Decisive heuristics to differentiate legitimate from phishing sites

Sophie Gastellier-Prevost, Gustavo Gonzalez Granadillo and Maryline Laurent
CNRS Samovar UMR 5157, Institut Telecom, Telecom SudParis
Evry, FRANCE
Email: {Sophie.Gastellier, Gustavo.Gonzalez_Granadillo, Maryline.Laurent}@it-sudparis.eu

***Abstract*—Phishing attacks are a major concern for preserving Internet users privacy, especially when most of them lead to financial data theft by combining both social engineering and spoofing techniques. As blacklists are not the most effective in detecting phishing sites because of their short lifetime, heuristics appears as a privileged way at time 0. Several previous studies discussed the different types of phishing characteristics that can help defining heuristics tests, as well as comparing them to blacklists. In our paper, we studied heuristics using a different approach. Based on the characteristics of phishing URLs and webpages, we defined 20 heuristics tests and implemented them in our own active anti-phishing toolbar (Phishark). Then, we tested the heuristics effectiveness and determined which heuristics are decisive to differentiate legitimate from phishing sites.**

I. Introduction

The development of online transactions through Internet is a great improvement for Internet users who can now benefit from an easy access to many services, with greater choice of products, regardless the time or their location. As any lucrative business, the main drawback of this new market place is to attract many people looking for easy and rapid profits. By using website forgery to spoof the identity of a company (typically a bank, an auction site,...), phishing attacks trick Internet users to reveal confidential information (such as login, password, credit card number,...). Yu et al. study [29] details phishing motivations and vulnerability causes.

As phishing attacks are now widespread and quite effective over the Internet, many works focused on preventing those attacks (cf. section II). One popular method aims to integrate anti-phishing protection into the web browser of the user (ie. anti-phishing toolbar), making use of two kinds of classification methods: blacklists and heuristic tests. Considering the paradox that one of the main issues associated to phishing sites detection is their short lifetime, and that blacklist-based anti-phishing toolbars need many hours to become effective to detect phishing sites [26], heuristics appears as a privileged way to efficiently detect phishing sites at time 0.

Several previous studies were conducted about anti-phishing heuristics in order to determine their efficiency and characteristics. However, beyond the effectiveness of heuristics regarding other methods, we found that existing research lacks of discussions about the prevalence of heuristics tests. Our paper differs from previous works by concluding which heuristics tests - for both URL and page content analysis - are decisive to identify a legitimate page from a phishing site.

In order to test the effectiveness of heuristics, we designed our own heuristics based-only anti-phishing toolbar (Phishark). By performing tests on up to 1230 URLs, we compared its performance to some of the most popular anti-phishing toolbars which make intensive use of blacklists. Then, we determined which heuristics are decisive to differentiate a legitimate site and a phishing site.

This paper is organized as follows: section 2 discusses about related work. Section 3 introduces and analyzes characteristics of phishing URLs and webpages. Section 4 lists the evaluated fields/heuristics with illustrations. Section 5 describes the test setup and Phishark toolbar implementation. Finally, section 6 presents our results and section 7 discusses the solution and results as well as giving some recommendations for webpage developers.

II. Related works

For the last ten years, the proliferation of fake web sites has been so important that many approaches were proposed to counteract phishing attacks. In this section, we discuss about previous works and their associated limits.

Considering that emails are an easy way to reach many users and lure them as much as possible, phishing attacks make extensive use of spam. Therefore, a first approach is to consider that phishing attacks should be blocked at email level. Many proposals address this spam problem [13] but they solve partially the anti-phishing problem as there are other means to redirect Internet users to fraudulent websites.

A second approach looks for new ways to improve authentication exchanges between the user and the server. SSL/TLS protocols [6] are widely used for Internet transactions, but previous studies already discussed about the difficulties of users to distinguish a secure connection [11] [9]. While most of phishing attacks are using unsecured connections, they are effective. Other proposals such as AntiPhish [17] and DOMAntiPhish [25] recommend to

store sensitive information (ie. login, password) and to issue alerts if the user types it into another website. The main problem with this kind of approach is to keep the storage of sensitive information secure (ie. user credentials or reference database for webpages comparison). In addition, this doesn't prevent phishing attacks that are using keyloggers.

A third approach focuses on web page identification by examining the content of the visited page against a previously stored database of signatures for legitimate or fraudulent pages [21] [16] [27]. The drawbacks of this approach are the same as the second approach.

A fourth approach looks for integrating protection into the web browser. Some solutions have been proposed to use a separated trust pop-up (e.g. using a specific pre-shared image as window's background) to enter user credentials, such as Dynamic Security Skins [9]. Like any solution that involves many changes on the server side, a large-scale deployment remains difficult. Finally, other solutions propose to integrate an anti-phishing toolbar into the web browser in order to alert -and/or block- the user in case of suspicious sites. Several previous studies have been conducted on the effectiveness of heuristics and most of them are discussed later in this paper.

It appears that the closest relevant papers to our study were performed by Garera et al. [12] and Ma et al. [19] who focused on URL analysis to identify phishing sites. Ludl et al. [18] paper, which analyzed legitimate and phishing webpages and deduced a classifier from learnt characteristics, can also be compared to our paper as section IV also quantifies content of evaluated heuristics, but they didn't discuss about decisive heuristics.

III. Characteristics of phishing attacks

A typical phishing attack uses several techniques both at URL and HTML content to lure as many users as possible. Previous studies such as works done by Prakash et al. [24] identified some characteristics of phishing URLs, and used them to develop a predictive tool that automatically generates a derived blacklist from a phishing URL. Garera et al. [12] also identified some characteristics of phishing URLs and classified them according to how often they appear in blacklists and whitelists. On the other hand, Pan et al. [23] looked for abnormal behaviors in phishing websites by examining the DOM structure of the webpage.

A. URL

By examining URLs of phishing sites, we can list several technics - which can be combined - to lure the user:

- **Replace the domain name by an IP address**: To hide the domain name of the visited website, some URLs contain an IP address instead of the domain name. e.g. *http://74.220.215.65* instead of *http://volleyballplayerz.com/* is a phishing website that fakes the Natwest bank site.
- **Mispell or derive the domain name**: Some phishing URLs use a domain name very similar to the legitimate one's, by replacing, adding or shifting characters. e.g. *http://verifymyfacebook.700megs.com/Index.html* fakes Facebook website, and *http://www.bhttle.net/* fakes *http://www.battle.net*.

- **Use large domain name and/or suspicious characters in the URL such as @, //**: Some phishing URLs use very large domain names to lure the user. e.g. *http://www.tsv1899benningen-ringen.de/chronik/update/alert/ibclogon.php* or *http://riviera-romagnola.sanmarinostate.com/....* Some of them also use the "@" character to redirect the user to a website different from the one that belongs to the domain name that appears within the address bar [31]. We noticed that the "@" character appears more often in ftp URLs. Moreover, if the path of the URL contains "//" characters, it can be suspected to contain a redirection to another website. e.g. *http://us.battlei.com/?login/login.xmlref=https://kr.battle.net/account/management/index.xml&app=bam*

- **Use short URL**: Some attackers use web services such as TinyURL [2] that shorten URLs, in order to lure URL analysis. By using short aliases, they can automatically redirect to long URLs.

- **Shift the legitimate domain name within the path of the url**: Some phishing URLs use the domain name of the legitimate website within the path of the URL instead of the hostname part e.g. *http://221.165.190.119/www.paypal.com/ws/www/\discretionary-us/webscr.html?cmd=_login-run* fakes Paypal website or *http://album.sibiu-design.\discretionary-info/hsbc.co.uk/1/2/HSBCINTEGRATION/* fakes HSBC website. We also noticed that many phishing URLs with different domain names have the same path structure.

- **Use multiple TLD within the domain name**: Some phishing URLs use several TLDs within the domain name. e.g. *http://user28251.vs.easily.co.uk/* or *http://www.ialp.org.br*

- **Use http instead of https**: As shown within a previous study [18], we noticed that most of phishing websites use unsecured connections. First attackers take advantage of user difficulties to distinguish a secure connection. Second, using a valid certificate is more complex for the attacker and this induces a risk for him/her to be traced. In case the attacker is using invalid certificates, the web browser automatically generates a security alert to the user.

- **Modify encoding of the URL**: Some phishing URLs modify encoding to lure URL analysis [22] and replace some characters by their percent-encoded value. e.g. *http://www.libertyreserve.com.l-en.l-customer-.lunblock.aspx.lid.5b.x7.pq.lr.v7.b1.sub4free.de/%77h%6Fi%73.p%68p?a%63ti%6Fn=l%6F%6Fkup&a%63c%6Funt*

- **Modify the port number**: Some phishing URLs lure the user about the protocol they use, by integrating redirection to a port number different from the one that appears in the URL [5]. e.g. *http://186.97.10.96:8081/https/bancolombia.olb.todo1.com/olb/Init.php* is a phishing URL that redirects to port 8081.

B. Webpage content

By examining HTML content of phishing websites, we first identified some characteristics, such as:

- **Integrate logos and images of the legitimate site.**

- **Move the SSL yellow padlock within the content of the webpage**, instead of the web browser status bar, so that the user believes he's using a secured connection.

- **Integrate security logos** such as VerySign.

- **Use the global structure of the legitimate website** such as sizing and positioning of images, texts and tables.

- **Keep as many legitimate links as possible**. We noticed that many phishing websites modify a minimal part of the legitimate website (i.e. login / password fields to fill in) and keep all other links redirected to the legitimate website. This is probably due to the fact that using a website mirroring tool is an easier and more efficient way to create a phishing site very similar to the legitimate one.

Second, many fields of the HTML structure can be analyzed to determine the legitimacy of the website, by looking for abnormal contents, such as:

- **Title and form fields don't match the domain name.** e.g. a phishing website *(http://www.top-pharmacies.com/ePHARMACIES_languages/English/admin/help/chaseupdate/chaseupdate/chaseupdate/Signon.htm?section=signinpage&=&cookiecheck=yes&=nba/signin)* that fakes Chase bank website uses the title of the legitimate website *Chase Personal Banking Investments Credit Cards Home Auto Commercial Small Business Insurance* in its <title> tag.

- **Links of images, buttons, etc. . . don't match the domain name.** e.g. with the same example as above, some links use the legitimate website *(http://www.chase.com/)* for many tags such as <href>.

We also examined many legitimate websites and we noticed that several of them contain abnormal content in several HTML fields. e.g. Title tag of Hotmail login page *(http://www.hotmail.com)* contains "Sign In".

Finally, we defined heuristics tests based on the above listed characteristics to determine their efficiency.

IV. Evaluated fields

Based on phishing characteristics and previous works [30], [31], [16], we defined and implemented 20 heuristics (see Table I) to evaluate the legitimacy of a website and determine the decisive heuristics.

A. URL analysis

This approach focuses on the analysis of different URL aspects (i.e. number of dots, special characters, port number, IP address. . .), with the objective of identifying irregularities in the URL.

- *Dots and special characters*
  Our detection engine captures the URL from the web browser and counts how many times it finds a dot (.) an at-sign (@) or a double slash (//); then, a score is assigned according to the result obtained. For instance, if the number of dots in the URL is greater than 3, the site is considered as phishing (i.e. *http://ccsts.ccst.gov.cn/manage/upload/tjsj/Customer-login.php*); if it is less than 2, it is considered as legitimate, (i.e. *http://www.ottawa-airport.ca/*); otherwise it is considered as risky (i.e. *http://www.america.gov/fr/subscribe2.html*).
  In addition, every time Phishark finds an IP address instead of a domain name, it considers that we might be accessing a risky site.
  Furthermore, the port number used in the URL - if present - is compared with the claimed protocol. For example, if the protocol displayed in the URL is "http", the port number should be either 80 or 8080. If this occurs, the site is seen as legitimate; if not, it is considered as phishing. In case the port number does not appear in the URL, no action from Phishark is performed.

- *Triplets and Phishing Keywords*
  A Bayesian analysis of whitelists and blacklists using Khiops tool [7] was done at Telecom SudParis in collaboration with Orange [15]. Thanks to it, it was determined that 240 triplets (a set of three alphanumeric characters) appear very often in the domain

TABLE I
IMPLEMENTED HEURISTICS

Group	N	Heuristics
Dots and Special Characters	1	Number of dots (.) in the URL
	2	Number of at-signs (@) in the URL
	3	Number of double slash (//) in the URL
	4	Existence of an IP address in the URL
	5	Port Number in the URL
Triplets and Phishing Keywords	6	Num. of triplets in the domain name
	7	Number of triplets in the path of the URL
	8	Number of phishing keywords in the URL
Country-Code and TLD	9	TLD Evaluation in the domain name
	10	TLD Evaluation in the path of the URL
	11	Country-Code and TLD Comparison
HTML Source-code	12	Title Tag Evaluation
	13	Form Tag Evaluation
	14	Image Tag Evaluation
	15	A href Tag Evaluation
Login Field & HTTPS	16	HTTPS and Login/Password Evaluation
Additional HTML Tags	17	Meta Description Tag Evaluation
	18	Meta Keywords Tag Evaluation
	19	Script Tag Evaluation
	20	Link Tag Evaluation

name of highly suspicious websites. These triplets correspond to the TLD that is most often used by spammers (e.g. .us, .cn, com); part of the name of some companies such as msn, ban (short for bank), bay (short for e-bay), the word sex and many others. The detection engine evaluates the triplets included in the URL, and based on the frequency of their appearance, it assigns a score.
Similarly, as previously identified within Garera et al. study [12], we selected some words (such as "http", "login", "paypal") that often appear in phishing URLs and we implemented a function to verify their appearance in the URL apart from the domain name. Consequently, the higher the number of phishing keywords in the URL, the higher the probability of being phished. As a result, some URLs such as *http://www.neural-net.ca/store/images/webscr/1/www.paypal.co.uk/* are considered as a risky site.

- *Country-Code and TLD*
  By analyzing phishing URLs and domains, McGrath et al. study [20] identified that most of the phishing websites are not hosted in the country claimed by their TLD. In addition, some APWG - Anti-Phishing Working Group - reports determined that a substantial number of phishing sites were hosted by few countries during the year 2009, with the U.S. as the head of the top ten. Based on this information, we classified the countries into two groups (according to their percentage of incidence) in order to evaluate the TLD of each URL.
  Moreover, we integrated a Firefox plug-in called World IP [4], whose main goal is to help identifying exact geographic location of an IP address. As such, we designed an algorithm that compares the TLD (that appears in the URL) with the hosting country code (provided by the World IP extension). In case of matching, the site is considered as legitimate.

B. HTML analysis

The analysis of the HTML tags goes beyond a simple URL evaluation by accessing the page's source code and comparing it against the information displayed by the site (URL, domain name).

- *HTML Source-Code*
  After analyzing some websites, we concluded that many phishing sites do not use the domain names in the HTML tags. Instead, most of them leave empty spaces in this area or keep information from the legitimate website. Then, it results into an abnormal behavior because the domain name of the phishing URL and the content of HTML tags don't match [8], [23].
  We decided to analyze four HTML tags: the <title> tag, which contains the document title; the <form> tag, which contains forms requesting user inputs (i.e. login, password, credit card number, etc...); the <img> tag, which embeds an image in an HTML page; and the <a href> tag, which is an anchor that creates a link to another document.
  The algorithm developed analyses the content of the HTML tags and if it matches with the domain name, the site is considered as legitimate; otherwise, it is seen as phishing.

- *Login Field and HTTPS*
  Based on our study of the HTML structure and previous research [18], we identified that most of the banking and e-commerce websites (e.g. Paypal, etc...) that request users to log in with IDs and passwords are secured with the https protocol. Therefore,

Fig. 1. Phishark anti-phishing toolbar

we developed an algorithm that checks if the page asks for login or password and then, it checks if it is secured with the https protocol. If this occurs, the system considers the page as legitimate; otherwise as phishing.
However, we found that many social network sites (i.e. Facebook, Viadeo, Twitter, etc.) and several mailing sites (i.e. hotmail, Voila, etc.) use the http protocol to render the authentication page to their users. We noticed that they use SSL protocol for a very short time to authenticate the user. This authentication - which is often invisible by the user - makes the algorithm to recognize the site as phishing.

- *Additional HTML Tags*
  In order to find other criteria that could help detecting phishing sites, we conducted a study on the HTML source-code in which we found that, very frequently, legitimate sites use <Meta> tags to provide metadata about the HTML document; <script> tags to define a client-side script; and <link> tags to retrieve other web resources. In most of the cases, the information of these tags matches the domain name. The algorithm developed first checks if the document contains the tags (<meta>, <script>, <link>), and then, it compares the extracted information with the domain name. In case they match, the site is considered as legitimate. Note that because all these tags are not mandatory within the HTML webpage, a non-matching content cannot be used to consider the visited website as suspicious.

V. Test-bed conditions

One of the most critical phases in the development of our toolbar - both in terms of whitelist and blacklists - was to calibrate the score assigned to each criterion as it can sometimes affect one list positively and the other negatively. As a consequence, several preliminary tests were conducted to determine decisional thresholds of each heuristic and evaluate the impact of each of them on the overall rating. This analysis allowed us to calibrate our toolbar in order to limitate both false positive and false negative results.

Furthermore, it is important to highlight that dataset - composed of hundreds of legitimate and phishing URLs - used to calibrate Phishark is different from the dataset of 500 legitimate sites and 730 phishing sites (described in section V-A) used to evaluate its effectiveness.

A. Whitelists and Blacklists

The experimental part of the project focused on the evaluation of 500 URLs for the whitelist, obtained as follows: 164 URLs from the Google top 1000 most visited sites, 125 URLs from the Alexa top 500 Global sites, 150 URLs from Netcraft database, 50 URLs from banking websites all around the world and the rest of the URLs were taken from a pharming study conducted at Telecom SudParis. 730 URLs for the blacklist were obtained as follows: 549 URLs from Phishtank website [1] (URLs identified as valid phishing sites) and 181 URLs from the Anti-Phishing Working Group. Both lists contain short URLs with domain names only, as well as long URLs with long paths, from different locations in the world, developed in different languages and using different TLD's in the domain name.

Since most of the URLs on the blacklist have a very short lifetime (Mc Grath et al. calculated that the average lifetime of a phishing site is 3 days, 31 minutes and 8 seconds [20]), it was important to obtain the most updated URLs declared as phishing to conduct the tests. However during the evaluation, we realized that some URLs were not longer available online, making us perceive that the lifetime of some phishing sites is sometimes reduced to few hours. As a result, the blacklist evaluation process was accomplished step-by-step from July 30, 2010 to August 6, 2010.

Most of the URLs we selected were reported as phishing sites for several hours. In addition, we waited 3 hours from the moment we collected the URLs to the moment we started the evaluation in all the machines, to take into account that updates of blacklist-based anti-phishing toolbars are not instantaneous.

B. Phishark toolbar implementation

Our Phishark toolbar was designed based on a performance evaluation study over several toolbars [14] and the determination, by researchers, that active notifications are highly recommended to warn the end-user on the security aspects of a website [10]. Phishark is developed as an add-on[1] for Mozilla Firefox (the first free and open source browser most worldwide used as of July 2010 [3]). It adopted most of the important aspects of an anti-phishing toolbar i.e., clearness and consistency of alert messages, explanation regarding the results, active blocking message, etc. (See Figure 1 and Figure 2).

[1]Phishark is developped using XUL language, for the visual interface, and Javascript for the detection engine.

TABLE II
WHITELIST RESULTS

N	Toolbar	True Positives and Neutrals		False Positives		No Information	
		N	%	N	%	N	%
1	Netcraft	500	100.0	0	0.0	0	0.0
2	Internet Explorer	499	99.8.0	0	0.0	1	0.2
3	Mozilla Firefox	496	99.2	4	0.8	0	0.0
4	Phishark	488	97.6	12	2.4	0	0.0
5	Web Of Trust	480	96.0	9	1.8	11	2.2

C. Selection of the anti-phishing solutions

Based on previous studies [30], [28], [14], the four best and most updated anti-phishing solutions on the market were selected to be compared with our anti-phishing toolbar. The following paragraphs detail the selected anti-phishing solutions:

- *Mozilla Firefox v.3.6.7* web browser contains built-in phishing and malware protection based on a third party service (Google) that provides an updated blacklist of malicious URLs every 30 minutes. Zhang et al. [30] suspected that Mozilla Firefox also uses heuristics in its detection process.
- *Netcraft Anti-Phishing Toolbar v.1.4.1.5*, installed as an extension in the web browser, compares suspicious URLs containing characters against a blacklist stored within Netcraft servers. In addition, the tool uses some heuristics (such as IP address, domain name, hosting country code, reverse DNS, etc.).
- *Internet Explorer v.8.0.601.18702 web browser* has a built-in anti-phishing protection called "SmartScreen Filter" that employs heuristics tests (e.g. website analysis, downloaded file verification, etc...) as well as a blacklist in order to check the website security features.
- *Web of Trust (WOT) v.20100503*, installed as an extension in the web browser, uses some heuristics (e.g. site's popularity, rank, server location and reputation...) as well as a black list powered by a global community of millions of users to verify the security features of a website.

D. PCs configuration

In order to properly compare the performance of our toolbar with other anti-phishing solutions, it was necessary to test every URL with each toolbar simultaneously., As such, we installed each solution in a separate PC DELL with the following characteristics: Intel (R) Core (TM) 2 Duo Processor, 0,99 Go of RAM and Microsoft Windows XP Professional, version 2002, Service Pack 3. All the anti-phishing solutions, except Internet Explorer, were installed as add-ons in Mozilla Firefox v.3.6.7.

Fig. 2. Warning message

VI. TEST RESULTS

This section provides the results obtained by evaluating up to 1230 URLs that were selected to determine the legitimacy of websites and decisive heuristics, out of which 500 URLs were obtained from a whitelist and 730 URLs were obtained from a blacklist. Table II and Table III show our results in comparison with the four anti-phishing solutions described in the previous section. Figure 3 shows our results regarding the decisive heuristics.

A. Heuristics efficiency : toolbars performances

- *Whitelist results*
  For the evaluation of the whole whitelist, not surprisingly, Netcraft performed the best among all the other anti-phishing solutions, since many of the URLs used in the evaluation were obtained from the Netcraft database. Internet Explorer did not detect one site, displaying an error message ("server could not be found").
  Mozilla Firefox wrongly blocked four legitimate websites because of certificate issues, requesting the user to add an exception for the proposed security certificate. Regarding the WOT performance, 1.8% of the URLs were wrongly detected by showing a poor or very poor reputation; and for 2.2% of cases, the toolbar was unable to identify the site, giving no information about the visited page.
  Regarding Phishark performance, even though the toolbar does not use any whitelist for its detection process, the performance was very satisfactory, reaching 97.6% for the true positives and neutrals. However, 2.4% of sites were wrongly considered as phishing; mainly due to the fact that some legitimate websites didn't fill in the HTML source-code tags or because they added information that didn't relate to the domain name.

TABLE III
BLACKLIST RESULTS

N	Toolbar	True Negatives and Neutrals		False Negatives		No Information	
		N	%	N	%	N	%
1	Phishark	490	98.0	10	2.0	0	0.0
2	Netcraft	453	90.6	43	8.6	4	0.8
3	Mozilla Firefox	423	84.6	77	15.4	0	0.0
4	Internet Explorer	406	81.2	94	18.8	0	0.0
5	Web of Trust	386	77.2	25	5.0	89	17.8

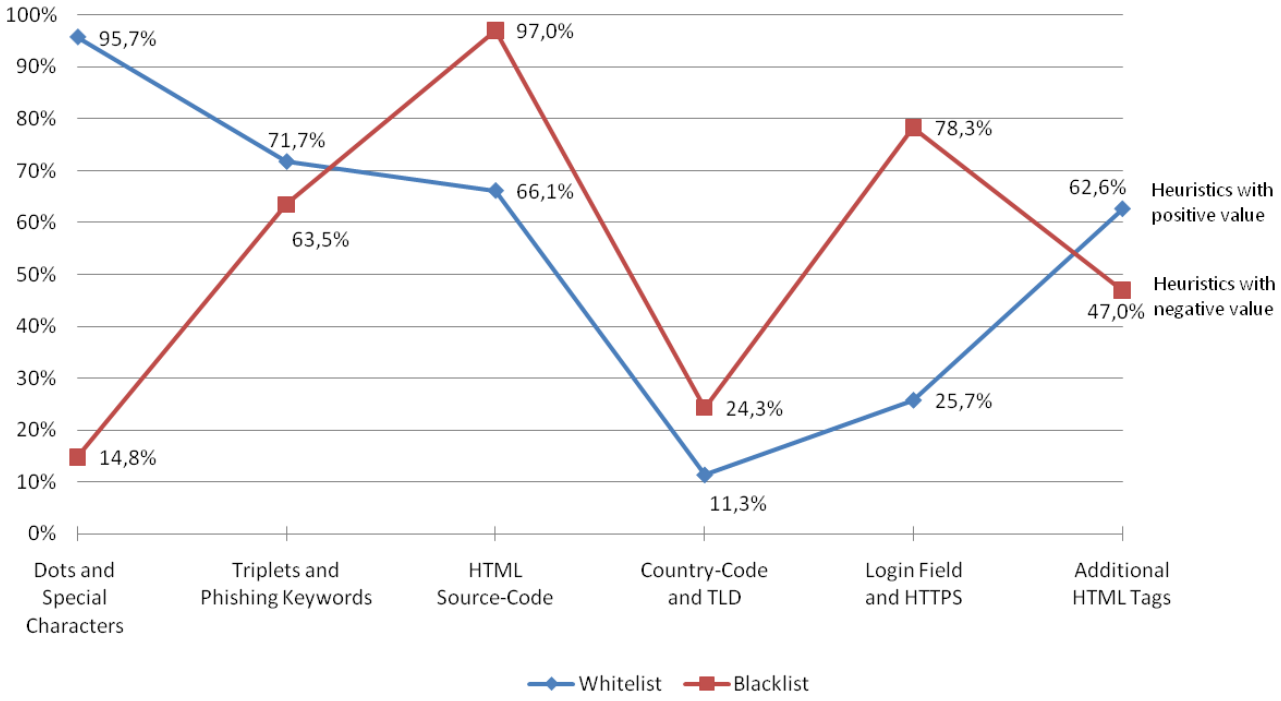

Fig. 3. Decisive heuristics

- *Blacklists results*
  For the evaluation of 500 URLs of the blacklist, Phishark performed the best among the five evaluated anti-phishing solutions, with a detection rate of 98% (see Table III). This high rate is probably due to the heuristics used in our solution rather than blacklists. During the evaluation process, we realized that the performance of the other solutions was negatively affected when the URLs were obtained as soon as they were reported to the anti-phishing databases. There is always a minimum synchronization delay between the time a site is reported as phishing and the time it is updated on the blacklist, thus causing the browser and/or the toolbar to wrongly detect a phishing site as legitimate, and allowing users to submit their personal and financial information. For this reason, we decided to start the evaluation process after getting a group of 100 URLs, taking an average of three hours to complete this process.
  In terms of true negatives, Netcraft performance is satisfactory, with an average of 90.6%, making it the second best anti-phishing detection engine among the five tested ones. Mozilla Firefox is ranked third, with 84.6% and Internet Explorer is ranked fourth with a rate of 81.2%. WOT gets the lowest performance during the evaluation process, with an average of 77.2% of true negatives.
  Blacklists used by Mozilla Firefox and Internet Explorer were detected to be likely different. In several cases, a site considered as phishing for one browser was considered as legitimate for the other. Moreover, we could perceive from the conducted tests that Mozilla Firefox is faster and more reliable than Internet Explorer when it comes to load the site. It takes sometimes a couple of seconds for IE to display the requested page, whereas the same page takes a fraction of a second for Firefox to be loaded or declared as fake.
  Unlike IE and Firefox whose rate on the false negatives was 15.4% and 18.8% respectively, WOT per-

formed better (5.0%). However, there is a high number of sites for which WOT was unable to provide any information, probably because it could not find the URL, neither on its whitelist nor on its blacklist. As such, the overall performance of WOT is ranked the poorest among the five anti-phishing solutions.

B. Decisive heuristics

- *Whitelist results*
  Regarding the effectiveness of heuristics to identify a legitimate site, our study over 230 URLs for the whitelist determined that the 4 more useful heuristics are: Dots and Special Characters, Triplets and Keywords, HTML Source-Code, and Additional HTML Tags (see Figure 3). The Dots and Special Characters heuristics, for instance, contribute in 95.7% of the cases with the appropriate detection on the whitelist. Similarly, the Triplets and Phishing Keywords heuristics identify as legitimate 71.7% of the sites; heuristics for the HTML Source-Code properly detects legitimate sites in 66.1% of the cases and the heuristics for the Additional HTML Tags provides a 62.6% of accuracy in the detection of legitimate websites. The heuristics of Login Field and HTTPS, as well as the Country-Code and TLD perform very poorly on the detection of legitimate sites with a rate of 25.7% and 11.3% respectively.

- *Blacklist results*
  Regarding the heuristics, and contrary to the results of the whitelist, the study conducted with 230 additional URLs for the blacklist determined that the HTML Source-Code is the one that best identifies phishing sites (97.0% of the cases); followed by the Login Field and HTTPS heuristics, with a detection rate of 78.3%; the Triplets and Phishing Keywords heuristics are ranked third, with a detection rate of 63.5% and the Additional HTML Tags are ranked fourth with an average of 47.0% (see Figure 3). The heuristics of Country-Code and TLD, as well as the Dots and Special Characters performed very poorly on the detection of phishing sites with a rate of 24.3% and 14.8% respectively.

VII. Conclusion

First, the results from the previous section demonstrate that combining URL-based and HTML-based heuristics is pretty effective to differentiate legitimate from phishing sites. In addition, it avoids the blacklist drawbacks which need time to update as well as an absolute matching for the URL.

However, many improvements can be performed in our heuristics-based-only detection engine in order to increase its accuracy to differentiate legitimate from phishing sites. For instance, we can go one step further by verifying the port number under which the website is connected (even if it doesn't appear in the URL) and doing comparison of it with the protocol displayed in the URL. Similarly, since most of the phishing sites have a 3-days average online lifetime [20], an additional function that evaluates the site lifetime and history can be integrated.

The average evaluation time of our Phishark engine is about 2 to 3 seconds. This keeps our solution quite proactive since the user is blocked - by an active warning message - before he can type in all his credentials on a suspicious website.

Phishark also integrates an "Authorize" button to allow the user to define his/her own whitelist. This can significantly reduce the false positive detection rate for whitelist and improve the evaluation time as well (for moderate sized whitelist).

Second, test results demonstrate that all heuristics are not equal in identifying legitimate and phishing sites. Especially, it appears that Dots and Special Characters (URL approach) are decisive to detect legitimate sites, while the HTML Source-Code and Login Field-HTTPS heuristics (HTML approach) are essential for detecting phishing sites (see Figure 3).

Based on these results, another way to make heuristics-based detection more efficient would be to recommend web developers to properly fill in all the different fields of the HTML Source-Code, by clearly identifying every HTML tag with some information related to their domain name. This could be very helpful to improve the whitelist evaluation. Note that an attacker can use the same method to elaborate better imitations of legitimate sites, but part of the HMTL Source-code evaluation will still succeed to detect phishing sites. Phishers are used to copy the whole legitimate site, by using mirroring tools, and making only small changes on either the hyperlinks or the domain names to make their attacks more undetectable (e.g. *http://paypoi.110mb.com/index.htm/* for a Paypal phishing site which uses most of the legitimate hyperlinks from Paypal such as *https://www.paypalobjects.com/...*), but it will require an additional effort for phishers to succeed in deceiving detection tools that rely on this kind of heuristics evaluation.

Finally, the main weaknesses of our approach are inherent to the web browser vulnerabilities and the detection engine protection. The next steps of improvement will aim at protecting the storage of the whitelist file as well as the detection engine thresholds and functions. Our anti-phishing toolbar Phishark is available on request.

References

[1] "PhishTank Home," http://www.phishtank.com/. [Online]. Available: http://www.phishtank.com/

[2] "TinyURL," http://tinyurl.com/. [Online]. Available: http://tinyurl.com/

[3] "Web browser market share." [Online]. Available: http://statowl.com/web_browser_market_share_trend.php?1=1&timeframe=last_6&interval=month&chart_id=4&fltr_br=&fltr_os=&fltr_se=&fltr_cn=&chart_id=

[4] "WorldIP free geolocation database, service and tools," http://www.wipmania.com/en/. [Online]. Available: http://www.wipmania.com/en/

[5] "Phishing attack trends report - q4 2009," Mar. 2010. [Online]. Available: http://www.apwg.org/reports/apwg_report_Q4_2009.pdf

[6] M. Badra, S. El-Sawda, and I. Hajjeh, "Phishing attacks and solutions," in *Proceedings of the 3rd international conference on Mobile multimedia communications*, vol. 329. Nafpaktos, Aitolokarnania, Greece: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Brussels, Belgium, 2007, p. Article No. 42.

[7] M. Boulle, "Software for data mining : France telecom scoring tool," http://www.khiops.com/. [Online]. Available: http://www.khiops.com/

[8] J. Chen and C. Guo, "Online detection and prevention of phishing attacks," Beijing, China, Oct. 2006.

[9] R. Dhamija and J. Tygar, "The battle against phishing: Dynamic scurity skins," in *Proceedings of the 2005 symposium on Usable privacy and security*, vol. 93. Pittsburg, Pennsylvania, USA: ACM, Jul. 2005, pp. 77–88.

[10] S. Egelman, L. Cranor, and J. Hong, "You've been warned: an empirical study of the effectiveness of web browser phishing warnings," in *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*. Florence, Italy: ACM, Apr. 2008, pp. 1065–1074. [Online]. Available: www.guanotronic.com/~serge/chi1210-egelman.pdf

[11] B. Friedman, D. Hurley, D. C. Howe, E. Felten, and H. Nissenbaum, "Users' conceptions of web security: a comparative study." Minneapolis, Minnesota, USA: ACM, Apr. 2002, pp. 746–747.

[12] S. Garera, N. Provos, M. Chew, and A. D. Rubin, "A framework for detection and measurement of phishing attacks." Alexandria, Viriginia, USA: ACM, 2007, pp. 1–8.

[13] S. Gastellier-Prevost, "Le spam," *Editions Techniques de l'Ingenieur*, vol. H5450, pp. 1–22, Apr. 2009.

[14] S. Gastellier-Prevost, N. Abid, and M. Laurent, "Anti-phishing toolbars evaluation," Jun. 2009.

[15] B. B. Hammouda, H. Bouraoui, D. Migault, and S. Gastellier-Prevost, "Quand un operateur part a la peche," Jun. 2009.

[16] M. Hara, A. Yamada, and Y. Miyake, "Visual similarity-based phishing detection without victim site information." Nashville, Tennessee, USA: IEEE, Apr. 2009, pp. 30–36.

[17] E. Kirda and C. Kruegel, "Protecting users against phishing attacks," vol. 1. Edinburgh, Scotland: IEEE Computer Society Press, Jul. 2005.

[18] C. Ludl, S. McAllister, E. Kirda, and C. Kruegel, "On the effectiveness of techniques to detect phishing sites," in *Proceedings of the 4th international conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, vol. Lecture Notes In Computer Science; Vol. 4579. Lucerne, Switzerland: Springer-Verlag Berlin, Heidelberg, 2007, pp. 20–39.

[19] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: Learning to detect malicious web sites from suspicious URLs," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. Paris, France: ACM, Jul. 2009, pp. 1245–1254.

[20] D. K. McGrath and M. Gupta, "Behind phishing: An examination of phisher modi operandi," in *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*. San Francisco, California, USA: USENIX Association Berkely, CA, USA, 2008, p. Article No. 4.

[21] E. Medvet, E. Kirda, and C. Kruegel, "Visual-Similarity-Based phishing detection," in *Proceedings of the 4th international conference on Security and privacy in communication networks*. Istanbul, Turkey: ACM, Sep. 2008, p. Article No. 22.

[22] G. Ollman, "The phishing guide - understanding and preventing phishing attacks," Sep. 2004.

[23] Y. Pan and X. Ding, "Anomaly based web phishing page detection," in *Proceedings of the 22nd Annual Computer Security Applications Conference*. IEEE Computer Society Press, Dec. 2006, pp. 381–392.

[24] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, "PhishNet: predictive blacklisting to detect phishing attacks," in *IEEE INFOCOM Proceedings*. San Diego, California, USA: IEEE, Mar. 2010, pp. 1–5.

[25] A. P. E. Rosiello, E. Kirda, C. Kruegel, and F. Ferrandi, "A layout-similarity-based approach for detecting phishing pages." Nice, France: IEEE, Sep. 2007, pp. 454–463.

[26] S. Sheng, B. Wardman, G. Warner, L. Cranor, J. Hong, and Z. Chengsan, "An empirical analysis of phishing blacklists," Moutain View, California, USA, Jul. 2009.

[27] L. Wenyin, G. Huang, L. Xiaoyue, Z. Min, and X. Deng, "Detection of phishing webpages based on visual similarity," in *Special interest tracks and posters of the 14th international conference on World Wide Web*. Chiba, Japan: ACM, May 2005, pp. 1060–1061.

[28] M. Wu, R. C. Miller, and S. L. Garfinkel, "Do security toolbars actually prevent phishing attacks?" in *Proceedings of the SIGCHI conference on Human Factors in computing systems*. Montreal, Quebec, Canada: ACM, 2006, pp. 601–610.

[29] W. D. Yu, S. Nargundkar, and N. Tiruthani, "A phishing vulnerability analysis of web based systems." Marrakech, Maroc: IEEE, Jul. 2008, pp. 326–331.

[30] Y. Zhang, S. Egelman, L. Cranor, and J. Hong, "Phinding phish: Evaluating Anti-Phishing tools," in *Proceedings of the 14th Annual Network & Distributed System Security Symposium*, San Diego, California, USA, Mar. 2007.

[31] Y. Zhang, J. Hong, and L. Cranor, "CANTINA : A Content-Based approach to detecting phishing web sites," in *Proceedings of the 16th international conference on World Wide Web*. Banff, Alberta, Canada: ACM, May 2007, pp. 639–648.