International Journal of Computer Networks and Communications Security

VOL. 5, NO. 11, NOVEMBER 2017, 248–257
Available online at: www.ijcncs.org
E-ISSN 2308-9830 (Online) / ISSN 2410-0595 (Print)

CPhD: Entire Insight to Detect Phishing Attack

AZAR HOSSEINI[1] and AREZOO HOSSEINI[2]

[1] Department of Electronic and Computer Engineering, School of Electrical Engineering, Iran University of Science and Technology, Tehran 16846, Iran

[2] Pardis Nasibe-Shahid Sherafat, Farhangian University, Tehran, P.O.BOX 19396-14464, Iran

[1]st.azar.hosseini@gmail.com, [2]a.hosseini@cfu.ac.ir

ABSTRACT

Phishing is a type of Internet procedure to seek to get a victim’s credentials such as passwords, credit card numbers, bank account details and other sensitive information by defrauded webpages. Deceptive webpages have particular features to deflect victims to fall into trap. Comprehensive Phishing Detector (CPhD) system can distinguish phishing websites from legitimate websites by extracting these features from URL pattern, website content, images and animations of webpages. The central goal of this system is reducing the run-time of investigation and using instant calculation method and algorithms. Approximately, for every 1,200,000 requests in DNS logs or TLS/SSL logs we need 12min to explore phishing-IPs, 14ms to compare every-two URLs for seeking suspicious URL, 1.20min to compare every two sites words and finally 3min to compare two images. These tests were implemented on 90 Iranian banks, social network sites, search engines and some other well-known sites.

Keywords: *Taxonomy-Phishing, URL, Levenshtein, Correlation, DNS.*

1 INTRODUCTION

Phishing attacks use both social engineering and technical subterfuge to steal consumers’ personal identity data and financial account credentials. Social engineering schemes use spoofed emails to lead consumers to counterfeit websites designed to Trick recipients into divulging financial data such as credit card numbers, account usernames, passwords and social security numbers. Hijacking brand names of banks, e-retailers and credit card companies and phishers often convince recipients to respond. CPhD encompasses defacement and phishing detection methods simultaneously. The proposed method in this system can detect deformed page and phishing because both of attacks have near options in implementation. This claim means, despite of their different aim we should walk the same path to detect these attacks. Phishing attacks intend to attain sensitive information particularly financial ones, while defacement attacks usually try to denial of service and hacked the pages which have highly visitors such as political, news and electronic payment sites. The Proposed method follows well-known procedures for image-processing, text-mining and DNS-Poisoning detection. We investigate recent studies to design an entire taxonomy with two main roots, classification and detection methods (Fig. 1). Classification branch focuses on various insights into deceive and grab victim’s information such as pop-up window to add interactivity and capture victim’s attention, Cross-site scripting (XSS) to execute some codes injection attack for accessing to victim’s information, DNS spoofing to cash the unrelated authority information, manipulating DOM tree and other procedures to impose fake links to victim.

Two passes at phishing taxonomy can be made by researching different types of attacks which include DNS poisoning, Email, phone, Web site and Distributed Attacks and detection methods to cover all kind of phishing attacks.

Section classification represents five collections: 1-DNS Poisoning: DNS cash poisoning which referred to as DNS spoofing, is a kind of security hacking. In this attack Phishers cache information of ISP’s DNS servers and spoof target IPs to redirect users to another pages [1], 2-Email: suspicious links and the fraudful context in email such as spear-phishing attacks[2]. This kind is allocated to definite companies which are highly momentous for business, 3-Phone: wireless activities and usage of its options have their own vulnerabilities that weaken communications. Voice and SMS phishing are renowned paradigm for sending suspicious address or compelling user to reveal his radical information [3], 4-Web site: content analyzing pursues invalid words. The approaches in this area encompass comparing words, Domain and URL analyzing, Dom tree analyzing and all other investigations about content. This branch is nearly common with second set. The only difference between these two parts is their aims. Second part is about particular content in email which implies to suspicious URL with the purpose of financial abuse, while this part has expanded view to respond to all suspicious manners in content of web sites [4], 5-Distributed Attack: distributed phishing is a highly vast attack on victims by a covert transmission to a hidden phisher [5]. Last part alludes to the final goal to implement distributed attack from wide range of victims from diverse zones.

Detection field of phishing attack divided into nine portions: 1-Black/White list based: Some studies illustrate that the usage of white and black list can be the big leap to short the procedure of analyzing [6]. This notion has been arisen from security reports which accentuated consistent attack with duplicate names. 2-Email based approaches: Phishing has wide sight to fraud victims and grab their sensitive information. Spear-phishing and email based are the most momentous fields of grabbing. Papers [7] are focused on how to use suspicious links or how to fraud the usual connection for stealing. 3-Heuristic approaches: these ways are about a novel content-based approach for detecting phishing web sites [8]. Traditionally, they utilize some methods for matching the features like the keywords, IP address, URL features, popup windows, SSL certificates, external hyperlinks, and so forth. The common clusters compass these methods are authentication of links, similarity of content such as financial keywords, rule based approaches like the rules of CSS design, extracted information of search engines and last and foremost option, visualization and all associated algorithm that used in image processing, 4-Honeypot approaches: they are deployed to collect critical information and generate the statistical data to later aid in security [9]. 5-Information based approaches: the application of these methods is in critical information which use for financial activities like what used in the kind of content based.

This field is particular about key words in critical cluster such as credit card number, passwords and username [10]. 6-Machine learning: this approach plays a key role in Data Mining and discovers sophisticated patterns. Different clustering and classification by various algorithms to detect malicious user are employed [11] in this field. Investigating features like what are considered in emails, URL, message-id, orthographic, host-based, lexical and web pages and related algorithms surrounds the most portions of studies. The most prominent difference between heuristic and machine learning is that heuristic talks about achievable features or explicit ones, but machine learning related to path for discovering new features and properties to detect the wide range of phishers' manner, 7- Network level approaches: This area includes malware, viruses and worm that are doing the specific purposes [12]. Some papers emphasize on the malicious application by particular software such as firewalls, IDS and other detector. 8- Visual clue based approaches: comparing the webpages based on the visual features. Visual contents are the feature set that holds webpage layout, images, logos, forms, background color, font color, and so on. [13], 9- Website features based approaches: the primary aim of these approaches are analysis the HTML structure and DOM Tree of webpages and define metrics for detecting phishing websites [14].

The paper is organized as follows: Section 2 presents the Paper work and total scheme of CPhD. Subsection 2.1 shows the procedure of compare-name module with the optimized Levenshtein algorithm. Subsection 2.2 associated with forged IP-address detection and the methodology of investigating the various inputs, DNS and TLS/SSL. This part is the controversial issue in DNS poisoning. Subsection 2.3 introduces the comparing contents of web sites by common words filtering. Subsection 2.4 reveals the experiments and results of combination of correlation algorithm and SIFT and then conclusion is given in Section 3.

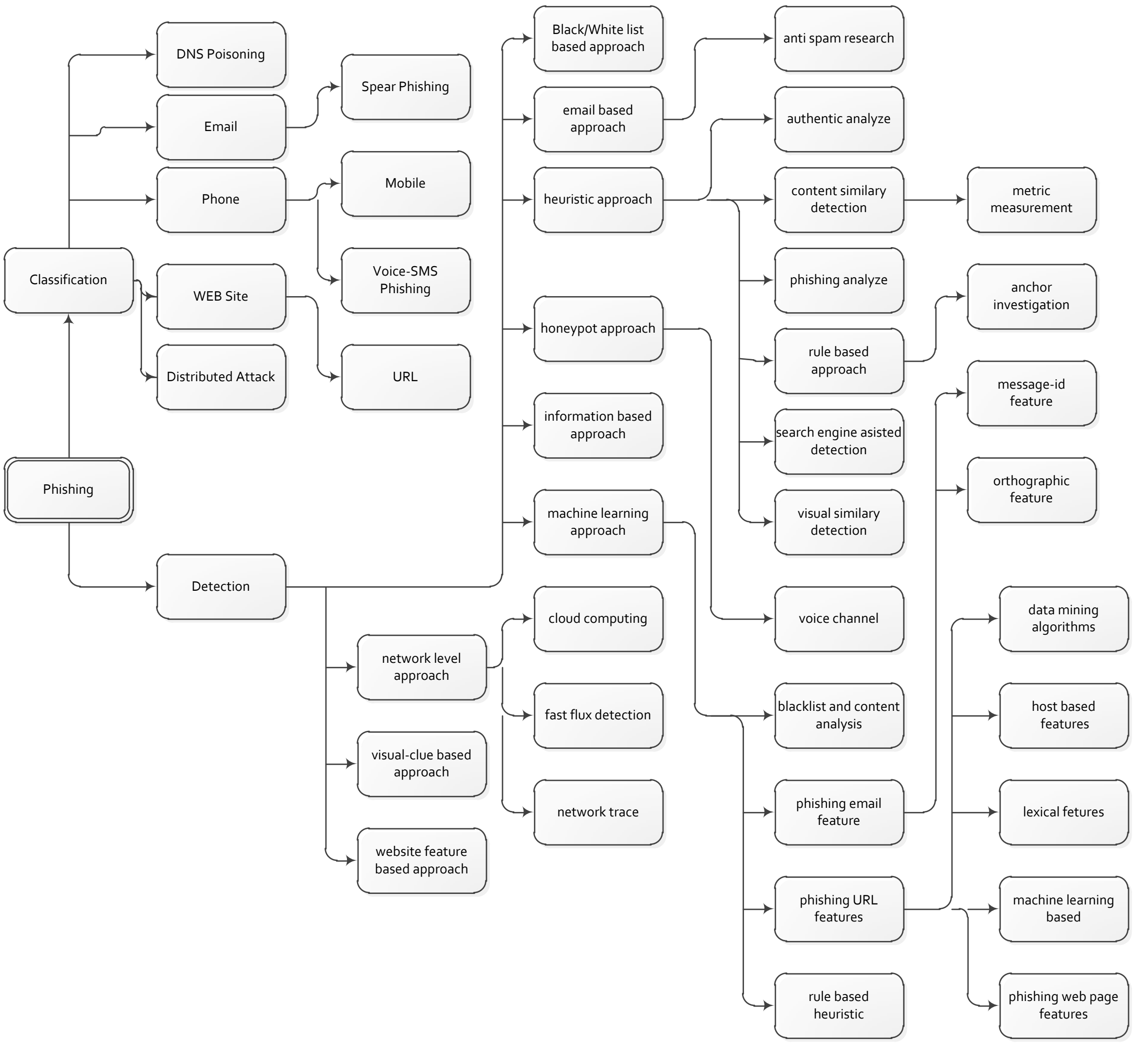

Fig. 1. Phishing Taxonomy

2 CPHD SYSTEM

Studying with a high attention score provided us with designing the comprehensive and prompt phishing detector. An unexaggerated state of the facts makes up the claim that this automatic system won the battle of similar products to detect inevitably a phishing web page. In CPhD system we have been able to optimize the Levenshtein algorithm by using the Trie algorithm [15], the numbers and symbols allowed in the domain name and the alphabetical similarity table. The sequel of this calculation is a module for discovering similar domain names and similar URLs.

Fig. 2 demonstrates the overall schema of the CPhD system.

Text-mining, image-processing and IP mapping techniques construct the CPhD to success to gain credible consequences in deal with suspected bulky data. CPhD scrutinizes different type of traffic such as DNS, TLS, HTTP/S in the shortest time possible and exploits the analytical tools such as Pentaho_ Intuitive and Scalable tools.

Text-mining, image-processing and IP mapping techniques construct the CPhD to success to gain credible consequences in deal with suspected bulky data. CPhD scrutinizes different type of traffic such as DNS, TLS, HTTP/S in the shortest time possible and exploits the analytical tools such as Pentaho_ Intuitive and Scalable tools.

First and foremost part that should be considered is the fragmentation of URL. The fragmentation duty is assigned to the compare-words module.

Following subsections will explain entirely the depiction of all modules.

2.1 Compare-words module

The first stage of phishing detection in CPhD is dedicated to the compare-words module.

This module runs simultaneously with the invalid IP Discovery module. Splitting the URL into different parts, and comparing the parts obtained with valid domain names are two radical tasks of this module. Valid names allude to a list of domains that are important for user who wants to trust them. For example, the user always wants to make sure that the bank's online payment page is valid. Therefore, the domain name of the online paid site will be one of the members of this list. The reason of splitting is exuded from wide testing the phishing URLs. They have almost suspicious words in different parts of address, from domain to all subdomains.

2.1.1 Fracturing URL module

Fracturing URL can be implemented with two points of view. Firstly, isolating meaningful parts of an address, such as hostname, domain, TLD, and SLD then compare with the same genre of the valid domain lists looking like domain name and TLD. This kind of split will not be comprehensive because phishing scams may put the valid domain name in the Path or subdomain or other parts of the URL where the user has deceived and entered his sensitive information on the page loaded. According to the compare-names module embedded in CPhD, We recommend that another procedure should be selected to split URLs. Suppose the list of valid domain names involves “abc.com” and “xyzq.ac.cu”.

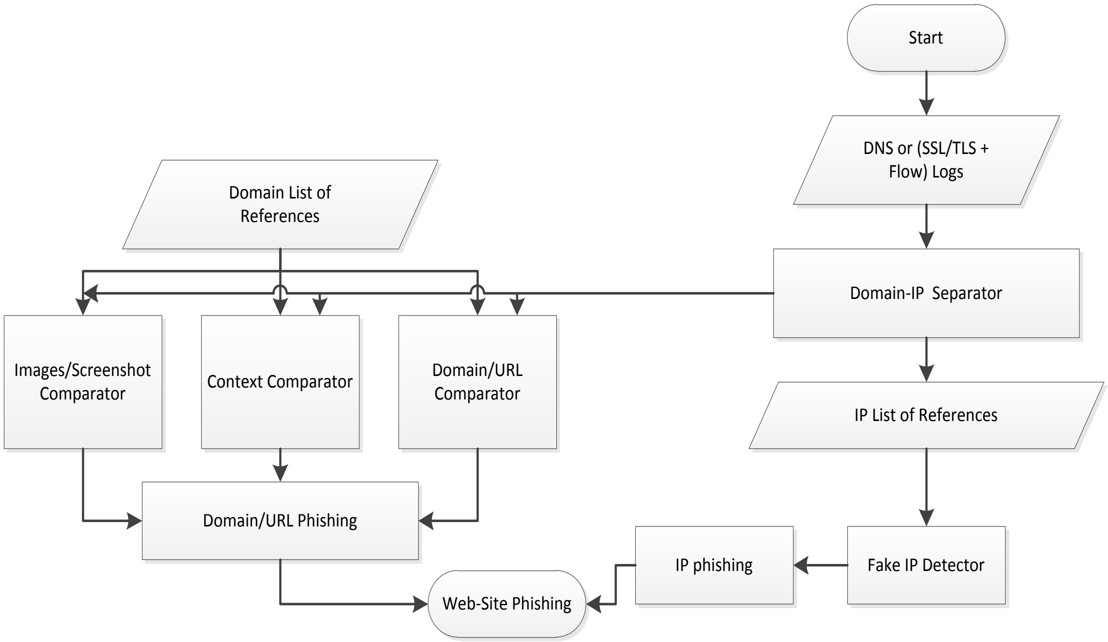

Fig. 2. CPhD System Architecture

The meaningful parts of these addresses are abc, com, xyzq, ac and cu respectively. As you can see, the selection ranges in length from 2 to 4 letters. Thus, the suspicious URL should be divided into all two letters, then all three letters, and finally all four letters parts. The speedup in implementation depends on the parallel proceedings. All two-letter extracted from suspicious URL, for example "Suspicious.sus.su", should be compared with meaningful parts of valid domain names, "abc" ,"com","xyzq","ac" and "cu" in parallel form. This way will be repeated for three-letter and four-letter of suspicious URL.

2.1.2 Comparing Process

In this section we use the optimized Levenshtein algorithm to compare the names. Levenshtein distance or edit distance in data theory and computer science is the yardstick to calculate the difference between the two strings. In traditional form the cost of insert, replace and delete operations are +1, 0 and -1 respectively. In spite of the reliable results of Levenshtein, CPhD proposes two perspectives to enhance the implementation, one hand accuracy optimization and the other hand diminishing run time.

2.1.2.1 Speed Optimization

The Trie algorithm or prefix tree is a tree data structure used for mappings[1], and the simplest way to optimize the comparison speed with the list of meaningful words that are restricted. The comparison between two terms is performed on the basis of meaningful words in the English dictionary, while the meaningful words in the compare-words module are the list of white domains and subdomains of valid sites such as banks, financial and credit institutions, libraries and online payment sites. Suppose you have received a doubtful email with a seductive title after opening it, check its content with the knowledge that it is likely to be infected with the intention of phishing operation. Here, you may have to check your bank account, so first you need to compare the email URL with valid addresses and make sure that it is correct. In CPhD, Trie algorithm is performed as a high-speed parallel operation with user's domains list as a meaningful words list. Aforementioned above for dividing obfuscated URL we require to know the shortest and the longest word in the user's list. If the shortest length is m and the longest is n, the URL will be divided into range of (m-n) length. If "f" is string with "q" length:

$$f = \{ f_1, f_2, f_3, f_4, ..., f_{z-1}, f_z, f_{z+1}, ..., f_{q-2}, f_{q-1}, f_q \} \tag{1}$$

Then the number of whole [m,n] words based on the target list is $\sum_{i=m}^{n} q-(i-1)$ Consequently, a URL with "q" length will have

$\sum_{i=m}^{n} i$ execution steps and $\sum_{i=m}^{n} q-(i-1)$

productive parts.

After comparing the first and second letters between both words (suspicious word and valid address) based on Levenshtein, the couple of letters in suspicious words should compare with all valid addresses based on Trie. Then resume comparing third letters and iterate again to compare triplet letters of suspicious word with valid addresses. The same procedure is performed to the end of the comparison of all letters from the two words being compared, in order to finish the Levenshtein and Trie operations in parallel. Because of the parallel implementation of both algorithms, this operation has a significant speed. In other words, instead of comparing every suspicious word with each word of valid list individually, it will compare with all valid addresses simultaneously.

2.1.2.2 Optimization of accuracy

Exploring non-structured data with natural language processing (NLP), statistical modeling and machine learning methods may be difficult and challenging because natural language texts are often contradictory. These texts often include ambiguities like syntaxes and semantics acting as slang terms, sarcastic speculation or the language of a particular age group. But CPhD merely encounter the fonts in colorful web browsers, Chrome, IE, Opera, Mozilla and etc., hence the most important problem in word processing has been removed from our procedures. As a result there is no need to process the image of the writing of the letters because they are not in noisy mode as well as not according to the type of personality. They are usually written with default fonts of browsers. The tests performed on the category of default fonts. Despite the fact that users can change defaults, the Arial, Times New Romand and courier fonts are mostly used for URL requests.

Thereupon, checking the similarity of the acceptable letters in the URL is limited to compare two letters based on the three fonts denoted. We

[1] mapping is a collection of ordered pairs (key, value), each key can map to at most one value

used (H: 400-V: 441/Pixels) for each letter {a-zA-Z0-9.:;?="'-_+} for comparing based on combination of correlation and SIFT algorithm. With regard to correlation, the density of the connecting lines between images of each couple of letters reflects the degree of dependency of them. The merit of this proposed algorithm was the

reliable SIFT points and cohesion among them. The result of the comparison was stored as the coefficient of similarity between pair letters in Table 1.

The final suggestion to amend the Table 3 is checking the accuracy of the comparison process during a referendum from forty three IT specialists.

It's interesting to know that the accuracy ratio of the correlation-SIFT algorithm is nearly 23% more than the sinner sight of the man. The results of the referendum were not reliable due to people's impatience, lack of understanding, neglect, and many other human’s fault parameters.

Table 2 is a brief explanation of the entire word comparison module of CPhD.

And Table 3 presents some result of comparing 2500 Domain requested with 90 banks of Iran, social network sites, some search engines and other well-known sites witch are assigned to references. All rows report this fact that sequels have coefficient value less than “2”.

Table 1: Coefficient of Similarity between Letters.

char	char	coefficient	char	char	coefficient	char	char	coefficient	char	char	coefficient	char	char	coefficient	char	char	coefficient	char	char	coefficient	char	char	coefficient	char	char	coefficient	char	char	coefficient	char	char	coefficient
1	2	0.13	a	u	0.18	b	5	0.18	d	o	0.26	e	0	0.25	g	p	0.45	j	l	0.53	m	p	0.06	o	q	0.43	s	t	0.06	x	8	0.08
1	4	0.08	a	6	0.26	b	f	0.13	d	6	0.26	e	l	0.06	h	8	0.06	j	p	0.06	m	q	0.09	p	9	0.34	s	x	0.06	x	y	0.44
1	7	0.2	a	g	0.26	b	o	0.49	d	h	0.33	e	z	0.14	h	p	0.19	j	q	0.13	m	r	0.19	p	q	0.6	s	1	0.06	x	z	0.13
1	9	0.06	a	v	0.06	b	6	0.41	d	p	0.54	e	5	0.24	h	i	0.06	j	1	0.31	m	0	0.06	p	r	0.09	s	z	0.33	x	2	0.06
2	9	0.11	a	8	0.06	b	g	0.24	d	7	0.06	e	n	0.13	h	q	0.19	j	r	0.06	m	u	0.09	p	0	0.25	s	2	0.13	x	4	0.06
2	3	0.06	a	n	0.14	b	p	0.65	d	i	0.13	f	7	0.06	h	k	0.06	j	7	0.06	m	3	0.06	p	t	0.06	s	5	0.71	y	9	0.05
2	5	0.13	a	w	0.06	b	8	0.19	d	q	0.53	f	p	0.14	h	r	0.03	j	t	0.19	m	w	0.33	p	1	0.06	s	6	0.21	y	z	0.13
2	7	0.25	a	0	0.48	b	h	0.74	d	8	0.06	f	8	0.06	h	l	0.13	j	9	0.13	n	0	0.31	p	y	0.13	t	u	0.06	y	1	0.13
3	0	0.03	a	9	0.29	b	q	0.53	d	j	0.06	f	r	0.31	h	t	0.13	j	y	0.19	n	r	0.3	p	4	0.06	t	1	0.43	y	2	0.06
3	7	0.06	a	o	0.46	b	9	0.19	d	r	0.06	f	9	0.06	h	0	0.06	k	1	0.09	n	1	0.06	p	6	0.06	t	4	0.06	y	7	0.2
3	8	0.34	a	x	0.06	b	i	0.13	d	0	0.26	f	t	0.36	h	m	0.25	k	x	0.21	n	s	0.06	q	2	0.09	t	7	0.06	z	1	0.06
3	9	0.18	a	1	0.06	b	r	0.06	d	9	0.3	f	i	0.24	h	u	0.06	k	6	0.09	n	4	0.06	q	t	0.06	u	x	0.06	z	2	0.55
4	7	0.09	a	b	0.18	c	d	0.19	d	k	0.06	f	j	0.13	h	1	0.13	k	y	0.1	n	t	0.06	q	4	0.18	u	0	0.18	z	4	0.06
4	9	0.03	a	p	0.26	c	s	0.13	d	u	0.11	f	k	0.06	h	n	0.76	k	l	0.19	n	9	0.06	q	y	0.21	u	y	0.24	z	5	0.06
5	0	0.06	a	y	0.06	c	e	0.74	d	1	0.06	f	1	0.24	h	6	0.13	k	n	0.06	n	u	0.35	q	5	0.06	u	1	0.06	z	7	0.41
5	6	0.19	a	2	0.06	c	u	0.06	d	e	0.06	f	l	0.25	h	o	0.13	k	r	0.06	n	o	0.53	q	6	0.09	u	5	0.03			
5	7	0.06	a	c	0.24	c	0	0.35	d	l	0.13	g	8	0.43	i	9	0.06	k	s	0.15	n	v	0.19	q	8	0.2	u	6	0.03			
6	0	0.23	a	q	0.41	c	g	0.06	d	4	0.06	g	q	0.6	i	q	0.13	k	t	0.09	n	p	0.19	q	0	0.28	u	v	0.59			
6	7	0.06	a	z	0.19	c	x	0.06	d	f	0.06	g	9	0.54	i	j	0.56	l	p	0.09	n	w	0.06	q	9	0.55	u	w	0.11			
6	8	0.16	b	0	0.25	c	2	0.13	d	n	0.06	g	s	0.06	i	r	0.29	l	q	0.09	n	q	0.13	q	1	0.06	v	z	0.06			
6	9	0.43	b	c	0.24	c	n	0.13	e	6	0.26	g	h	0.06	i	k	0.29	l	1	0.76	n	x	0.13	q	r	0.06	v	1	0.06			
7	9	0.11	b	k	0.13	c	z	0.13	e	o	0.24	g	u	0.03	i	t	0.39	l	r	0.35	o	0	0.79	r	v	0.13	v	7	0.06			
8	0	0.2	b	u	0.2	c	5	0.24	e	9	0.19	g	0	0.24	i	l	0.59	l	6	0.09	o	s	0.06	r	x	0.06	v	w	0.56			
8	9	0.13	b	1	0.19	c	o	0.41	e	p	0.2	g	j	0.19	i	y	0.13	l	t	0.46	o	5	0.15	r	1	0.3	v	x	0.19			
9	0	0.13	b	d	0.71	c	6	0.2	e	g	0.13	g	y	0.19	i	m	0.06	l	7	0.15	o	u	0.38	r	y	0.18	v	y	0.65			
a	4	0.19	b	l	0.19	c	p	0.25	e	q	0.13	g	4	0.18	i	1	0.65	l	y	0.09	o	6	0.2	r	6	0.11	w	7	0.06			
a	d	0.4	b	v	0.13	c	9	0.06	e	i	0.06	g	l	0.06	i	n	0.13	l	9	0.09	o	v	0.06	r	7	0.11	w	x	0.19			
a	s	0.06	b	4	0.06	c	q	0.31	e	r	0.05	g	5	0.06	i	7	0.06	l	n	0.09	o	8	0.25	r	t	0.13	w	y	0.19			
a	5	0.11	b	e	0.19	d	5	0.06	e	j	0.06	g	o	0.25	i	p	0.13	m	n	0.6	o	9	0.26	r	u	0.06	w	z	0.06			
a	e	0.35	b	n	0.06	d	g	0.38	e	u	0.06	g	6	0.13	j	k	0.13	m	o	0.15	o	p	0.43	s	7	0.06	x	7	0.06			

Table 2: Psuedo Code of Compare-Word Module.

```
Algorithm optimized Levenshtein
1.  Set n to be the length of R. // R is Reference Domain
2.  Set m to be the length of Req.// Req is Requested Domain
3.  If n = 0, return m and exit.
4.  If m = 0, return n and exit.
5.  Construct Trie_node *children[m]
        a. Construct a matrix containing 0..m rows and 0..n columns.
        b. Initialize the first row to 0..n.
        c. Initialize the first column to 0..m.
6.  Examine each character of R (i from 1 to n).
7.  Examine each character of Req (j from 1 to m).
8.  Set MAX_cost to 1000:
        a. If r[i] equals req[j], then:
           //r is lower letter of R
           cost ← cost(previous_row[ column - 1 ]).
        b. If r[i] doesn't equal req[j], then:
           //req is lower letter of Req
                i. Trace coefficient in Table.1:
                        1. True: assign to cost.
                        2. False:  cost ← cost(previous_row[ column - 1 ] + 1)
9.  Set cell d[i,j] of the matrix equal to the minimum of:
        a. The cell immediately above plus 1: d[i-1,j] + 1.
        b. The cell immediately to the left plus 1: d[i,j-1] + 1.
        c. The cell diagonally above and to the left plus the cost: d[i-1,j-1] + cost.
10. After the iteration steps (6-9) are complete, the distance is found in cell d[n,m].
```

Table 3: Some example of discovery phishing URLs.

Requests	References	Coefficient
www.banknellat.ir	www.bankmellat.ir	0.4
www.bankmellet.ir	www.bankmellat.ir	0.65
www.banlmellat.ir	www.bankmellat.ir	0.81
www.pm1.ir	www.bmi.ir	0.7
www.qmp.ir	www.qmb.ir	0.35
www.gmp.ir	www.qmb.ir	0.75
www.faceboak.com	www.facebook.com	0.54
www.gmgir.com	www.gmail.com	1.39
www.ehow.com	www.ebay.com	1.61
www.etsy.com	www.ebay.com	1.94
www.mail.cdhoo.com	www.mail.yahoo.com	1.6
www.youtubeta.com	www. youtube.com	2

2.2 Invalid IP detection

Phishers use their own methods to poison DNS servers and change the IP-table then send malicious DNS responses to victims. Thus, Invalid IP detection module can check simultaneously the IP resolves and the domain names of the requests to prevent misunderstanding in suspicious communications.

The Invalid IP detection module inspects TLS/SSL and DNS logs then if the requested IPs run counter to the IPs associated with the valid domains list, they will be reported as phishing IPs. The functionality of this module will be permanent, if the IPs list corresponding to the valid domain names is updated consistently.

2.3 Comparing contents

Financial sites most often use certain words such as "account, payment, cash, master card, bank, waybill, freight, cheque, financial, brisk, exchange, fee, money, shipment, warrant, saleable, wholesale, marketing, business, credit, cost, loan, budget, grant, dollar and pound". The contents of defacement sites have been changed, in one hand the hacker's specifications like team name and the other hand the common word of hacks acting as, "hacker, hacked, down, lose, unavailable, failed, cyber". This section plays a key role in defacement and phishing attacks because it can reduce the execution time of detection process.

This level of processing deals with the language of the web pages. For example, tokenizing the words in Persian language is associated with the distance definition between them. Some words are separated by a half-space and others by a full space. Accordingly, tokenize words varies in different languages. Due to the versatile trait of CPhD to detect multifarious languages, both distances are considered in the tokenize operation. The procedure of this section has three steps: 1- extract and tokenize all words of requested and reference sites, 2-find the reference site's words in requested one, 3-calculate the ratio of common words between two sites, 4-if the acquired ratio is more than 60 percent then the requested site will be warned. Table 4 shows the reasonable sequels of these processes.

Table 4: The consequences of the comparing-contents module.

References site	Requested site	Similarity factor
http://cid.ir	http://CAR.Ir	94/173=54%
http://bmi.ir	http://dmk.ir	1200/1277=94%
http://bim.ir	http://hrg.iR	63/199=31%
http://bmi.ir	http://qmt.ir	1199/1277=93%

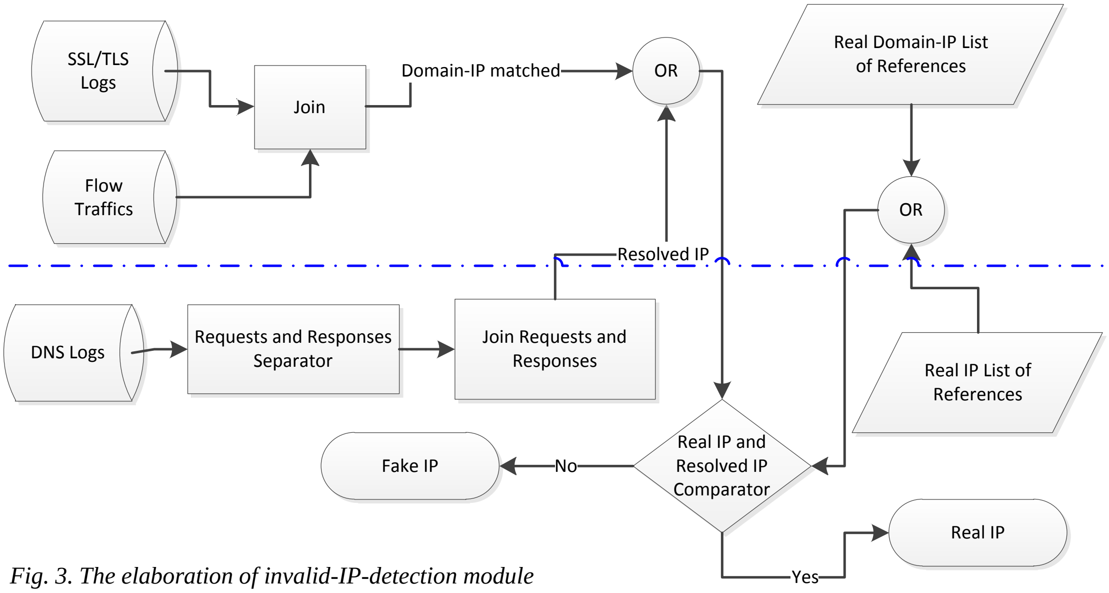
SSL/TLS Logs
Flow Traffics
Join
Domain-IP matched
OR
Real Domain-IP List of References
OR
Resolved IP
DNS Logs
Requests and Responses Separator
Join Requests and Responses
Real IP List of References
Fake IP
No
Real IP and Resolved IP Comparator
Real IP
Yes

Fig. 3. The elaboration of invalid-IP-detection module

2.4 Comparing images and animations

Comparing images is one of the most promising ways to discover defacement and phishing sites. Some articles just refer to Comparison of Screenshots of pages [16], while this method is not suitable for dynamic sites with moving pictures. We recommend that download all the images and animations to compare reputable site with phishing paradigms. All downloads should be engaged our proposed algorithm to compare peer-to-peer images. This suggestion may take much time but has a highly degree in confidential process. The best solution for reducing time consumption is parallel comparing images.

As mentioned previous, we exploit combination of two independent algorithms. Firstly, SIFT algorithm that is a well-known method of image processing. This algorithm is a method for detecting, extracting, and describing key points in images which can be used in applications such as image-matching, object identification, 3D-scene reconstruction, and etc. Secondly, correlation algorithm to compute the solidarity between SIFT points. Some other studies emphasize on comparing the histogram and the intervals between the corresponding points [17].

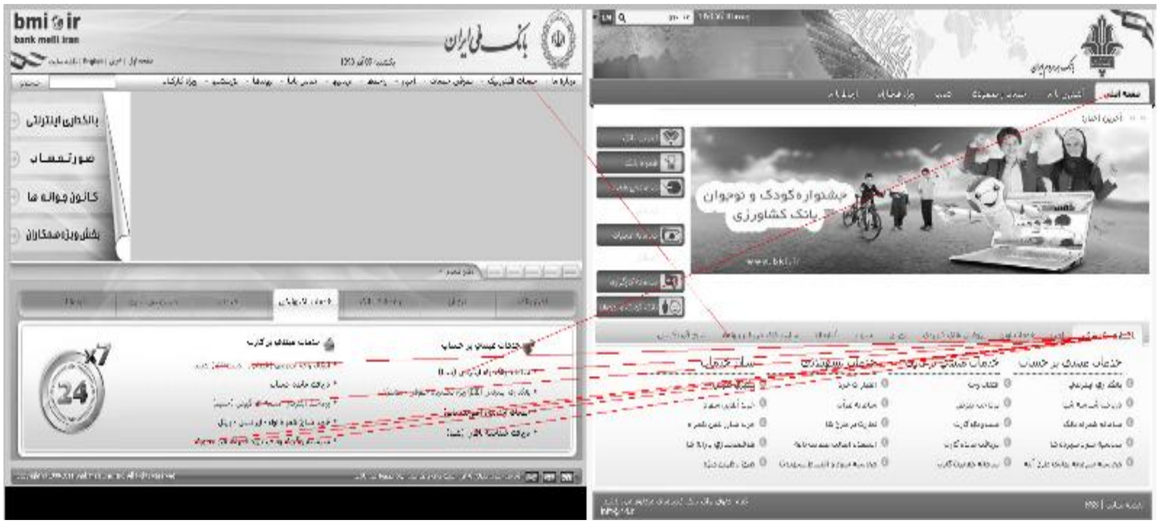

Fig. 4. Discrepancy between www.bmi.ir and www.bki.ir

Figure 4 exposes the result of our algorithm of comparing images on two bank sites. These banks have almost similar name because their analogous coefficient was less than “2”. In this article we assumed that if coefficient was less than 2 then respective sites should be sent to the next modules for comparing their contents and images.

2.4.1 SIFT- Correlation

SIFT constitutes the fundamental initiation in our compare-images module. This algorithm can choose the key points of picture for comparing with intended algorithms like chi-square, correlation, root-mean-square and others. Exclusive points of each image seek their corresponding points from other image during computation of correlation. Correlation algorithm can calculate the nearest point based on cohesion between key points. Key points are not arbitrary rather they are the result of constant scales, therefore every picture has own key points. This assumption helps us to rely on consequence of correlation. Our computing is end to the lines which connect correlated points to each other. The threshold of similarity between web pages is more than 200 lines.

3 CONCLUSIONS

Phishing attacks have always been one of the major concerns of cyber security and there have been many studies in this area. One of the key challenges in phishing detection is the high volume of input data and the length of the comparison process. In data centers, parallel methods are commonly used, and we tried to consider this parameter during CPhD system processing. The main purpose of this system is to cope with the high volume of incoming traffic, reduce the time of comparison and increase the accuracy of the results. CPhD is coded in Python and run on the computer with a 2.26-GHz Intel Core i5 i5-430M CPU, 4GB of RAM. The privilege of CPhD is its capable usage at analytical systems like Splunk which equipped with capacious database for online analysis. Each pair of Domains consumes 14ms in compare-words module and 2.4m in compare contents module and finally 3m in compare-image module. Because of parallel implementation of last two parts, for every couple of URLs we approximately need 180.014s to caveat phishing webpage. The duration of discovering a fake IP in invalid IP detection module is 0.6ms. These reports track 1,200,000 domains requested per day, with 90 valid addresses.

4 REFERENCES

[1] Sun Bin ,et al, “A DNS Based Anti-phishing Approach”, Networks Security Wireless Communications and Trusted Computing (NSWCTC), 2010.

[2] Jingguo Wang, “Research Article Phishing Susceptibility: An Investigation Into the Processing of a Targeted Spear Phishing Email”, IEEE Transactions on Professional Communication, Volume: 55, Issue: 4, 2012.

[3] Alta van der Merwe, “Phishing in the System of Systems Settings: Mobile Technology”, Systems, Man and Cybernetics, IEEE International Conference, 2005.

[4] GUANG XIANG, “CANTINA+: A Feature-rich Machine Learning Framework for Detecting Phishing Web Sites”, ACM Transactions on Information and System Security (TISSEC), Volume 14 Issue 2, 2011.

[5] Markus Jakobsson, “Distributed Phishing Attacks”, Workshop on Resilient Financial Information Systems, 2005.

[6] JungMin Kang, “Advanced White List Approach for Preventing Access to Phishing Sites”, Convergence Information Technology, 2007.

[7] Madhusudhanan Chandrasekaran, Ramkumar Chinchani, “PHONEY: Mimicking User Response to Detect Phishing Attacks”, International Symposium on a World of Wireless, Mobile and Multimedia Networks, 2008.

[8] Mahmoud Khonji, “Phishing Detection: A Literature Survey”, IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 15, NO. 4, 2013.

[9] Prof. Gayathri Naidu, “A Survey On Various Phishing Detection And Prevention Techniques”, International Journal Of Engineering And Computer Science, Volume 5, Issue 09, 2016, pp. 17823-17826.

[10] Ram Basnet, “Detection of Phishing Attacks: A Machine Learning Approach”, Soft Computing Applications in Industry, volume 226, 2008, pp 373-383.

[11] Saeed Abu-Nimeh, “A Comparison of Machine Learning Techniques for Phishing Detection”, anti-phishing working groups 2nd annual eCrime researchers summit, 2007, pp. 60-69.

[12] Tianyang Li, “LARX: Large-scale Anti-phishing by Retrospective Data-Exploring Based on a Cloud Computing Platform”, Computer Communications and Networks (ICCCN), 2011.

[13] Huajun Huang, “Browser-side Countermeasures for Deceptive Phishing Attack”, Fifth International Conference on Information Assurance and Security, 2009.

[14] Noor hazarina hashim, “Investigating Internet Adoption and Implementation by Malaysian Hotels: An Exploratory Study”, Doctor of Philosophy thesis, 2008.

[15] Morten Wærsland, “Text Pattern Discovery & Extraction”, master’s thesis, University of Stavanger, 2016.

[16] Kuan-Ta Chen, et al., “Fighting Phishing with Discriminative Keypoint Features of Webpages”, IEEE Internet Computing, May, 2009.

[17] Omid Asudeh, “A New Real-Time Approach For Website Phishing Detection Based On Visual Similarity”, Master Thesis, The University Of Texas At Arlington, 2016.

AUTHOR PROFILES:

Azar Hosseini Master of Science (M.Sc.) in Secure Communication Eng., Iran University of Science and Technology, 2013. Bachelor of Science (B.Sc.) in Electrical Eng., Dr. Shariaty Technical College, Iran, 2008. Fields of Interest: Machine Learning, Data Mining, Information Security, Internet of Things (IoT), Cognitive Radio, Communication Networks Mobile Communications, Wireless Sensor Networks.
Email: st@azar-hosseini.com
Web: http://azar-hosseini.com/

Arezoo Hosseini Doctor of Philosophy (PhD) in Pure Mathematics at Topological Groups, University of Guilan, Iran, 2012. Master of Science (M.Sc.) in Pure Mathematics at Topological Groups, 2008. University of Guilan, Iran. Bachelor of Science (B.Sc.) in Mathematics, Iran University of Shahid Rajaee, 2006. Fields of Interest: Topological Groups and Dynamical system, cohomological Groups.