Figure 2: Assessment of experiment



VIII. Conclusion

This paper has presented a novel method for detecting the phishing web pages as well as their target. Here, we have identified the phishing pages in a three step process; in the first step we crawl the suspicious page and construct a RDF model for the element set chosen, also we give the keywords as input to the Search engine to get the most appropriate results. In the second step, we construct RDFs for the top results obtained from the search engine. In the final step, we go for a comparison methodology that classifies the suspicious web page and also identifies its target. Apart from phishing detection, this paper also gives clear picture on how to construct RDF for a web page where we have discussed about writing user defined schemas and generating the appropriate RDF models. Our approach purely depends on the content of the web page and doesn't require any prior knowledge about the site and will not make use of any training data.

References

- [1] RSA Anti-Fraud Command Center, RSA Monthly Online Fraud Report, January 2012: http://www.rsa.com/solutions/consumer_authentication/intelreport/11635_Online_Fraud_report_0112.pdf, accessed on December 8, 2012.
- [2] Brian Pennington, RSA's January Online Fraud Report 2013 including an excellent summary of Phishing in 2012: <http://brianpennington.co.uk/2013/01/30/rsas-january-online-fraud-report-2013-including-an-excellent-summary-of-phishing-in-2012/>, accessed on January 12, 2013.
- [3] Barrett, Michael, and Dan Levy, "A Practical Approach to Managing Phishing", PayPal Whitepaper, April (2008).
- [4] "Gartner Technical Report", <http://www.gartner.com/technology/research.jsp>, accessed on January 12, 2013.
- [5] Frank Manola, Eric Miller, "RDF Primer W3C Recommendation 10 February 2004", <http://www.w3.org/TR/rdf-primer/>, accessed on February 5, 2013.
- [6] Han.W, et al., "Using automated individual white-list to protect web digital identities". Expert Systems with Applications (2012), DOI:10.1016/j.eswa.2012.02.020.
- [7] Prakash, Pawan, Manish Kumar, RamanaRaoKompella, and Minaxi Gupta. "Phish-net: predictive blacklisting to detect phishing attacks." In INFOCOM, 2010 Proceedings IEEE, pp. 1-5.IEEE, 2010.
- [8] Jian Zhang, Phillip Porras and Johannes Ullrich, Highly predictive blacklisting, In proc. of the 17th conference on Security symposium, USENIX Association Berkeley, CA, USA, 2008, Pages 107-122.
- [9] N. Chou, R. Ledesma, Y. Teraguchi, D. Boneh, J. Mitchell : Client-side defence against web-based identify theft, in: Proc. of the 11th Annual Network and Distributed System Security Symposium, NDSS'04, San Diego, CA, February 2004, Vol. 380.

- [10] I. Fette, N. Sadeh, A. Tomasic : Learning to detect phishing emails, in Proc. Of the 16th Intl. Conf. on World Wide Web, Banff, Alberta, Canada, May 2007, pp. 649–656.
- [11] YueZhang , Jason I. Hong , Lorrie F. Cranor : CANTINA - a content-based approach to detecting phishing web sites, In Proc. of the 16th international conference on World Wide Web, Banff, Alberta, Canada, May 08-12, 2007, pp. 639-648
- [12] Xiang, G., Hong, J., Rose, C. P., and Cranor, L. : CANTINA+: A feature-rich machine learning framework for detecting phishing Web sites. *ACM Trans. Inf. Syst. Security*.14, 2, Article 21 (September 2011), 28 pages.
- [13] Joshi, Yogesh, Samir Saklikar, Debabrata Das, and SubirSaha. "PhishGuard: a browser plug-in for protection from phishing." In *Internet Multimedia Services Architecture and Applications*, 2008.IMSAA 2008. 2nd International Conference on, pp. 1-6. IEEE, 2008.
- [14] Yue, Chuan, and Haining Wang. "BogusBiter: A transparent protection against phishing attacks." *ACM Transactions on Internet Technology (TOIT)* 10, no. 2 (2010): 6.
- [15] Fu, a. Y, Wenyin, I., and Deng, X. 2006. Detecting phishing Web pages with visual similarity assessment based on earth mover's distance (EMD). *IEEE Trans. Depend. Secure Computing*. 3, 4, 301-311.
- [16] E. Medvet, E. Kirda, and C. Kruegel. Visual-similarity-based phishing detection. In *IEEE International Conference on Security and Privacy in Communication Networks*, Istanbul, Turkey, September 2008. IEEE Computer Society Press.
- [17] K.T. Chen, J.-Y.Chen, C.R. Huang, and C.-S. Chen, Fighting phishing with discriminative key point features, *IEEE Internet Computing*, vol. 13, no. 3, pp. 56-63, 2009.
- [18] Chen, T.C., Dick, S., and Miller, J. 2010.Detecting visually similar Web pages: Application to phishing detection.*ACM Trans. Intern.Tech.* 10, 2, Article 5 (May 2010), 38 pages.
- [19] Shahriar, Hossain, and Mohammad Zulkernine. "Trustworthiness testing of phishing websites: A behaviour model-based approach." *Future Generation Computer Systems* (2011).
- [20] Wenyin, Liu, Ning Fang, XiaojunQuan, Bite Qiu, and Gang Liu. "Discovering phishing target based on semantic link network." *Future Generation Computer Systems* 26, no. 3 (2010): 381-388.
- [21] Wenyin, Liu, Gang Liu, Bite Qiu, and XiaojunQuan."Anti-phishing through Phishing Target Discovery." *Internet Computing*, *IEEE* 16, no. 2 (2012): 52-61.
- [22] Debajyothi et al., "A New Semantic Web Services to Translate HTML pages to RDF", *10thInternational Conference on Information Technology, ICIT 2007 Proceedings; Rourkela, India; IEEE Computer Society Press*, California, USA; December 17-20, 2007; pp. 292-294.
- [23] Charlotte Jenkins, Mike Jackson, Peter Burden, Jon Wallis, "Automatic RDF metadata generation for resource discovery", *Elsevier Science* (1999) 227-242.
- [24] Faisal Alkhateeb, Ahmed M. Manasrah and Abed Al RaoofBsoul, "Bank Web Sites Phishing Detection and Notification System Based on Semantic Web technologies", *International Journal of Security and Its Applications*, Vol. 6, No. 4, October 2012.
- [25] Dublin core metadata initiative, <http://dublincore.org/documents/2012/06/14/dcmi-terms/?v=elements#>, accessed on February 4, 2013.
- [26] Resource Description Framework (RDF) Schema Specification 1.0, <http://www.w3.org/TR/2000/CR-rdf-schema-20000327/>, accessed on February 12, 2013.
- [27] Apache Jena Java framework, <http://jena.apache.org/>, accessed on February 12, 2013.
- [28] PhishTank database, <http://www.phishtank.com/>, accessed on December 20, 2012.
- [29] XHTML vocabulary, version date:2010-01-27, <http://www.w3.org/1999/xhtml/vocab#>, accessed on February 22, 2013.
- [30] HTTP vocabulary, <http://www.w3.org/2011/http#>, accessed on February 23, 2013.
- [31] Jeremy J. Carroll, "Matching RDF Graphs", *Information Infrastructure Laboratory, HP Laboratories Bristol, HPL-2001-293*, Dated: November 26, 2001.
- [32] Standard set of English stop words, <https://github.com/arc12/Text-Mining-Weak-Signals/wiki/Standard-set-of-english-stopwords>, accessed on February 5, 2013.
- [33] Chen, Ping-I., and Shi-Jen Lin. "Automatic keyword prediction using Google similarity distance." *Expert Systems with Applications* 37.3 (2010): 1928-1938.