Journal of Theoretical and Applied Information Technology
31st March 2018. Vol.96. No 6
© 2005 – ongoing JATIT & LLS

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

NEW PHISHING HYBRID DETECTION FRAMEWORK

[1]YOUNESS MOURTAJI, [2]MOHAMMED BOUHORMA, [3]DR. DANIYAL ALGHAZZAWI

[1]Faculty of Sciences and Techniques University, Computer science, systems and telecommunication Laboratory, Tangier, Morocco

[2]Professor, Faculty of Sciences and Techniques University, Computer science, systems and telecommunication Laboratory, Morocco

[3]Professor, King Abdulaziz University, Information Systems Department, Jeddah, Saudi Arabia

E-mail: [1]mourtaji.y@gmail.com, [2]bouhorma@gmail.com, [3]dghazzawi@kau.edu.sa

ABSTRACT

Internet use is growing every day, accessing a website via its URL (Uniform Resource Locator) address is a daily task, but not all websites are benign to be accessed without any fear from malicious aims - not matter where those websites are being accessed from (Web Browsers, e-mails body, chat application, SMS, VoIP) neither the nature of the operating system or the device. Our thesis aim is being able to detect the kind of websites that try to steal any user's (normal users, communities, societies, laboratories, etc.) personal information like name, date of birth, e-mail, credentials, login and passwords from e-banking services for example or any other web services. Unlike traditional techniques that consists of penetrating data sources of web services providers by decrypting algorithms, the man idea of this kind of criminal activities is letting the victims give those informations unconsciously, by creating fake emails or websites that looks very similar of original ones and tell victims to fill some forms with their informations for some fake reason, this technique is called phishing. This article aims to discuss some used techniques in detecting phishing websites, like Black-list based, Lexical based, Content based and Security and Identity based methods combined with some machine learning classifier to classify if a test URL is a safe or phishing website and to propose a new hybrid framework to detect phishing web pages from only their URL without need to access it visually with a browser. The data used for building the model and classification is a collection of active phishing websites gathered from PhishTank[1].

Keywords: *Machine Learning, Malicious URL Detection, Network Security Intelligence, Phishing, Smart Systems and Communication*

1. INTRODUCTION

Today we can do almost everything within internet, it becomes the base of banking transactions, shopping, entertainment, resource sharing, news, and social networking [2]. This growth of internet use has become also a primordial center of interest for cyber criminals, also their goals and techniques become more sophisticated than ever, all scenarios of malware use and design become stealthier, intelligent and polymorphic. Stealing private data is now the most wanted goal for cyber criminals (black mailing, selling in black market, enter a private place with the name of a victim…) specially they are called phishers.

Phishing attack is one of the main threat on the Internet nowadays [3][4].

The taxonomy of Phishing come from the fact it uses techniques like real fishing: lure, hook, and catch.

- The lure is the social engineering techniques (messages, visual perfection, …) used to catch the attention of the user, generally its main idea is telling victims to fill their informations in the hook
- The hook is the fake website or email used.
- The catch is when criminals use gathered informations in nefarious manners.

Phishers attempt to acquire all possibly confidential information by disguising as an online trustworthy communication, and of course all is about economic aim.

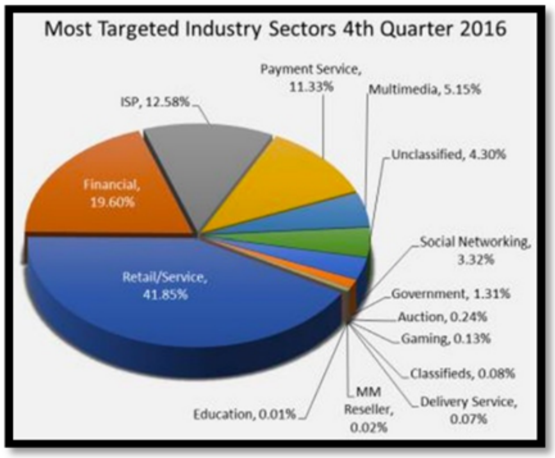


Figure 1: Phishing trends and most affected domains [5].

2. PHISHING DETETCTION TECHNIQUES

2.1 Black-List Based Method

The most commonly used technique is injecting a Blacklisting module or plugin into the browser, Black-listed URLs are databases created by companies (Google Safe Browsing [6], SpoofGuard[7], NetCraft[8]) that contains malicious websites that have been already analyzed, so if the user try to access an already black-listed website it will be noticed that the current website is malicious. Sure, that this method offers a high accuracy [9], because this website had been analyzed before, but what is known today is that sometimes, phishers create malicious websites for a short life time to do a special attack, so the blacklisting will offer nothing if those websites weren't analyzed yet or the client browser database is out of date or some website were misclassified.

This technique is also used to block IP addresses to defend not only phishing but also other kind of threats like spam or malware websites and e-mails. But, the main Disadvantage of this technique is that it cannot prevent zero days phishing attacks, it only detects already black-listed websites.

In fact, this technique demands reaction from clients, also companies could not analyze all websites, so client's reaction to ask those companies to analyze some suspicious websites is necessary. Also, most of applications that use this method is related to some programs like web browser's plugin or e-mail management plugins, so, phishers attack those failures and can sometimes easily break security of those programs.

2.2 Lexical and Host Based Method

Other methods try to analyze a website URL from a lexical point of view. The URL structure is as follows:<protocol><hostname><path>

e.g. **https://mail.google.com:443/mail/#inbox**

- https is the protocol,
- mail.google.com is the hostname or domain,
- 443 is the port,
- mail is the path,
- # is called anchor to indicate to go directly to inbox directory from the path.

Malicious URL have shown different lexical forms than legitimate (also called benign or safe) ones. For example, some of them use a long URL, so that normal user doesn't pay attention to read it carefully, others use an IP address instead a normal hostname (because the use of an IP address makes you sure that you are not relying on a DNS server and to avoid DNS spoofing), others also use the # or @ symbol to redirect the current URL to another malicious URL. So, the method of analyzing the textual or lexical structure of an URL is called lexical based method, according to [10] and [11] URL lexical properties include the length of the hostname, the length of the entire URL, as well as the number of dots in the hostname, these are real-valued features. the URL is split according to strings delimited by '/', '?', '.', '=', '-' and '_' to extract the binary feature for each token in the hostname (delimited by '.') and in the path URL. This is also known as a "bag-of-words.". After extracting those features, they are generally used with a binary classifier like Naives Bayes, "which is a probabilistic classifier, it assumes that the presence or absence for a feature of a class is unrelated to the presence or absence of any other feature even if these features depend on each other or upon the existence of the other features", or with regression algorithms to detect if there is any relation between the model lexical features and this test URL's extracted features. For sure using machine learning algorithms is a subject to discuss, each algorithm has pros and cons in using it. Results will be displayed in Our Method part of this paper. Generally, those extracted features can be combined with other features, like WHOIS, AS, MX numbers informations, methods that use those additional features are called Host Based method. Those features contain informations like "where" malicious sites are hosted, "who" own them, and

Journal of Theoretical and Applied Information Technology
31st March 2018. Vol.96. No 6
© 2005 – ongoing JATIT & LLS

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

"how" they are managed. The following are properties of the hosts (there could be multiple) that are identified by the hostname as part of the URL. For example, the advantage of the use of IP address instead of normal domain, is that attackers can change it permanently. Another example, is when the length of URL is so long, the objective is to hide malicious techniques of redirection like using '@' symbol, or to use some abnormal queries [13]. Another recent attack is that phishers uses punycode(IDN) URLs like: https://xn--80ak6aa92e.com/ which exploits a failure in web browsers, that will translate it to unicode URL that will be https://www.apple.com/, this attack is known as homograph attack. In our dataset we added this feature: we will transform this punycode URLs to Unicode and look if there is a match with a legitimate hostname (we use http://www.alexa.com/ to have a list of legitimate websites and their ranking).

Generally, lexical based method combined with host based method can generate a lexical profile of a phisihing URL that can be used not only to detect, but also to prevent new URLs using those generated profiles, then for sure it will be combined with other features from different phishing detection techniques in our framework.

2.3 Content Based Method

Content based methods analyze the source code of a web page in websites. The main technique used by phishers is to create perfect similar pages of the original ones in visual point of view. Researchers show also that phishing web pages contain malicious javascript codes that allow for deception on the client side using scripts to hide information or to activate changes in the browser [1]. Iframes allow to inject a web page in other web page, that can be specified with visibility to the client or not, e.g.:

<iframe src='http://maliciouswebsites.com/' width='1′ height='1′ style='visibility: hidden;'></iframe>

So, phishers can inject malicious websites that can track users clicks or gather victim informations certainly without its permission. Other techniques used by phishers are for example, disabling right click on the page, pop-up windows and redirecting tags, some of those informations will be used as features in our method. Content based method can be combined with visual based methods, that is based on comparing css values in fake and original websites. Or, it can be combined with some tools like CANTINA [20, 21] which is terms based, it is based on frequency of each term in a specific web page and use heuristics to generate a lexical identity for the web page to detect if it appears in search engine like google. Generally, the technique used by this method is calculating the number of each one of those features (Redirect or not, disabling right mouse click, pop-up windows, iframes, …), and for some threshold it profiles the page in question as suspicious or benign. This method is also used for preventing malicious web pages. But, in those days, other problems are found like having the possibility to obfuscate the source code of web

2.4 Security and Identity Based Method

Several phishing attacks leverage links to misdirect users to websites or drive-by downloads. Since URLs bring information to users about what resource they try to consult, phishers use obfuscation techniques to make phishing URLs look trustworthy for example using https:// for secure http. Studies show that many websites use https, but when looking for Whois informations, they found that the certificate of those websites is not gathered from a trusted issuer and the age of this certificate is very short (less than one year), generally safe companies have a certificate that lives for years [22].

URLs typically contain DNS information i.e. the real IP address of the domain name and hostname of the server, the path of directories and files indicating the location of the consulted resource on the server. Most users read URLs but are not knowledgeable about the information they contain or the DNS hierarchy.

Phishers use this lack of knowledge to mix terms of legitimate URLs to create phishing ones and fool users. Except using https://, other feature can be collected from those URLs; the age of the domain for example, studies show that phishers use very short age of domain, an average of one year. Other features will be shown in the next part of our work. This method is generally used like in criminology field that is profiling 'criminal' websites to detect them or prevent attacks before happening, is an important task when we don't know yet the suspect. But it's an abstract method because profiling identities cannot be precise. And can be combined with profiling normal user's behaviors while using internet.

Journal of Theoretical and Applied Information Technology
31st March 2018. Vol.96. No 6
© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2.5 Conclusion

Methods or criterion based to detect phishing websites are very various, some of them uses heuristics combined with Black-listing, lexicography, content, visual, behavioral or identity based methods, all of them have their pros and cons in detecting false negative and false positive targets, accuracy and precision.

In the next section, we will discuss our method which is based on a combination of black-listing, content, lexical and identity combined with machine learning classifier algorithms, we achieve a new based rule and good accuracy to detect and also to prevent phishing websites.

3. OUR FRAMEWORK

3.1 Collecting Data

3.1.1 Dataset to Build the Base Model

The data used to build our model of identifying phishing is gathered from PHISHTANK, it contains more than 2k of active phishing web pages and websites, then we apply some rules inducted from part discussed above to construct the database model to be used as train data. We used python as a language of programming. We use StandardScaler and 10-fold cross validation from the framework Scikit-learn to build the model, note that we rebuild the final model systematically to get updated with new features that can appear in new incoming test datas. We split data to 80% to train the model and we let 20% to be test of the model built (explication in Figure 6). The way how we built this database and the meaning of the rules will be discussed in part -3.2: Process of our framework.

3.1.2 Test Data to Evaluate the Efficiency of The Mode

To test our model, the new incoming test URLs are collected from internet network traffic to be used as real-time processing or for batch processing i.e. by user request. Our framework works as well if needed as a firewall between the user and the router or to test the safety of an URL by demand. Figure 3 explains how we retrieve test URL data, neither as stream data or in batch process.

3.2 Process of Our Framework

Uniform Resource Locators (URLs), sometimes called web links are the first gateway to access websites [12]. Our goal is being able to detect or prevent phishing websites from only URLs. Out thesis aims to create a firewall that operates on network streaming data and analyze websites features to detect malicious URLs especially phishing websites. In general, our architecture is built as follows:

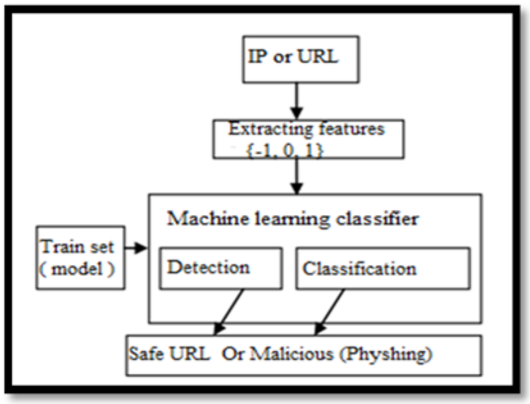

IP or URL
Extracting features {-1, 0, 1}
Machine learning classifier
Train set (model)
Detection
Classification
Safe URL Or Malicious (Physhing)

Figure 2: Global life cycle of our framework

The main idea is to use URL as main entry to our framework, then to combine techniques of detecting phishing discussed above, we will mix the lexical, content and identity base methods beside black-listing based method to extract binary features then to have an optimal based rule detection technique. We start with the blacklist based method, if the URL is already is already blacklisted, it will be assigned as phishing directly (then extract features and update our model), if it is not, we will pass to analyze lexical properties, we believe that criminals generally follow the basic ideologies of precedents, and they also believe that victims do not have a knowledge about URLs structures, after that we use the security based method then the content and the visual properties. For example, if the URL uses an IP address instead of a normal domain, the feature that we called 'is_ip' will have 1 if it is the case or -1 if not.

The final result of the classification will have the name 'Result'; it will be -1 if it is safe or 1 if it is phishing. Generally, we collect 35 binary features (features will be numeric discrete values which can take: -1 for safe, 0 for suspicious and 1 for phishing) to analyze our dataset which is gathered from PhishTank and our system will try to predict the nature of a test URL if it's a phishing URL or legitimate.

The first step is gathering URL from PhishTank dataset or from sniffing packets in real network; an URL is like:

https://172.16.141.29/ww1.wo-site.gt.gy.com/wpincludes/kop/index.html@http://axrotoosh.com/modules/blockcustomerprivacy/translations/1amampa/5a/index_2.html/index_2.html/index_2.html

We can observe that this URL contains an IP address instead of normal domain name, also we can see that it uses @ to redirect directly to URL after @ symbol and at left will be ignored. For example, for the IP address feature, which is included in lexical analysis category, we extract rule as:

Rule IF for 'is_ip' feature (features names as mentioned in Table 1):

*If the Domain Part uses an IP Address → Phishing, is_ip=1

*Otherwise→ Legitimate, is_ip=-1.

And for the length of the whole URL, the rule is like:

Rule IF for 'length_url' feature:

*URL length<54 characters → Legitimate, length_url = -1

*else if URL length≥54 and ≤75 → Suspicious, length_url=0
*otherwise→ Phishing, length_url=1

Also, we can observer the presence of https protocol so normal users will think that is secured website, in our analysis we found that using HTTPS is to delude normal users that is a secured website, it is among the most used tricks by cyber criminals in phishing attacks, in our dataset we found that 57.33% of phishing URLs use HTTPS protocol. To discover the reliability of SSL certificates of each website, we check if the certificate assigned is included in the extent of the trusted certificate issuer, and the certificate age. Certificate Authorities that are consistently listed among the top trustworthy names include: 'GeoTrust, GoDaddy, etc. [23]', and we extract rule as follow:
Rule IF for 'ssl' feature
*URL uses https and certificate issuer is in ca root trusted and its Age≥ 1 Years →Legitimate, ssl=-1
*Using https and issuer is not trusted → Suspicious, ssl=0
*Otherwise→ Phishing, ssl=1

Either we observe too that the domain name contains both '-'symbol and has 2 more sub-domains which is very suspicious, legitimate websites generally puts only one sub-domain and rarely use '-'. Generally, there are 35 characteristic or feature that we extract from only the URL; those features include Black_list, lexical, content of the web page and its identity.

Our framework can be used as real-time analysis or batch processing, each period of 3 days the model is rebuilt again to be up to date with new forms on new URLs, also the selection of features is changed; for example, URLs that uses 'https' protocols we focused on identity first and if is it already blacklisted, if is the case we don't examine lexical and content criterions. Also, if the web page contains only a form to fill with informations, we check if the domain name of the URL contains any of these words:{ 'confirm', 'account', 'banking', 'secure', 'ebayisapi', 'webscr', 'login', 'signin'}, then we analyze the content of the page to look up for any anomalies like Iframes, or malicious redirections. In the worst of all cases, we extract all criterions from the test URL, and we compare it with the model using machine learning classifier algorithms such as: SVM, Decision Trees and Random Forest.

In general, this is how our framework works in real time or in batch processing:

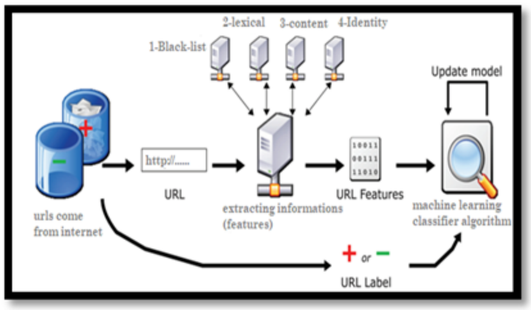


Figure 3: Our real time phishing detection framework

As we can see, when URLs come from internet, we extract informations that we call features that can only have 3 discrete values -1, 0 or 1 [13-15], the bellow table explains it well, all the attributes having a binary values space are generally denoting the absence or presence of respective attribute. Attributes with three possible values are generally representing the strength (safe, suspicious or phishing).

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Features Group	**Phishing Factor Indicator**	**Values**	**Feature in Our DataSet**
Lexical based analytics method on the URL	Having IP Address instead normal hostna-me characte-rs	-1, 1	is_ip
	Having long url	-1, 0,1	length_url
	Uses Shortni-ng Service	-1, 1	short_service
	Having '@' Symbol in the url	-1,1	has_at
	Using '//' redirect-ing	-1,1	double_slash_redirect
	Use non standard port	-1,1	nsd_port
	Having '-' symbol in the hostna-me	-1,1	has_minus
	How many dots in domain	-1,0,1	dots_num
	Having 'https' in the hostna-me	-1,1	https_in_domain
	Is the web page is a form	-1, 1	is_form
	Does the domain name contain stop word	-1, 1	has_stop_words
	Domain registratio n length	-1, 1	domain_length
Abnormal Based Feature (Lexical and host based analytic method) on the URL	Bad Request URL	-1,1	bad_req
	Abnor-mal URL anchor	- 1,0,1	anchor
	Links in tags	-1,0,1	links_of_tag
	SFH: Does the informa-tion gatherd are sent to another domain rather the domain that mad the request	-1,0,1	server_handle_form
	Submitt-ing to email	-1,1	to_email
	Abnor-mal URL	-1,1	abnormal_structure
	Use Punyco-de url	-1,1	punycode
HTML and JavaScript based Features (Content based analytics method) on the web page	Redirect	-1,1	is_redirect
	on mouse over	-1,1	mouse_on_over
	Favicon	-1,1	favico
	Disabli-ng Right Click	-1,1	right_click
	popUp Window	-1,1	pop_up

	Iframe count	-1,1	iframe_num
	Detected by Cantina Frame-work	-1,1	cantina_result
Domain based Feature (Security and identity based method)	Age of domain	-1,1	age_of_domain
	Having good ssl certifica-te	-1, 0,1	ssl
	Number of forward-ing	-1, 0,1	fwd_num
	DNS Record	-1,1	dns
	Web traffic	-1, 0,1	traffic
	Page Rank	-1,1	is_ranked
	Google Index	-1,1	index
	Links pointing to page	-1,0,1	page_links
	Statistic-al report	-1,1	statistics_report
Result	*Final result*	*-1,1*	*result*

Table 1: Our phishing dataset, based on real still active phishing URLs, it contains all informations that we extracted from an URL (-1 means safe, 0 for suspicious and 1 for phishing)

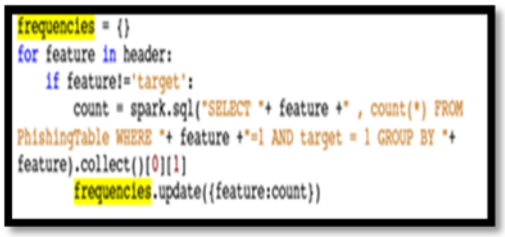
The '*result*' feature will have two final values: 1 if is phishing or -1 if it's a legitimate website. The aim of our heuristic analysis is to transform the problem of analyzing phishing URLs to a binary classification problem to be able to apply machine learning algorithms of binary classification like Decision trees, SVM or Random forest. Our dataset contains 5.5k rows of real active phishing data gathered from PhishTank.

We are using Spark Framework [16] which is an open source distributed processing framework for running large-scale data analytics applications across clustered and standalone hosts. In this paper, we use all the analysis in one single node, to show that Spark works very well to treat data in single node which very enough for our framework. Apache Spark can process data from a variety of data sources, including the Hadoop Distributed File System (HDFS)[17], NoSQL databases and relational data stores such as Apache Hive. Spark supports in-memory processing to boost the performance of big data analytics applications, but it can also do conventional disk-based processing when data sets are too large to fit into the available system memory [16, 17]. The main difference between hadoop and Spark, is that Spark performs real time processing using in-memory computing while Hadoop cannot, it only performs batch processing, second is that Spark is best of the algorithms that uses multiple iterations. We use Spark Sql DataFrames and PySpark to be able sql queries directly and we use Python as programing language because Python is a leader coding language for data science, a simple example is to show which features is the most used by phishing in our dataset, so the phishing feature is when it equals to 1 and target is also 1(target=1 refers to phishing result), we wrote a sql query as follow:

The header of our dataset is as follow:

header=*"is_ip, length_url, short_service, has_at, double_slash_redirect, nsd_port, has_minus, dots_num, https_in_domain, domain_length, is_form, has_stop_words, bad_req, anchor, links_of_tag, server_handle_form, to_email, abnormal_structure, punycode, is_redirect, mouse_on_over, favico, right_click, pop_up, iframe_num, cantina_result, age_of_domain, ssl, fwd_num, dns,traffic, is_ranked,index, page_links, statistics_report, result"*

So, the query will be:



```
frequencies = {}
for feature in header:
    if feature!='target':
        count = spark.sql("SELECT "+ feature +" , count(*) FROM PhishingTable WHERE "+ feature +"=1 AND target = 1 GROUP BY "+ feature).collect()[0][1]
        frequencies.update({feature:count})
```

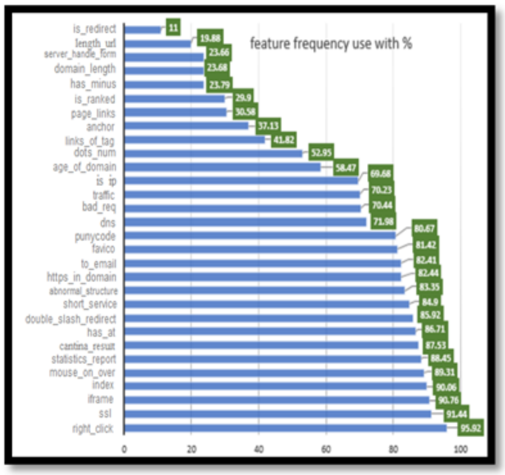


Figure 4 and 5: most used features by phishers

3.3 Classification Process

To have the ability to evaluate our model building, we refer to try different classification algorithms that can work with discrete numeric values and large data:

Random Forest: This algorithm is known to be the best accurate classification algorithm for large data and can deal with most types of data. It chooses randomly the best subset of features that will be used for classification, for each data input. It controls over-fitting and accuracy by itself while choosing those features.

Decision trees: It is a predictive model which creates a tree. It chooses a category of output target data by learning some based rules inferred from a vector of features.

Kernel SVM: Overall, as an approach, the goal is to find that hyperplane effectively divides the class representation of data in 2-D or 3-D. It creates non-linear combinations from features to uplift the test data onto a higher-dimensional feature space to use a linear decision boundary to separate output classes.

Table 2: Confusion matrix for Predicted values

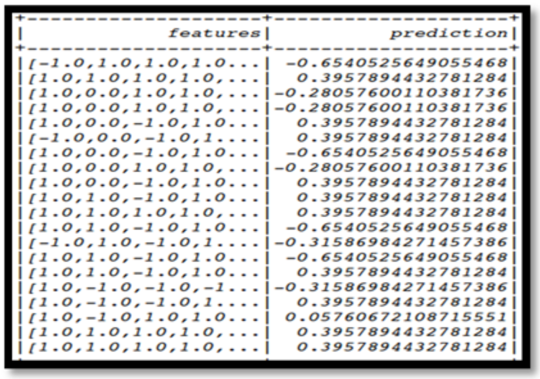
		Phishing	**Safe**
Actual	**Phishing**	TP : True positive	FN(False Negative)
	Safe	FP(False Positive)	TN(True Negative)

- TP: Number of URLs that were classified as phishing correctly
- FP: Number of phishing URLs that were classified as safe
- FN: Number of safe URLs that were classified as phishing
- TN: Number of safe URLs that were classified correctly as safe

From table 2 we can mesure the efficiency of our model, by measuring accuracy, precision and recall, by the following relations:

- Precision-phishing = TP/(TP + FP)
- Recall-phishing = TP/(TP + F N)
- Accuracy = (TP + T N)/(TP + FP + T N + F N)

Figure 6 represents an example from our dataset table (test data that has been transformed to binary features): the input data point to be classified at left, and the predicted output (Phishing or safe) at right with some machine learning classifier.



```
+--------------------+--------------------+
|            features|          prediction|
+--------------------+--------------------+
|[-1.0,1.0,1.0,1.0...| -0.6540525649055468|
|[1.0,1.0,1.0,1.0,...|  0.3957894432781284|
|[1.0,0.0,1.0,1.0,...|-0.28057600110381736|
|[1.0,0.0,1.0,1.0,...|-0.28057600110381736|
|[1.0,0.0,-1.0,1.0...|  0.3957894432781284|
|[-1.0,0.0,-1.0,1....|  0.3957894432781284|
|[1.0,0.0,-1.0,1.0...| -0.6540525649055468|
|[1.0,0.0,1.0,1.0,...|-0.28057600110381736|
|[1.0,0.0,-1.0,1.0...|  0.3957894432781284|
|[1.0,1.0,-1.0,1.0...|  0.3957894432781284|
|[1.0,1.0,1.0,1.0,...|  0.3957894432781284|
|[1.0,1.0,-1.0,1.0...| -0.6540525649055468|
|[-1.0,1.0,-1.0,1....|-0.31586984271457386|
|[1.0,1.0,-1.0,1.0...| -0.6540525649055468|
|[1.0,1.0,-1.0,1.0...|  0.3957894432781284|
|[1.0,-1.0,-1.0,-1...|-0.31586984271457386|
|[1.0,-1.0,-1.0,1....|  0.3957894432781284|
|[1.0,-1.0,1.0,1.0...| 0.05760672108715551|
|[1.0,1.0,1.0,1.0,...|  0.3957894432781284|
|[1.0,1.0,1.0,1.0,...|  0.3957894432781284|
```

Figure 6: Example of predicted target 'result' in test data using Random Forest classifier algorithm (each line is the row corresponding of a test URL that we already transformed to binary features as in Features in the table 1, note that we know the 'result' of each row, the goal of this test is to test the efficiency of the model we built)

So, if the prediction is less than 0 we will say that is safe, else it's phishing URL. In general, machine learning classifiers gave us those performance results:

		Precision-Phishing	Recall-Phishing	Accurcy
Clas-sifier	Kernel SVM	95%	96%	96.5%
	Decision tree	95.5%	96%	97%
	Rand-om Forest	97%	97.5%	99.1%

Table3: Results of different classification algorithms built on our model. Note that, our dataset contains 5.5k input of real active phishing and safe URLs

As predicted, Random Forest gave us the best results, because A random forest is a collection or ensemble of decision trees. Decision tree is built on the whole dataset using all ensemble of features, therefor random forests only subset of rows are used randomly and a number of features are selected at random to train on and a decision tree is built on this subset. SVM shows that is a good classifier also. Note that, to have a good accuracy, the model must be always updated, because phishing techniques vary and use novel manners in each time that they appear.

To perform a critical view on how phishers, proceed their techniques, we did a statistical treatment to have an idea on the most important features in our model, to see what the most features that phishers try to use, we found that hiding the identity is the most used technique in comparison to perform lexical or content based techniques. The figure below shows the most features used by phishers in our model.

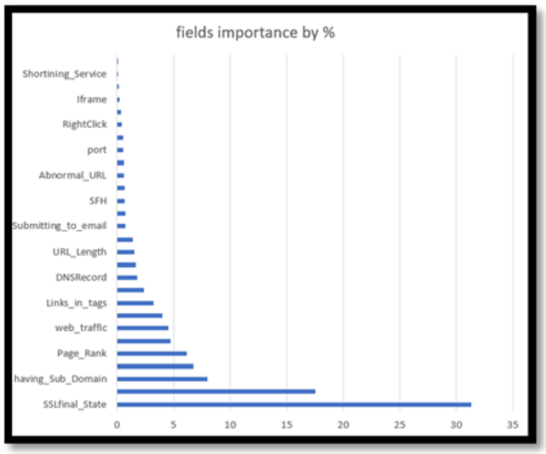


Figure 7: most features used by phishers, all consists into hiding identity

Features importance is very important to our work to see how phishers work and how they evolve. This information is calculated each time we make an update to our model, because it is an information that changes every time.

Actually, our work detects and prevents phishing web pages, our results show that we success into 99.1%. We made rule based method to ne be obliged to do all treatments because we extract so much informations about the current URL and to make our solution to work as a plugin in browsers. As future work and we want to extend our work to be more rapid and also to be able other kind of attacks like Spam or Malwares.

4. CONCLUSION AND PERSPECTIVES

In this paper, the aim is combining different point of view to detect and prevent phishing websites or web pages using only the URL as man entry. The idea is to perform a structural, behavioral and identity based analysis. Our future work will consist to collect data from a large real network and pre-process them to convert the input URLs to discrete binary values as shown before, so the whole treatment will be in real time with streaming data, making a browser plugin for large public use or locally as a WAF (Web Application Firewall). In the era of big data, we should deal of the huge of generated data every moment (Volume), data are heterogeneous because it comes from different sources (Veracity), and the data are generated very quickly (Velocity), and in the same time try to answer those real questions:

- What is the effective minimal set of features that can be utilized to predict a phishing URL?
- How to effectively evaluate data mining based rule techniques to predict phishing websites with best accuracy?

REFERENCES:

[1] https://phishtank.com/developer_info.php

[2] Abubakr Sirageldin, Baharum B. Baharudin, Low Tang Jung, "Malicious Web Page Detection:A Machine Learning Approach", *Advances in Computer Science and its Applications* Volume 279 of the series Lecture Notes in Electrical Engineering, 2013, pp 217-224.

[3] Van-Nam Huynh,Thierry Denoeux,Dang Hung Tran,Anh-Cuong Le,Son Bao Pham, "Phishing Attacks Detection Using Genetic, Programming", *Knowledge and Systems*

Engineering Proceedings of the Fifth International Conference KSE, Volume 2, 2014, pp 185.

[4] D. Kevin McGrath, Minaxi Gupta, Behind "Phishing: An Examination of Phisher Modi Operandi", *Conference: First USENIX Workshopon Large-Scale Exploits and Emergent Threats*, San Francisco, CA, USA, Proceedings, 2008, article No 4.

[5] APWG 2016, Phishing Activity Trends Report 3rd Quarter 2016, https://www.antiphishing.org/.

[6] Google Safe Browsing, https://developers.google.com/safe-browsing/

[7] Spoof Guard, https://crypto.stanford.edu/SpoofGuard/

[8] Neil Chou Robert Ledesma Yuka Teraguchi Dan Boneh John C. Mitchell, "Client-side defense against web-based identity theft", *Conference: Proceedings of the Network and Distributed System Security Symposium*, NDSS, San Diego, California, USA, 2004.

[9] Steve Sheng, Brad Wardman, Gary Warner, Chengshan Zhang, "An Empirical Analysis of Phishing Blacklists", *Proceedings of Sixth Conference on Email and Anti-Spam (CEAS)*, 2009.

[10] Myriam Abramson, David W. Aha, "What's in a URL? Genre Classification from URLs", *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence, Intelligent Techniques for Web Personalization and Recommendation*, 2012.

[11] Aaron Blum, Brad Wardman, Thamar Solorio, Gary Warner, "Lexical feature based phishing URL detection using online learning", *Proceeding AISec '10 Proceedings of the 3rd ACM workshop on Artificial intelligence and security,* ACM New York, NY, USA, 2010, pp 54-60.

[12] URL Definition and Structure, https://docs.oracle.com/javase/tutorial/networking/urls/definition.html

[13] Anh Le, Athina Markopoulou, Michalis Faloutsos, "PhishDef: URL Names Say It All", *Cryptography and Security (cs.CR); Learning (cs.LG); Networking and Internet Architecture (cs.NI),* arXiv.org, 2010.

[14] Yue Zhang, Jason I. Hong, Lorrie F. Cranor, "CANTINA: A Content-Based Approach to Detecting Phishing Web Sites", *Proceeding WWW '07 Proceedings of the 16th international conference on World Wide Web*, ACM New York, NY, USA, 2007, pp 639-648.

[15] Sadia Afroz, Rachel Greenstadt, "PhishZoo: Detecting Phishing Websites by Looking at Them", *Semantic Computing (ICSC), Fifth IEEE International Conference*, 2011.

[16] Lightning-fast cluster computing framework, http://spark.apache.org/.

[17] Hadoop Project, http://hadoop.apache.org/.

[18] Sunila Gollapudi, "Practical Machine Learning", *https://www.packtpub.com/big-data-and-business-intelligence/practical-machine-learning*, January, 2016.

[19] Spark Machine Learning Framework, https://spark.apache.org/docs/2.1.0/mllib-linear-methods.html

[20] Guang Xiang, Jason Hong, Carolyn P. Rose, Lorrie Cranor, "CANTINA+: A Feature-rich Machine Learning Framework for Detecting Phishing Web Sites", *ACM Transactions on Information and System Security (TISSEC):* Volume 14 Issue 2, September, 2011.

[21] Ee Hung Chang, Kang Leng Chiew, San Nah Sz, and Wei King Tiong, "Phishing Detection via Identification of Website Identity", *Conference: International Conference on IT Convergence and Security (ICITCS),* 2013.

[22] Xun Dong, John A. Clark, Jeremy L. Jacob, User Behaviour Based Phishing Websites Detection, *Proceedings of the International Multiconference on Computer Science and Information Technology*, 2008, pp. 783–790.

[23] List of Updated SSL Certificates Issuer, https://ccadb-public.secure.force.com/mozilla/IncludedCACertificateReport.