Web Phishing Detection: A Literature Survey

Abstract :

This article surveys the literature on the detection of phishing attacks. Phishing attacks target vulnerabilities that exist in systems due to the human factor. Many cyber attacks are spread via mechanisms that exploit weaknesses found in end- users, which makes users the weakest element in the security chain. The phishing problem is broad and no single silver-bullet solution exists to mitigate all the vulnerabilities effectively, thus multiple techniques are often implemented to mitigate specific attacks. This paper aims at surveying many of the recently proposed phishing mitigation techniques. A high-level overview of various categories of phishing mitigation techniques is also presented, such as: detection, offensive defense, correction, and prevention, which we belief is critical to present where the phishing detection techniques fit in the overall mitigation process.

Index Terms—phishing;

social engineering; phishing detection; security; email classification;

1.INTRODUCTION

Phishing is a social engineering attack that aims at ex- ploiting the weakness found in system processes as caused by system users. For example, a system can be technically secure enough against password theft, however unaware end users may leak their passwords if an attacker asked them to update their passwords via a given Hypertext Transfer Protocol (HTTP) link, which ultimately threatens the overall security of the system.

Moreover, technical vulnerabilities (e.g. Domain Name Sys- tem (DNS) cache poisoning) can be used by attackers to construct far more persuading socially-engineered messages (i.e. use of legitimate, but spoofed, domain names can be far more persuading than using different domain names). This makes phishing attacks a layered problem, and an effective mitigation would require addressing issues at the technical and human layers.

Since phishing attacks aim at exploiting weaknesses found in humans (i.e. system end-users), it is difficult to mitigate them. For example, as evaluated in [1], end-users failed to detect 29% of phishing attacks even when trained with the best performing user awareness program. On the other hand, software phishing detection techniques are evaluated against bulk phishing attacks, which makes their performance practically unknown with regards to targeted forms of phishing attacks. These limitations in phishing mitigation techniques have practically resulted in security breaches against several organizations including leading information security providers [2], [3].

This survey begins by defining the phishing problem in Section II, presenting background and related works in Section III, and an overview of phishing mitigation approaches in Section IV which also presents the taxonomy of various mitigation techniques, such as: detection, prevention, and correction techniques. Subsequent sections in this survey will then focus on phishing detection techniques, which include detection techniques through user awareness, as presented in Section VI, and a number of software techniques. The software-based detection techniques are: blacklists in Section VII, heuristic detection techniques in Section VIII, visual similarity detection techniques in Section IX, and data mining detection techniques in Section X. The evaluations of the surveyed detection techniques are presented in Section XII, followed by the lessons that we have learned in Section XIII. The conclusion is drawn in Section XIV.

II.DEFINITION

The definition of phishing attacks is not consistent in the literature, which is due to the fact that the phishing problem is broad and incorporates varying scenarios. For example, according to PhishTank:

“Phishing is a fraudulent attempt, usually made through email, to steal your personal information”

PhishTank’s definition holds true in a number of scenarios which, roughly, cover the majority of phishing attacks (al- though no accurate studies have been made to reliably quantify this). However, the definition limits phishing attacks to stealing personal information, which is not always the case. For example, a socially engineered message can lure the victim to install a Man in the Browser (MITB) malware (e.g. in forms of web browser ActiveX components, plugins or email attachments) which would in turn transfer money to the attacker’s bank account, whenever the victim logs in to perform his/her banking tasks, without the need to steal the victim’s personal information. Thus we consider that PhishTank’s definition is not broad enough to encompass the whole phishing problem.

Another definition is provided by Colin Whittaker et. Al.

“We define a phishing page as any web page that, without permission, alleges to act on behalf of a third party with the intention of confusing viewers into performing an action with which the viewer would only trust a true agent of the third party”

The definition by Colin Whittaker et. al. aims to be broader than PhishTank’s definition in a sense that attackers goals are no longer restricted to stealing personal information from victims. On the other hand, the definition still restricts phishing attacks to ones that act on behalf of third parties, which is not always true.

For example phishing attacks may communicate socially engineered messages to lure victims into installing MITB malware by attracting the victims to websites that are supposed to deliver safe content (e.g. video streaming). Once the malware (or crimeware as often named by Anti-Phishing Working Group (APWG)2) is installed, it may log the victim's keystrokes to steal their passwords. Note that the attacker in this scenario did not claim the identity of any third party in the phishing process, but merely communicated messages with links (or attachments) to lure victims to view videos or multimedia content. In order to address the limitations of the previous definitions above, we consider phishing attacks as semantic attacks which use electronic communication channels (such as EMails, HTTP, SMS, VoIP, etc. . .) to communicate socially engineered messages to persuade victims to perform certain actions (without restricting the actions) for an attacker's benefit (without restricting the benefits).

Definition 1: Phishing is a type of computer attack that communicates socially engineered messages to humans via electronic communication channels in order to persuade them to perform certain actions for the attacker's benefit. For example, the performed action (which the attacker persuades the victim to perform it) for a PayPal user is submitting his/her login credentials to a fake website that looks similar to PayPal. As a perquisite, this also implies that the attack should create a need for the end-user to perform such action, such as informing him that his/her account would be suspended unless he logs in to update certain pieces of information [5].

III. BACKGROUND

A. History According to APWG, the term phishing was coined in 1996 due to social engineering attacks against America On-line (AOL) accounts by online scammers. The term phishing comes from fishing in a sense that fishers (i.e. attackers) use a bait (i.e. socially-engineered messages) to fish (e.g. steal personal information of victims). However, it should be noted that the theft of personal information is mentioned here as an example, and that attackers are not restricted by that as previously defined in Section II. The origins of the phishing replacement of the character f in fishing is due to the fact that one of the earliest forms of hacking was against telephone networks, which was named Phone Phreaking. As a result, phishing became a common hacking character replacement of f. According to APWG, stolen accounts via phishing attacks were also used as a currency between hackers by 1997 to trade hacking software in exchange of the stolen accounts. Phishing attacks were historically started by stealing AOL accounts, and over the years moved into attacking more profitable targets, such as on-line banking and e-commerce services. Currently, phishing attacks do not only target system endusers, but also technical employees at service providers, and may deploy sophisticated techniques such as MITB attacks.

B. Phishing Motives

According to Weider D. et. al. [6], the primary motives behind phishing attacks, from an attacker's perspective, are:

• Financial gain: phishers can use stolen banking credentials to their financial benefits.

• Identity hiding: instead of using stolen identities directly, phishers might sell the identities to others whom might be criminals seeking ways to hide their identities and activities (e.g. purchase of goods).

• Fame and notoriety: phishers might attack victims for the sake of peer recognition.

C. Importance

According to APWG, phishing attacks were in a raise till August, 2009 when the all-time high of 40,621 unique3 phishing reports were submitted to APWG. The total number of submitted unique phishing websites that were associated with the 40,621 submitted reports in August, 2009 was 56,362. As justified by APWG, the drop in phishing campaign reports in the years 2010 and 2011 compared to that of the year 2009 was due to the disappearance of the Avalanche gang4 which, according to APWG's 2nd half of 2010 report, was responsible for 66.6% of world-wide phishing attacks in the 2nd half of 2009 [7]. In the 1st half of the year 2011, the total number of submitted phishing reports to APWG.

was 26,402, which is 35% lower than that of the peak in the year 2009 [8]. However, according to APWG, the drop in phishing attacks was due to the switch in the activities of the Avalanche gang from traditional phishing campaigns into malware-based phishing campaigns. In other words, the Avalanche gang did not stop phishing campaigns but rather switched their tactics toward malware-based phishing attacks (which still requires electronic communication channels and social engineering techniques to deliver malware). Among the various types of malware that are used in phishing attacks, Trojan horses software seem to be in a raise, and are the most popular type of malware deployed by phishing attacks. According to APWG, Trojans software contributed 72% of the total malware detected in the 1st half of 2011, from the previous value of 55% in the 2nd half of 2010. It is also important to note that although the number of phishing attack reports dropped since the peak in 2009, the number of phishing attack reports are still high ,compared to that of the 2nd half of 2008 which faced an average of 28,916 unique reports, and ranged between 22,000 and 26,000 of unique reports each month in the 1st half of 2011. On the other hand, the 2nd half of 2011 saw a raise in phishing reports and websites, which seems to be correlated with holidays season [9] as depicted in Figures 1 and 2. Which is further amplified when knowing that each phishing campaign can be sent to thousands or even millions of users via electronic communication channels. The year 2011 saw a number of notable spear phishing attacks against well known security firms such as RSA [10] and HB Gary [2], which resulted in further hacks against their clients such as RSA's client Lockheed Martin [3]. This shows that the dangers of phishing attacks, or security vulnerabilities due to the human factor, are not limited to the naivety of endusers since technical engineers can also be victims.

Minimizing the impact of phishing attacks is extremely important and adds great value to the overall security of an organization.

D. Challenges

Because the phishing problem takes advantage of human ignorance or naivety with regards to their interaction with elec tronic communication channels (e.g. E-Mail, HTTP, etc. . .), it is not an easy problem to permanently solve. All of the proposed solutions attempt to minimize the impact of phishing attacks. From a high-level perspective, there are generally two commonly suggested solutions to mitigate phishing attacks:

• User education; the human is educated in an attempt to enhance his/her classification accuracy to correctly identify phishing messages, and then apply proper actions on the correctly classified phishing messages, such as reporting attacks to system administrators.

• Software enhancement; the software is improved to better classify phishing messages on behalf of the human, or provide information in a more obvious way so that the human would have less chance to ignore it.

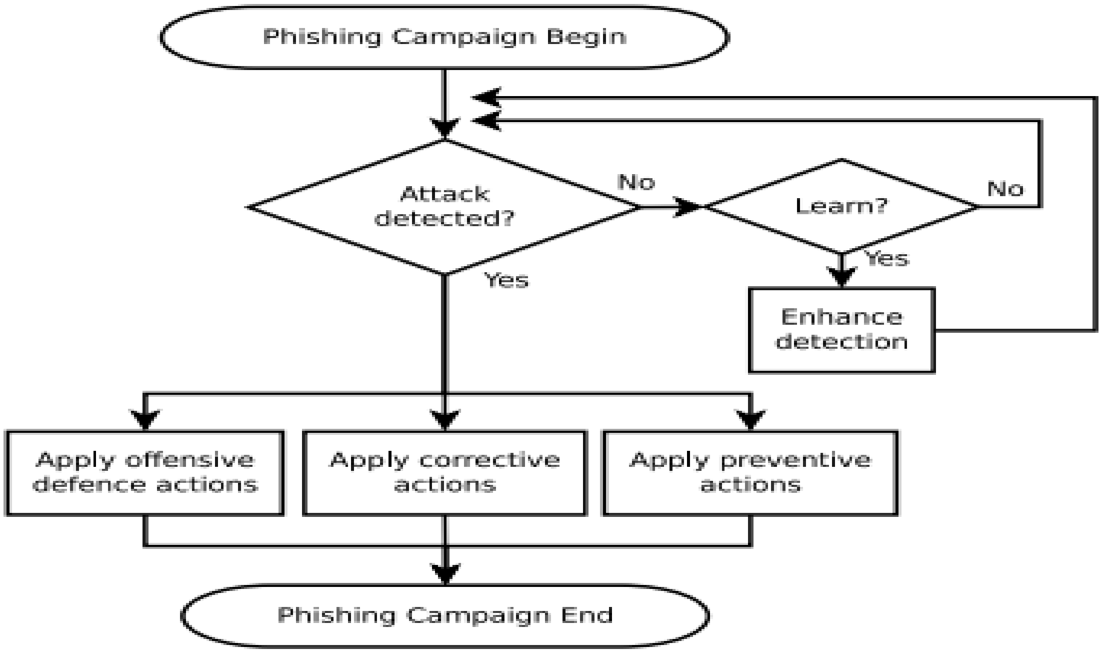
The challenges with both of the approaches are:

• Non-technical people resist learning, and if they learn they do not retain their knowledge permanently, and thus training should be made continuous. Although some researchers agree that user education is helpful [1], [11], [12], a number of other researchers disagree [13], [14]. Stefan Gorling [13] says that: “this is not only a question of knowledge, but of utilizing this knowledge to regulate behavior. And that the regulation of behavior is dependent on many more aspects other than simply the amount of education we have given to the user”

• Some software solutions, such as authentication and security warnings, are still dependent on user behavior. If users ignore security warnings, the solution can be rendered useless.

• Phishing is a semantic attack that uses electronic communication channels to deliver content with natural languages (e.g. Arabic, English, French, etc. . .) to persuade victims to perform certain actions.

The challenge here is that computers have extreme difficulty in accurately understanding the semantics of natural languages. A notable attempt is E-mail-Based Intrusion Detection System (EBIDS) [15], which uses Natural Language Processing (NLP) techniques to detect phishing attacks, however its performance evaluation showed a phishing detection rate. The life-cycle of phishing campaigns from the perspective of antiphishing techniques. of only 75%. In our opinion, this justifies why most well-performing phishing classifiers do not rely on NLP techniques. IV. MITIGATION OF PHISHING ATTACKS: AN OVERVIEW Due to the broad nature of the phishing problem, we find important to visualize the life-cycle of the phishing attacks, and based on that categorize anti-phishing solutions. Based on our review of the literature, we depict a flowchart describing the life-cycle of phishing campaigns from the perspective of anti-phishing techniques, which is intended to be the most comprehensive phishing solutions flowchart.



When a phishing campaign is started (e.g. by sending phishing emails to users), the first protection line is detecting the campaign. The detection techniques are broad and could incorporate techniques used by service providers to detect the attacks, end-user client software classification, and user awareness programs. More details are in Section IV-A. The ability to detect phishing campaigns can be enhanced whenever a phishing campaign is detected by learning from such experience. For example, by learning from previous phishing campaigns, it is possible to enhance the detection of future phishing campaigns. Such learning can be performed by a human observer, or software (i.e. via a machine learning algorithm). Once the phishing attack is detected, a number of actions could be applied against the campaign. According to our review of the literature, the following categories of approaches exist: • Offensive defense — these approaches aim to attack phishing campaigns to render them less effective. This approach is particularly useful to protect users that have submitted their personal details to attackers. More details are in Section IV-B. • Correction — correction approaches mainly focus on taking down the phishing campaign. In case of phishing websites, this is achieved by suspending the hosting account or removing phishing files. More details are in Section IV-C. • Prevention — phishing prevention methods are defined differently in the literature depending on the context. In this survey, the context is attempting to prevent attackers from starting phishing campaigns in the future. More details are in Section IV-D.

However, if the phishing campaign is not detected (let it be detected by a human or a software classifier), then none of these actions can be applied. This emphasizes the importance of the detection phase.

A. Detection Approaches

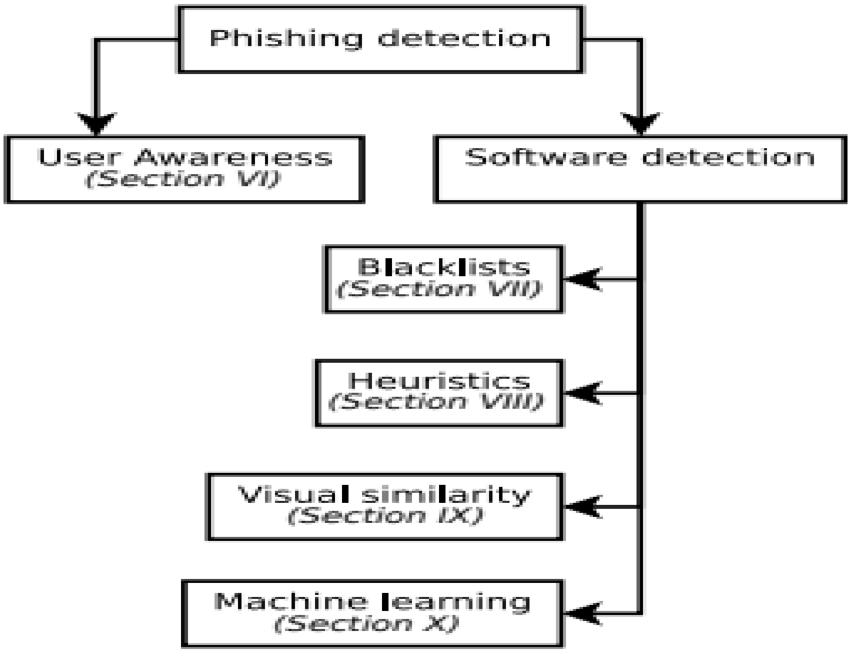
In this survey, we consider any anti-phishing solution that aims to identify or classify phishing attacks as detection solutions.

This includes:

- User training approaches — end-users can be educated to better understand the nature of phishing attacks, which ultimately leads them into correctly identifying phishing and non-phishing messages. This is contrary to the categorization in [16] where user training was considered a preventative approach. However, user training approaches aim at enhancing the ability of end-users to detect phishing attacks, and thus we categorize them under "detection". Further discussions on the human factor are presented in Section VI.

- Software classification approaches — these mitigation approaches aim at classifying phishing and legitimate messages on behalf of the user in an attempt to bridge the gap that is left due to the human error or ignorance. This is an important gap to bridge as user-training is more expensive than automated software classifiers, and usertraining may not be feasible in some scenarios (such as when the user base is huge, e.g. PayPal, eBay, etc. . .).

Further discussions on software classification approaches are presented in Sections VII, VIII, IX and X. The performance of detection approaches can be enhanced during the learning phase of a classifier (whether the classifier is human or software). In the case of end-users, their classification ability can be enhanced by improving their knowledge of phishing attacks by learning individually through their online experience, or by external training programs. In the case of software classifiers, this can be achieved during the learning phase of a Machine Learning-based classifier, or the enhancement of detection rules in a rule-based system. Detection techniques not only help in directly protecting end-users from falling victims to phishing campaigns, but can also help in enhancing phishing honeypots5 to isolate phishing spam from non-phishing spam. It is also important to note that the detection of phishing attacks is the starting point of the mitigation of phishing attacks. As depicted in Figure 3, if a phishing campaign is not detected, none of the other mitigation approaches can be applicable. For example, all of the mitigation techniques, such as correction, prevention and offensive defense depend on a functional and accurate detection phase. The primary focus of this survey is covering the detection phase of phishing attacks. Figure 4 depicts an overview of phishing detection techniques that are covered in the subsequent sections of this survey.



B. Offensive Defense Approaches

Offensive defense solutions aim to render phishing campaigns useless for the attackers by disrupting the phishing campaigns. This is often achieved by flooding phishing websites with fake credentials so that the attacker would have a difficult time to find the real credentials. Two notable examples are:

- BogusBiter [17] — A browser toolbar that submits fake information in HTML forms whenever a phishing website is encountered. According to BogusBiter, the detection of phishing websites is done by other tools. In other words, instead of simply showing a warning message to the enduser whenever a phishing website is visited, BogusBiter also submits fake data into HTML forms of the visited phishing website. Submitting fake data into the HTML forms is intended to disrupt the corresponding phishing campaigns, with the hope that such fake data may make the attackers task of finding correct data (among the fake data) more difficult. This is an attempt to save the stolen credentials of other users that have been captured by the phishing campaign by contaminating the captured results with bogus data. However, the limitations are:

– Toolbars need to be installed on a wide enough user base to render this effective.

– If the user base is wide enough, BogusBiter may cause Denial of Service (DOS) floods against servers that host legitimate shared hosted websites as well, simply because one of the shared web-hosts may have a phishing content.

– Increased bandwidth demand.

– Non-standard HTML forms are not detected by BogusBiter.

– The empirical effectiveness of this solution is not accurately measured.

- Humboldt [18] — Similar to BogusBiter, except that BogusBiter relies on submissions from end-user clients, while Humboldt relies on distributed and dedicated clients over the Internet instead of end-user toolbars that may visit phishing sites, in addition to a mechanism to avoid causing DOS floods against servers. This can make Humboldt more effective against phishing websites due to the more frequent submission of data to phishing pages.

The limitations are:

– Increased bandwidth demand.

– Non-standard HTML forms are not detected by Humboldt.

– The empirical effectiveness of this solution is not accurately measured. Although offensive defense approaches can theoretically make the attackers task more difficult in finding a victim's personal information, it is not known how difficult it really becomes. For example, a phisher might simply set up a script to test the credentials in a loop, and by using anonymous web surfing techniques attackers sessions will be difficult to track by the target web server. In other words, the actual returned security value of offensive defense approaches are not accurately evaluated and can be questioned.

C. Correction Approaches

Once a phishing campaign is detected, the correction process can begin. In the case of phishing attacks, correction is the act of taking the phishing resources down. This is often achieved by reporting attacks to Service Providers. Phishing campaigns often rely on resources, such as:

- Websites — could be a shared web host owned by the phisher, a legitimate website with phishing content uploaded to it, or a number of infected end-user workstations in a botnet.

- E-mail messages — could be sent from a variety of sources, such as: free E-mail Service Provider (ESP) (e.g. Gmail, Hotmail, etc. . .), open Simple Mail Transfer Protocol (SMTP) relays or infected end-user machines that are part of a botnet.

- Social Networking services

— web 2.0 services, such as Facebook and Twitter, can be used to deliver socially engineered messages to persuade victims to reveal their passwords.

- Public Switched Telephone Network (PSTN) and Voice over IP (VoIP) — similar to other forms of phishing attacks, attackers attempt to persuade victims to perform actions. However, the difference is that attackers attempt to exploit spoken dialogues in order to collect data (as opposed to clicking on links). Moreover, due to the way VoIP protocols (e.g. Session Initiation Protocol (SIP)) function, and the way many VoIP provider systems are configured, spoofing Caller IDs are used by attackers as tools to increase their persuasion [19]. In order to correct such behavior, responsible parties (e.g. service providers) attempt to take the resources down. For example: • Removal of phishing content from websites, or suspension of hosting services. • Suspension of email accounts, SMTP relays, VoIP services • Trace back and shutdown of botnets. This also extends to the shutdown of firms that frequently provide services to phishing attackers. The shutdown process can be initiated by organizations that provide brand protection services to their clients, which may include banking and financial companies that are possible victims of phishing attacks. When phishing campaigns are identified, they can be reported to their hosting Internet and web hosting service providers for immediate shutdown. Depending on the country where phishers and phishing campaigns exist, the penalties and procedures can differ.

D. Prevention Approaches

The “prevention” of phishing attacks can be confusing, as it can mean different things depending on its context: • Prevention of users from falling victim — in this case, phishing detection techniques will also be considered prevention techniques. However, this is not the context we refer to when “prevention” is mentioned in this survey. • Prevention of attackers from starting phishing campaigns — in this case, law suits and penalties against attackers by Law Enforcement Agencies (LEAs) are considered as prevention techniques. In this survey, whenever the keyword “prevention” is mentioned, it refers to the second previous item which is minimizing the possibility of attackers starting phishing campaigns via LEA. Usually, LEA may take a number of weeks to complete their investigation and response procedures. Thus, it is common to apply prevention techniques after all other mitigation techniques, which is due to the expensive nature of LEA investigations that makes them consume a relatively large period of time. Once the sources of the phishing attacks are traced, LEA can then file law suits which in turn may issue penalties such as: imprisonment, fines and forfeiture of equipment used to convey the attacks.

V. DETECTION OF PHISHING ATTACKS: THE HUMAN FACTOR

Since phishing attacks attempt to take advantage of the inexperienced users, an obvious solution is educating the users, which would in turn reduce their susceptibility to falling victims of phishing attacks. A number of user training approaches have been proposed throughout the past years. The human factor is broad. Simply educating end-users alone does not necessarily regulate their behavior [13]. This section will present and discuss some of the work contributed in the field of user training in relation to phishing attacks.

A. Phishing Victims

Julie S. Downs et al. [20] surveyed 232 computer users to study what are the criteria that can predict the susceptibility of a user to fall victims for phishing emails. The survey was formed in a role play where each user was expected to analyze emails as well as answering a number of questions. The outcome of the study was that those who had a good knowledge about the definition of "phishing" were significantly less likely to fall for phishing emails, while knowledge about other areas, such as cookies, spyware and viruses did not help in reducing vulnerability to phishing emails. Interestingly, the survey showed that knowledge about negative consequences (e.g. credit card theft) did not help in reducing vulnerability to phishing emails. The study concluded that user educational messages should focus on educating users about phishing attacks rather than warning them about the dangers of negative consequences.

Another study that confirms the study in [20] was made by Huajun Huang et. al. [21], which concluded that the primary reasons that lead technology users to fall as victims for phishing attacks are: • Users ignore passive warnings (e.g. toolbar indicators).

• A large number of users cannot differentiate between phishing and legitimate sites, even if they are told that their ability is being tested. A demographic study made by Steve Shen et. al. [1] shows a number of indirect characteristics that correlate between victims and their susceptibility to phishing attacks. According to their study, gender and age strongly correlate with phishing susceptibility. They conclude that:

• Females tend to click on email links more often than males.

• People between 18 and 25 years old were much more likely to fall victim to phishing attacks than other age groups. This was justified to be caused by a lack of sufficient technical knowledge and experience, which further confirms [20] and [21]. B. User-Phishing Interaction Model Xun Don et. al [5] described the first visual user-phishing interaction model. The model describes user interaction from the decision making point of view; which starts the moment a user sees phishing content, and ends when all user actions are completed (see Figure 5). The goal is assisting the process of mitigating phishing attacks by firstly understanding how users interact with phishing content. Inputs to the decision making process are:

- External information: could be anything learned through the User Interface (UI) (Web/mail client and their content), or expert advice. The phisher only has control over what is presented by the UI. Usually, the user does not ask for expert advice unless he is in doubt (i.e. if a user is convinced that a phishing site is legitimate, he might not ask for expert advice in the first place).
- Knowledge and context: the user's current understanding of the world, which is built over time (e.g. news, past experience).

- Expectation: users have expectations based on their understanding and the outcome of their actions. During the decision making process, two types of decisions can be made, which are:

- Planning a series of actions to be taken.

- Deciding on the next action in sequence to be taken. This is influenced by the outcome resulting from the previous action. The first action is often done consciously, while the subsequent actions are done sub-consciously. As a result, the outcome of the first action affects user's ability in detecting phishing attacks in subsequent actions. A phishing example that takes advantage of the above behavior is, an email that presents a non-existent Uniform Resource Locator (URL) pointing to a legitimate website, followed by a backup URL pointing to a phishing site. The first action a victim could take is clicking on the primary legitimate URL to view (say) an e-card (as claimed by the phisher), which obviously would result in a 404 page not found error. The second action would be clicking the backup link pointing to a phishing site. Since the phishing site was visited as a second action by the victim user, he would have a lower probability to detect inconsistencies in the backup URL. Each of the two types of decisions mentioned above, follow the following steps:

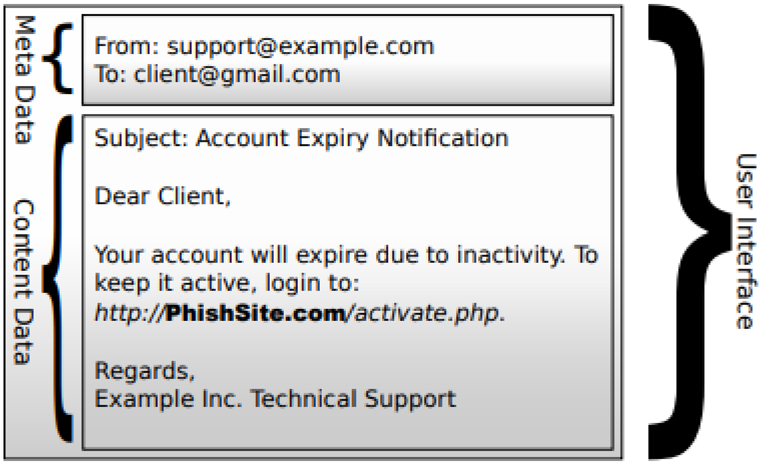
- Construction of perception: constructed through the context where the user reads (say) an email message. Such as, senders/recipients, conversation cause, or suggested actions by the email. In legitimate messages, there are no inconsistencies between the reality and message claims (e.g. senders are the real senders whom they claim to be, and suggested actions by email content does what it says).

However, in phishing messages there are inconsistencies (e.g. if the sender's ID is spoofed, or the message's content claims to fix a problem while attempting, in reality, to obtain personal information). If the end-user discovers inconsistencies in a given phishing message, the phishing attack would then fail to persuade the enduser.

- Generation of possible solutions: users usually find solutions through available resources. However, with phishing emails, the user is not requested to generate a possible solution in the first place, as the phisher already suggests a solution to the user. For example, if the phishing email content presents a problem, such as account expiry, it will also present a solution, such as activating the account through logging in a URL from which expiry is prevented.

- Generation of assessment criteria: different users have different criteria that reflects how they view the world, their emotional state, personal preferences, etc. . . . As the paper claims, most phishing attempts do not take into account such details, but rely on generic common-sense criteria instead; for example: an attacker might place a tick box labeled “Secure login” to meet a security criteria most users require. Phishing attacks aim to match user criteria as much as possible. As stated earlier, phishers can only modify the decision process of users through providing external information through the UI. The user interface provides two data sets (see Figure 6):

- Meta data: such as URL presented in web browser address bars, or email addresses. • Content data: such as site or email content. Phishing attacks succeed if a phishing attack convinces the user that both meta data and content data are legitimate. Users may use meta data to decide whether an email message is legitimate. Phishers may also spoof meta data in order to further trick the users. As stated in [5], the solution to meta-data integrity problem is not through user education or awareness as it is very difficult for users to validate whether the source IP address is legitimate in case the domain name was spoofed. Users should not be expected to validate the meta-data as it is rather a system design or implementation problem. On the other hand, through social engineering, phishers create convincing and legitimate-looking content data. A common solution to this is user awareness. C. Service Policies Although different methods have been developed to deliver warnings and notifications to end-users, them having a knowledge of phishing is required to render warnings or notifications useful. Users should be made aware of various types of social engineering attacks, and a low level of awareness would result in exposing user’s credentials irrespective of software or hardware protection layers. A relatively common solution is educating users via periodic messages (Emails, SMS, etc. . .). Organizations should have strict policies against the distribution of confidential data over email, Short Message Service (SMS), or VoIP, coupled with user awareness programs (e.g. periodic educational messages) to ensure that users are aware of the policies. Users that are made aware of this have a higher



From: support@example.com
To: client@gmail.com

Subject: Account Expiry Notification

Dear Client,

Your account will expire due to inactivity. To keep it active, login to:
*http://*__PhishSite.com__*/activate.php.*

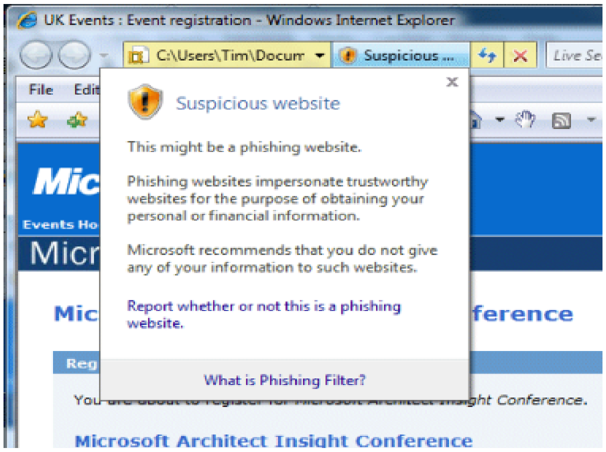
Regards,
Example Inc. Technical Support

chance of detecting inconsistencies within a phishing message if the content data, for example, asked for confidential information. Service providers should also strictly enforce their policies against illicit use of their services. Many hosting providers take services down in cases where they are abused. As researched by T. Moore, and R. Clayton in [22], service take-down is increasingly common in the way security problems are handled. Users should also be aware of the environment that they interact with. e.g. mail and web clients display meta data, which users need to use to validate content data; e.g. Thunderbird displays notifications if a mail signature is not valid. Ignoring warnings, such as X.509 certificate verification failure messages, could lead users into permanently trusting illicit certificates. If the trusted certificate was caused by a Man in the Middle (MITM) attack, then such behavior could break the trust model of Public Key Infrastructure (PKI), making Hypertext Transfer Protocol Secure (HTTPS) or Secure/Multipurpose Internet Mail Extensions (S/MIME) the same as their insecure counterparts. User education should be coupled with clear IT policies as well as applied practices. Otherwise, the policy could reverse the educational message. For example, an IT policy that warns about the dangers of HTTPS sites with invalid X.509 certificates, while publishing local sites with self-signed certificates (such as the default setup of MS Outlook Web Access), could lead into an implied message that the security policies and warnings are not of great importance. Ultimately, this leads to the spread of habits opposing the actual intentions of security policies. Another challenge is the widespread use of computers in environments other than the workplace, where proper IT policies do not exist. Frequent misuse of technology could lead end users into considering ignoring security warnings as the norm, which might gradually be taken into their workplace as well. The psychological aspect of the phishing problem is broad, and plays a major role on the end user behavior. Due to the generic nature of the challenge, a highly predictable solution remains difficult to achieve, if not impossible by today’s scientific advancements in understanding human mental functions and behavior.

D. Passive and Active Warnings

User interfaces can show security warnings based on triggered actions, such as viewing phishing web pages, as commonly deployed by many web browsers. There are generally two ways of presenting the warnings to the end user:

- Passive warnings — the warning does not block the content-area and enables the user to view both the content and the warning as in the snapshot .



Suspicious website

This might be a phishing website.

Phishing websites impersonate trustworthy websites for the purpose of obtaining your personal or financial information.

Microsoft recommends that you do not give any of your information to such websites.

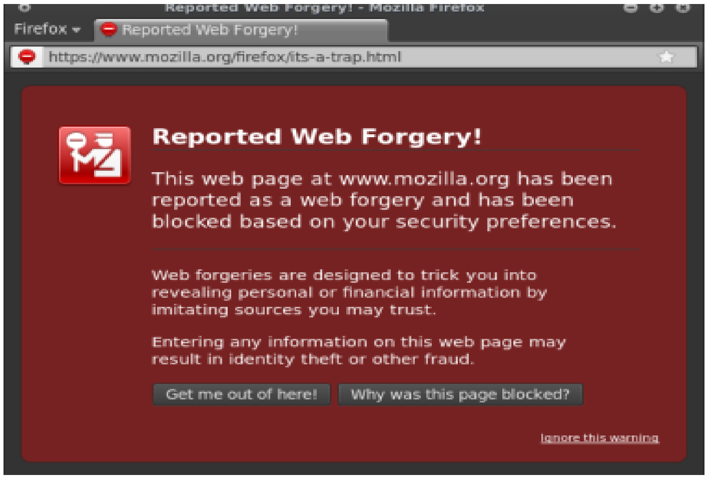
Report whether or not this is a phishing website.

What is Phishing Filter?

You ght Conference.

Microsoft Architect Insight Conference

- Active warnings — the warning blocks the content-data, which prohibits the user from viewing the content-data while the warning is displayed as in the snapshot.


Reported Web Forgery! - Mozilla Firefox
Firefox
Reported Web Forgery!
https://www.mozilla.org/firefox/its-a-trap.html

Reported Web Forgery!

This web page at www.mozilla.org has been reported as a web forgery and has been blocked based on your security preferences.

Web forgeries are designed to trick you into revealing personal or financial information by imitating sources you may trust.

Entering any information on this web page may result in identity theft or other fraud.

Get me out of here!
Why was this page blocked?
Ignore this warning

According to a study conducted by Egelman et. al. [23], passive warnings are ineffective. Only 13% of the participants heeded to passive browser warnings, while 79% of the participants heeded for active warnings. Similarly, toolbars have mostly followed passive warnings expressed by notification icons, which justifies the failure of security toolbars to properly notify end-users of phishing risks. According to a study conducted by Min Wu et. al [24], users failed to heed toolbar passive warnings. Both of the studies, [23] and [24], concluded that active warnings are superior to passive ones. Users do not heed warnings unless they are forced to by blocking the contentdata portion of the UI to display a clear warning message.

E. Educational Notices E-services, such as e-banking, often communicate periodic educational messages to warn their clients of potential phishing threats. The messages are often delivered via SMS's and 10 e-mails. According to a study conducted by Kumaraguru et. al. [11], periodic security notices are ineffective and, although they may improve the knowledge of end-users, the periodic notices fail to change their behavior. Alternatively, Kumaraguru et. al. [11] proposes and evaluates the design of an alternative periodic method to send educational notices, that are embedded into the daily activity of the end-users. The study shows that embedded training systems are more effective than periodic security notices (89% of participants who were trained by the periodic notices were victims for testing phishing emails, while 30% of participants who were trained by comics embedded training messages were victims of testing phishing emails).

The evaluation in [11] also compared multiple approaches in educating users, such as text, annotated figure and comics. The study concluded that embedded comics were the most effective approach. The proposed system functions as follows:

- The e-mail administrator prepares a number of fake phishing emails.
- The phishing emails are communicated to the victims. No warnings are shown at this stage.
- Once the victim interacts with a phishing email, such as by clicking on a phishing link, the user is then shown a security warning teaching him the risks of phishing attacks. The advantage of this embedded training method is that it teaches the end-user in the most receptive moment — which is when he falls as a victim for a fake phishing attack. Kumaraguru et. al. [11] then concluded with a number of design principles that should be followed to enhance user educational notices:

- Training messages to be embedded into the daily activity of the user, without the need to read external sources (e.g. other websites or SMS).
- The warning message should clearly and concisely explain the causes; warning messages can fail if they have too much of textual data.
- The warning message should clearly and concisely explain proper actions to be taken by the end-user to enhance his/her security.
- The warning should not be delayed, but be shown immediately following the moment when the user falls as victim and clicks on an email link.
- The fake phishing messages used for training purposes should mimic closely phishing messages in the wild.
- Enhancing the security warning text with story-based comics enhances readability. However, the drawbacks of the proposed embedded training system are:
- The system requires a human administrator to craft the messages. This adds delay and increases the maintenance cost of the solution.
- The crafted phishing emails by the administrator are limited by the administrator’s understanding of phishing attacks. If the administrator is unaware of latest trends in phishing emails, his/her crafted phishing attacks might be less effective. In other words, a highly

important parameter to the proposed method is the administrator who is tasked to create phishinglike messages, and the proposed solution’s performance is heavily based on the performance of the administrator whom in turn is a variable as not all administrators are equivalent.

VI. PHISHING DETECTION BY BLACKLISTS

Blacklists are frequently updated lists of previously detected phishing URLs, Internet Protocol (IP) addresses or keywords. Whitelists, on the other hand, are the opposite, and could be used to reduce F P rates. Blacklists do not provide protection against zero-hour phishing attacks as a site needs to be previously detected first in order to be blacklisted. However, blacklists generally have lower F P rates than heuristics [25].

As studied in [25], blacklists are found to be ineffective against zero-hour phishing attacks, and were able to detect only 20% of them. The study [25] also shows that 47% to 83% of phishing URL were blacklisted after 12 hours. This delay is a significant issue as 63% of phishing campaigns end within the first 2 hours.

A. Google Safe Browsing API

Google Safe Browsing API enables client applications to validate whether a given URL exists in blacklists that are constantly updated by Google [26]. Although the protocol is still experimental, it is used by Google Chrome and Mozilla Firefox.

The current implementation of the protocol is provided by Google, and only consists of two blacklists named goog-phishshavar and goog-malware-shavar, for phishing and malware respectively. However the protocol itself is agnostic to the list type as well as to the provider of the list. The API requires client applications to communicate with providers through HTTP while adhering to syntax specified in Protocolv2Spec [27], which is the second version of the protocol; the first version faced scalability and efficiency issues that are also outlined in [27]. Notable changes in the second version are:

- Version 2 of the protocol divides the URL data into chunks and allows partial updates. Such partial updates were not available in version 1 of the protocol.

- Version 2 of the protocol does not always send full 256- bit hashes of blacklisted URLs to web browsers, instead it initially sends a list of 32-bit truncated forms of the hashes. However, if a visited URL had a hash such that its 32-bit truncated form matched any of those in the truncated URLs hash list, then the browser will download all complete 256-bit hashes that have the same first 32- bits as that of the suspect URL. This allows saving bandwidth when transmitting the URLs hash blacklist, specially when knowing that most visited URLs are legitimate in the common case (i.e. the no-match:match ratio is high in reality), and that the first 32-bits are still sufficient (for most cases) to yield mismatches between different URLs. In other words, the 256-bit URL hashes are only sent to resolve possible hash collisions with 11 regards to the first 32-bits of the hashes (i.e. to avoid false positives due to hash collisions).

Version 1 of the protocol did not support such use of truncated hashes, and always sent the full 256-bits (which resulted in inefficient use of bandwidth). More details are presented in the background section in Protocolv2Spec [27].

The client application downloads lists from providers, and maintains its freshness by incremental updates that contain additive and subtractive chunks. The protocol follows a pull model that requires the client to connect to the server at certain integer time intervals measured in minutes (tm). To reduce the server load as the number of clients increases, the following mechanism is followed by the client applications:

1) The first update occurs after 0 to 5 minutes following client start. tm ← rand(0, 5).

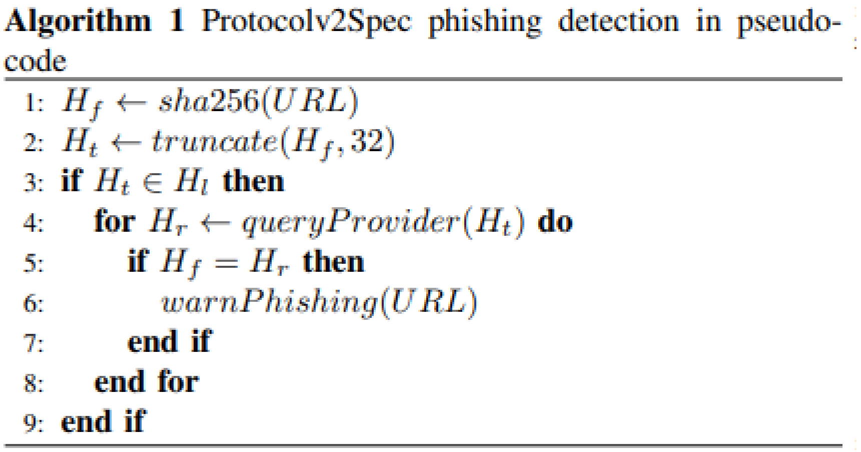
2) tm would then be set explicitly by the server for subsequent updates. Otherwise, the client would assume tm ← rand(15, 45). In case of error or timeout:

1) tm ← 1. 2)

If an error or timeout was returned, then tm ← 30(rand(0, 1) + 1).

3) If an error or timeout was also returned, then tm ← 2tm.

4) Step 3 is repeated until tm = 480. The client then fixes the interval at 480 minutes until the server responds. Since Version 2 of the protocol distributes the first 32-bit of every SHA-256 hash, the client has to apply Algorithm 1 in order to handle possible hash collisions with regards to the first 32-bit segments of the hashes.



Algorithm 1 Protocolv2Spec phishing detection in pseudo-code

```
$H_f \leftarrow sha256(URL)$
$H_t \leftarrow truncate(H_f, 32)$
if $H_t \in H_l$ then
    for $H_r \leftarrow queryProvider(H_t)$ do
        if $H_f = H_r$ then
            $warnPhishing(URL)$
        end if
    end for
end if
```

B. DNS-Based Blacklist

DNS-based Blacklist (DNSBL) providers use the standard DNS protocol. Due to its use of the standard DNS specification, any standard compliant DNS server could act as a DNSBL. However, since the number of listed entries is large, a server that is not optimized for handling large amounts of DNS A or TXT Resource Records (RRs) may face performance and resource strains. rbldnsd7 is a fast DNS server, designed specifically to handle large RR suited for DNSBL duties.

When a Message Transfer Agent (MTA) establishes an inbound SMTP connection, it can verify whether the connecting source is listed in phishing blacklists8 , by sending a DNS A RR query to a DNSBL server on the connecting IP address. For example, if the source IP address of the SMTP connection is 86.96.226.20, and assuming that the DNSBL server is dnsbl2.uceprotect.net, then a DNS A RR query should be sent for 20.226.96.86.dnsbl-2.uceprotect.net to a DNS server; which is the reverse form of the suspect IP address appended by the DNSBL domain name.

If the DNSBL server found an entry, then it would mean that the IP address is blacklisted. Listing 1 presents the output of a DNS A RR query to a DNSBL server with the program host executed in a Unix machine. The output indicates that there exists a DNS A RR entry, which means that the respective IP address (i.e. 86.96.226.20) is blacklisted.

C. PhishNet:

Predictive Blacklisting Any changes to a Phishing URL would result in no match. PhishNet [28] addresses the exact match limitation found in blacklists. To solve this, PhishNet processes blacklisted URLs (parents) and produces multiple variations of the same URL (children) via 5 different URL variation heuristics, which are listed below:

• Replace Top Level Domains (TLD): Each URL will fork into 3,210 variations, each with a different TLD.

• Directory structure similarity: If multiple Phishing URLs have similar directory structures with minor variations, multiple children URLs will be created to assemble differences across all the attack URLs that have similar directory structures. For example:

– http://www.abc.com/online/ebay.html.

– and http://www.xyz.com/online.paypal.com. would result into forking the following children

URLs: – http://www.xyz.com/online/ebay.html. – and http://www.abc.com/online.paypal.com.

• IP address equivalence: URLs with similar directory structure but different domain names are considered as a match if they point to the same IP address.

• Query string substitution: Similar to “directory structure similarity” except that it forks multiple variations of a URL with different query strings. For example:

– http://www.abc.com/online/ebay.php?ABC. – and http://www.abc.com/online.paypal.com?XYZ. would result into forking the following children URLs: – http://www.abc.com/online/ebay.php?XYZ. – and http://www.abc.com/online.paypal.com?ABC.

• DNS query: does the domain name exist?

• TCP connect: is the resolved name running a HTTP server? • HTTP header response: does the page exist? (HTTP 200/202 == found).

D. Automated Individual White-List

Automated Individual White-List (AIWL) [29] maintains a whitelist of features describing trusted Login User Interfaces (LUIs) where the user submitted his/her credentials. Every LUI will cause a warning except if trusted. Once a LUI is trusted, its features will be stored locally in a whitelist. There are two primary components of AIWL, namely: • Whitelist — a list of trusted LUIs. Its objective is suppressing warnings with regards to LUIs that are trusted. In order to detect whether a suspect LUI is trusted, the LUI is represented as a feature vector and then compared against those feature vectors in the whitelist. If the LUI features of a suspect page do not match all of the features in any whitelisted LUI, the suspect page is assumed to be untrusted and warnings are presented to the end-user. Such features that uniquely identify LUIs are: – URL address of the trusted LUI. – IP addresses that correspond to the trusted LUI. Since a DNS A Resource Record can be mapped to more than one IP address (e.g. for reasons related to load balancing), multiple IP addresses can be mapped to a single LUI. – The hash of the X.509 certificate that is associated with the trusted LUI. – The Document Object Model (DOM) path of username and password input fields of the trusted LUI. For example, if the input field username exists in a form named loginform, which is in turn in a frame named mainframe, then it will be represented as mainframe/loginform/username. In other words, username and password input fields are identified as a directory structure. The objective of this structure is detecting phishing attacks that use iframe tags to present login forms of legitimate websites. For example, an attacker could construct a phishing page that presents another page (e.g. PayPal) via the iframe HTML tag, and then monitors user's interaction with the page via JavaScript. Identifying username and password fields in a directory structure based on DOM helps in detecting differences between input fields that are presented normally in their original website, and those that are presented in iframe HTML tags. • Automated whitelist maintainer — a classifier that decides whether a suspect LUI to be installed in the whitelist. The merit of the whitelist maintainer is that if an end-user logs in successfully for enough amount of times via a given LUI, then that LUI is trusted (and thus whitelisted). This means that the automated whitelist maintainer requires a mechanism from which it is able to decide whether a login was successful.

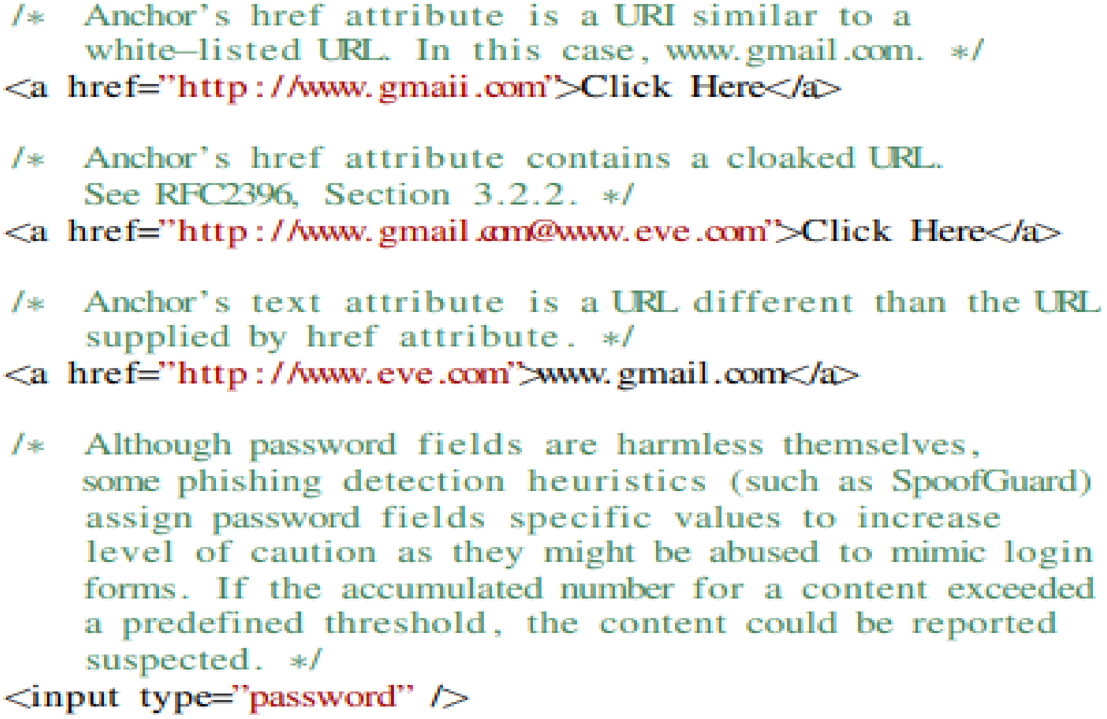
VII. PHISHING DETECTION BY HEURISTICS

Software could be installed on the client or server side to inspect payloads of various protocols via different algorithms. Protocols could be HTTP, SMTP or any arbitrary protocol. Algorithms could be any mechanism to detect or prevent phishing attacks. Phishing heuristics are characteristics that are found to exist in phishing attacks in reality, however the characteristics are not guaranteed to always exist in such attacks. If a set of general heuristic tests are identified, it can be possible to detect zero-hour phishing attacks (i.e. attacks that were not seen previously), which is an advantage against blacklists (since blacklists require exact matches, the exact attacks need to be observed first in order to blacklist them). However, such generalized heuristics also run the risk of misclassifying legitimate content (e.g. legitimate emails or websites).

Currently, major web browsers and mail clients are built with phishing protection mechanisms, such as heuristic tests that aim at detecting phishing attacks. The clients include Mozilla FireFox, Internet Explorer, Mozilla Thunderbird and MS Outlook. Also, phishing detection heuristics could be included in Anti Viruses, similar to ClamAV.

A. SpoofGuard SpoofGuard [30], a web browser plug-in developed by Stanford University, detects HTTP(S)-based phishing attempts as a web browser toolbar, by weighting certain anomalies found in the HTML content against a defined threshold value. Listing 4 shows samples of HTML heuristically detectable by SpoofGuard. Listing 4. Sample of heuristic phishing detection based on content.

Listing 4. Sample of heuristic phishing detection based on content.



```
/*  Anchor's href attribute is a URI similar to a
    white–listed URL. In this case, www.gmail.com. */
<a href="http://www.gmaii.com">Click Here</a>

/*  Anchor's href attribute contains a cloaked URL.
    See RFC2396, Section 3.2.2. */
<a href="http://www.gmail.com@www.eve.com">Click Here</a>

/*  Anchor's text attribute is a URL different than the URL
    supplied by href attribute. */
<a href="http://www.eve.com">www.gmail.com</a>

/*  Although password fields are harmless themselves,
    some phishing detection heuristics (such as SpoofGuard)
    assign password fields specific values to increase
    level of caution as they might be abused to mimic login
    forms. If the accumulated number for a content exceeded
    a predefined threshold, the content could be reported
    suspected. */
<input type="password" />
```

B. Collaborative Intrusion Detection Many phishing detection and prevention mechanisms are based on finding the source IP address of the attacker. Fast-flux [31], on the other hand, enables attackers to frequently change their IP addresses. By having a sufficient number of infected hosts (usually home users), hosts can behave as front-end proxies for phishing websites. Multiple front-end proxies relay the traffic back to a main phishing site to fetch the content from (also known as mothership).

Load balancing is achieved by means of low Time to live (TTL) DNS A RR, which enables a quicker change of mapped IP addresses than if higher TTL values were used. Low TTL also helps in reducing the downtime when dead front-end proxies are removed. The DNS A RR are updated by the attacker through a fixed DNS NS RR. Although fast-flux has a fixed NS record pointing to the attacker's real IP address, double fast-flux complicates the task further by relaying DNS RR update to front-ends as well. A proposed solution to this is through the use of Collaborative Intrusion Detection System (CIDS) to exchange phishingrelated data among a number of Intrusion Detection Systems (IDSs).

The distributed system should be installed globally. Each local CIDS should monitor its local DNS cache, and list DNS zones with a high number of DNS A RR that are combined with low TTL values. The list of IP addresses and domains are sent to global CIDS for further analysis. Each recipient CIDS would in turn have to monitor connections of reported IP addresses (which are infected home users in most cases). By monitoring the inbound and outbound connections of suspected source IP, it should be possible to find the mothership that is distributing the original phishing content to other front-end proxies. The proposed mechanism counts the number of connections to the front-end proxies (infected hosts), and distributes such numbers to global CIDS. If the threshold reaches a certain number, it would then be assumed that it is a connection to the mothership (since the mothership has one-to-many connections, it is assumed to have a high number of accumulated connections globally).

This proposed solution has not been implemented yet due to difficulty in studying fast-flux or double fast-flux attacks due to their unrepeated nature so far. It also faces challenges in determining which connection is toward the mothership. Connection count is not a definitive criteria since many phishing networks could also be connected to other legitimate networks (such as Internet Relay Chat (IRC) or game networks), from which they initially appeared. Thus, doing a simple connection count could lead into false positives.

C. PhishGuard:

A Browser Plug-in The work in [32] bases its protection against phishing on the idea that phishing websites do not often verify user credentials, but merely store them for later use by the phisher. The authors in [32] acknowledge that, in the future, phishing sites would be more sophisticated and pipe (or tunnel) their output from legitimate sites, acting as a man in the middle attack, which would in turn result in proper success or failure login warnings (as communication is simply piped back and forth).

However, the paper states that such use is not common yet, and mainly focuses its detection on non-piped phishing attempts. PhishGuard's implementation in [32] is a proof of concept that only detects phishing attacks based on testing HTTP Digest authentications. However, integrating other authentication mechanisms, such as through HTML form submission, is possible. PhishGuard follows the following steps to test a suspected page.

1) The user visits a page.

2) If the visited page sends an authentication request, and if the user submitted the authentication form, then PhishGuard starts its testing procedures.

3) PhishGuard would send the same user ID, followed by a random password that does not match the real password, for random n times.

4) If the page responded with HTTP 200 OK message, then it would mean the page is a phishing site, and is simply returning fake authentication success messages.

5) If the page responded with HTTP 401 Unauthorized message, then it could possibly mean:

• The site is a phishing site that blindly responds with failure authentication messages.

• The site is a legitimate site.

6) To distinguish between the two possibilities above, PhishGuard would send real credentials to the website for the n + 1 time.

7) If the page responded with a HTTP 200 OK message following the login request when the real credentials were used, then the site passes the test and is deemed legitimate.

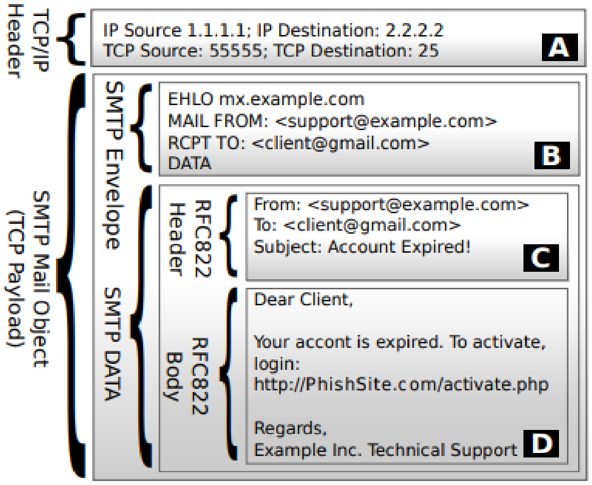
8) If the page responded with a HTTP 401 Unauthorized message, then it could possibly mean:

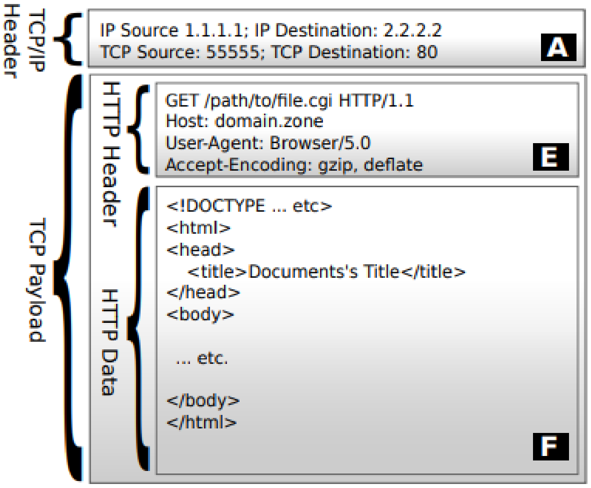
• The web site is a phishing site that blindly responds with failure authentication messages. In this case the login credentials of the user were submitted to the phisher. Clearly, the technique is only able to prevent password theft for a subset of phishing websites.

• The user submitted the wrong password. 9) To ensure that the submitted password is not a wrong password mistyped by the user, PhishGuard stores password hashes in a file and verifies future login requests against it:

• If the hash of the entered password matches any entry in the file, then PhishGuard concludes that the password was correct, and the site was a phishing website.

• If no match was found, then PhishGuard would conclude the supplied password is wrong, and the user is then alerted to correct his/her password.


TCP/IP Header
IP Source 1.1.1.1; IP Destination: 2.2.2.2
TCP Source: 55555; TCP Destination: 25
A
SMTP Mail Object (TCP Payload)
SMTP Envelope
EHLO mx.example.com
MAIL FROM: <support@example.com>
RCPT TO: <client@gmail.com>
DATA
B
SMTP DATA
RFC822 Header
From: <support@example.com>
To: <client@gmail.com>
Subject: Account Expired!
C
RFC822 Body
Dear Client,
Your accont is expired. To activate, login:
http://PhishSite.com/activate.php
Regards,
Example Inc. Technical Support
D


TCP/IP Header
IP Source 1.1.1.1; IP Destination: 2.2.2.2
TCP Source: 55555; TCP Destination: 80
A
TCP Payload
HTTP Header
GET /path/to/file.cgi HTTP/1.1
Host: domain.zone
User-Agent: Browser/5.0
Accept-Encoding: gzip, deflate
E
HTTP Data
<!DOCTYPE ... etc>
<html>
<head>
<title>Documents's Title</title>
</head>
<body>
... etc.
</body>
</html>
F

VIII. EVALUATIONS OF SOFTWARE PHISHING DETECTION TECHNIQUES

Analyzed data parts by phishing detection techniques are depicted in Figures 9 and 10, which are used in Table III to compare the detection techniques as they are evaluated in the literature. This implies the use of different data sets, and that the results are not directly comparable. However, since the evaluation samples are taken from the same population (i.e. phishing and legitimate websites and emails in the Internet), the differences should not be significant.

The rates F P and F N are measured as presented in Section V. Table III presents generic approaches (e.g. blacklists, whitelists, heuristics, Machine Learning) used by the phishing detection, and their detection rates in terms of F P and F N. The table also indicates whether the presented techniques require access to resources over the Internet in order to function. It can be observed that the majority of the techniques do not rely on the resources over the Internet in order to perform classification decisions (six out of the twenty surveyed techniques require access to the Internet for performing such classification decisions). We believe that this is reflected due to the fact that accessing resources over the Internet can be expensive, and could form a potential bottleneck.

It should be also noted that among the techniques that require Internet access include CIDS (which was not implemented [52]; also reflects the associated difficulty with such distributed system over the Internet), and two techniques that are related to Google [4], [34] (which is a scenario where accessing search engine rankings is not necessarily over the Internet since the results are cached and analyzed as part of Google's internal infrastructure). Blacklists are able to achieve low F P rates. However, blacklists have been evaluated to be ineffective against zerohour attacks, as they only detect 20% of phishing attacks at hour zero. The best performing anti-phishing email classifiers used Machine Learning techniques, namely those by Andre Bergholz et. al. [46] and Fergus Toolan et. al. [47].

Contrary to heuristics, Machine Learning techniques were also able to achieve low F P rates. For example, Andre Berghol'z model-based email classifier and Google's large scale website classifiers achieved 0% F P rates. Fergus Toolan’s R-Boost achieved 1.3% F P rate, however it should be noted that the aim of the R-Boost technique is to minimize the F N rate only, which the technique reached by achieving an F N rate of 0%. Natural Language Processing (NLP) techniques are rarely found in the phishing mitigation literature, which — to the best our knowledge — to be due to lack of sufficient maturity in NLP techniques with regards to correctly understanding the semantics of messages written in natural languages that also may contain typos. Email and web browsing are critical tasks and software techniques still finds it extremely challenging to correctly understand the semantics of natural languages. Moreover, many email messages can have typos, which further complicates the job of NLP techniques.

An example of an NLP-based anti-phishing technique is EBDIS [15] with an F P rate of 1.9%, and an F N rate of 25% which is not as accurate as the competition. However, although NLP techniques — 25 alone — do not achieve competing F P and F N rates, their addition to other techniques (e.g. ML) can be promising. Existing visual similarity-based phishing detection techniques heavily rely on use of blacklists or whitelists of website snapshot salient point descriptors. Conceptually, they are still blacklists or whitelists and require frequent updates. However, blacklists of website snapshot salient points are able to address zero-hour attacks, while URL blacklists can not. On the other hand, visual similarity-based detection techniques assume that phishing websites are always similar to the websites of targeted brands (e.g. PayPal phishing websites look similar to PayPal’s legitimate website), which might not be always true. To assess their true effectiveness, it would be necessary to empirically measure the susceptibility of end-users to fall victims for phishing websites that are not very similar to their targeted brands. Until such demographic study is made, the effectiveness of visual similarity techniques might not be accurately known, but rather assumed.

IX. LEARNED LESSONS

A. User education and awareness Since phishing is a social engineering attack, an obvious solution can be educating the end-user. However, as discussed in [13] education and user awareness — alone — are not enough, and that what is needed is regulating the behavior of end-users instead. Various incidents, such as the one described in [2], have shown that even security providers themselves have fallen victims for phishing attacks.

Although phishing seems to be a simple attack that exploits the naivety of end-users, it is able to persuade security-aware engineers as well. This indicates the possibility that systems’ complexities are raising beyond the cognition limits of many humans, and that simply educating them is not enough. A more promising solution could be enhancing the system usability via:

- Better user interfaces. It is visible to us that the software industry is moving towards this direction. For example, older versions of web browsers used passive warnings, while recent ones moved to active warnings. A security warning should be active (i.e. blocks the content) and should visually hint the user of risks even without reading its content (since most end-users do not read warning messages [23]).

• Enhancing the behavior of the systems, so that the harmful messages are automatically detected and quarantined on behalf of the end-user. For example, blacklists, heuristic rules and ML techniques can be used to automatically filter harmful content from end-users. Such features are currently implemented in web browsers, email clients and server-side filters.

The ideal phishing mitigation direction seems to us to be debatable so far, which could be due to the short history depth of information technology in general. However, in our opinion, systems are in reality moving towards adapting to their end-users (as opposed to having end-users adapting to their systems).

B. Blacklists Blacklists are effective when minimal F P rates are required, which is achieved due to the way blacklists are constructed (e.g. many of them involve human administration, such as PhishTank).

Blacklists also have the advantage of requiring low resources on the host machine. This elevates the need of extensively analyzing the content of websites and emails. However, blacklists are known to be behind the line when the objective is mitigation of zero-hour phishing attacks. It is critical to mitigate against zero-hour phishing attacks since most of the phishing campaigns are short-lived [25].

The primary reason behind blacklists inability to mitigate zero-hour phishing campaigns is due to the following factors:

• The time spent to blacklist a resource (e.g. phishing URLs) — a phishing campaign’s URL or IP address should be detected first prior to its addition to the blacklist. Blacklist providers, such as PhishTank, rely on humans to vote on phishing URLs. According PhishTank’s statistics27, the median detection time is 2 hours. This is a significant delay as 63% of phishing campaigns end within the first 2 hours as well [25].

• The time spent to propagate an updated blacklist to enduser clients — For example, according to Google Safe Browsing API v2, the browser should synchronize its blacklist after 0–5 minutes (chosen randomly) after its startup, and then the browser should keep updating it periodically as specified by the blacklist server. This can leave a 5 minutes gap following the browser startup, and an arbitrary amount of time as specified by the blacklist server. This also includes delay caused by network and application service providers, specially with DNS-Based Blacklists (DNSBLs) as they send DNS queries for each suspect resource (e.g. IP address).

C. Heuristic tests and visual similarity

Unlike blacklists, heuristic tests are able to constantly detect phishing campaigns including zero-hour attacks [25]. However, they tend to have higher F P rates [25] than blacklists, which — in our opinion — is caused due to the following:

- The complicated nature of adversarial attacks makes it hard for humans to manually construct heuristic tests that mitigate the attacks without causing F P.

- Techniques used by phishing campaigns evolve with time, and thus heuristic tests require to be continuously updated. Although this periodic update is less frequent than blacklists', it is more expensive as it requires classifier designers to analyze the data and generalize tests that mitigate the phishing attacks.

- In order to take a snapshot of an analyzed website, its content should be rendered first (i.e. similar to Internet Explorer and FireFox). This requires parsing the HTML, Cascading Style Sheets (CSS), and JavaScript codes, as well as Flash and Java objects, which will add an arbitrary amount of processing time depending on the website's content.

- Storing information that describes images can be expensive. For example, storing snapshots (or descriptors of snapshots) of websites may require more space than simply storing their URL, which is the case with techniques that are presented in this survey.

D. Machine Learning-based classifiers

Similar to heuristic tests, ML-based techniques can mitigate zero-hour phishing attacks, which makes them advantageous when compared with blacklists. Interestingly, ML techniques are also capable of constructing their own classification models by analyzing large sets of data. This elevates the need of manually creating heuristic tests as ML algorithms are able to find their own models. In other words,

ML techniques have the following advantages over heuristic tests:

- Despite the complicated nature of adversarial attacks, it is possible to construct effective classification models when large data set samples are available, without the need of manually analyzing data to discover complex relationships.

- As phishing campaigns evolve, ML classifiers can automatically evolve via reinforcement learning. Alternatively, it is also possible to periodically construct newer classification models by simply retraining the learner with updated sample data sets.

X. CONCLUSION

User education or training is an attempt to increase the technical awareness level of users to reduce their susceptibility to phishing attacks. It is generally assumed that the addition of user education materials compliments technical solutions (e.g. classifiers). However, the human factor is broad and education alone may not guarantee a positive behavioral response.

• Detection accuracy with regards to zero-hour phishing attacks. This is due to the fact that phishing websites are mostly short-lived and detection at hour zero is critical.

• Low false positives. A system with high false positives might cause more harm than good. Moreover, end-users will get into the habit of ignoring security warnings if the classifier is often mistaken. Generally,

software detection solutions are:

• Blacklists.

• Rule-based heuristics.

• Visual similarity.

• Machine Learning-based classifiers.

The findings in Section IX show that the use of Machine Learning techniques is promising as they have led to the most effective phishing classifiers in the publicly known literature. The Machine Learning-based detection techniques achieved high classification accuracy for analyzing similar data parts to those of rule-based heuristic techniques.

REFERENCES

[1] S. Sheng, M. Holbrook, P. Kumaraguru, L. F. Cranor, and J. Downs, “Who falls for phish?: a demographic analysis of phishing susceptibility and effectiveness of interventions,” in Proceedings of the 28th international conference on Human factors in computing systems, ser. CHI ’10. New York, NY, USA: ACM, 2010, pp. 373–382.

[2] B. Krebs, “HBGary Federal hacked by Anonymous,” http: //krebsonsecurity.com/2011/02/hbgary-federal-hacked-by-anonymous/, 2011, accessed December 2011.

[3] B. Schneier, “Lockheed Martin hack linked to RSA’s SecurID breach,” http://www.schneier.com/blog/archives/2011/05/lockheed martin.html, 2011, accessed December 2011.

[4] C. Whittaker, B. Ryner, and M. Nazif, “Large-scale automatic classification of phishing pages,” in NDSS ’10, 2010.

[5] X. Dong, J. Clark, and J. Jacob, “Modelling user-phishing interaction,” in Human System Interactions, 2008 Conference on, may 2008, pp. 627 –632.

[6] W. D. Yu, S. Nargundkar, and N. Tiruthani, “A phishing vulnerability analysis of web based systems,” in Proceedings of the 13th IEEE Symposium on Computers and Communications (ISCC 2008). Marrakech, Morocco: IEEE, July 2008, pp. 326–331.

[7] Anti-Phishing Working Group (APWG), “Phishing activity trends report — second half 2010,” http://apwg.org/reports/apwg report h2 2010. pdf, 2010, accessed December 2011.

[8] Anti-Phishing Working Group (APWG), “Phishing activity trends report — first half 2011,” http://apwg.org/reports/apwg trends report h1 2019.pdf, 2019, accessed December 2019.

[9] Anti-Phishing Working Group (APWG), “Phishing activity trends report — second half 2011,” http://apwg.org/reports/apwg trends report h2 2011.pdf, 2011, accessed July 2012.

[10] B. Schneier, “Details of the RSA hack,” http://www.schneier.com/blog/archives/2011/08/details of the.html, 2011, accessed December 2011.

[11] P. Kumaraguru, Y. Rhee, A. Acquisti, L. F. Cranor, J. Hong, and E. Nunge, “Protecting people from phishing: the design and evaluation of an embedded training email system,” in Proceedings of the SIGCHI conference on Human factors in computing systems, ser. CHI ’07. New York, NY, USA: ACM, 2007, pp. 905–914.

[12] A. Alnajim and M. Munro, “An anti-phishing approach that uses training intervention for phishing websites detection,” in Proceedings of the 2009 Sixth International Conference on Information Technology: New Generations. Washington, DC, USA: IEEE Computer Society, 2009, pp. 405–410.

[13] S. Gorling, “The Myth of User Education,” Proceedings of the 16th Virus Bulletin International Conference, 2006.

[14] G. Gaffney, “The myth of the stupid user,” http://www.infodesign.com.au/articles/themythofthestupiduser, accessed March 2011.

[15] A. Stone, “Natural-language processing for intrusion detection,” Computer, vol. 40, no. 12, pp. 103 –105, dec. 2007.

[16] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, “A comparison of machine learning techniques for phishing detection,” in Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit, ser. eCrime ’07. New York, NY, USA: ACM, 2007, pp. 60–69.

[17] C. Yue and H. Wang, “Anti-phishing in offense and defense,” in Computer Security Applications Conference, 2008. ACSAC 2008. Annual, 8-12 2008, pp. 345 –354.

[18] P. Knickerbocker, D. Yu, and J. Li, “Humboldt: A distributed phishing disruption system,” in eCrime Researchers Summit, 2009, pp. 1–12.

[19] L. James, Phishing Exposed. Syngress Publishing, 2020.

[20] J. S. Downs, M. Holbrook, and L. F. Cranor, “Behavioral response to phishing risk,” in Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit, ser. eCrime ’07. New York, NY, USA: ACM, 2007, pp. 37–44.

[21] H. Huang, J. Tan, and L. Liu, “Countermeasure techniques for deceptive phishing attack,” in International Conference on New Trends in Information and Service Science, 2009. NISS ’09, 2009, pp. 636 – 641.

[22] T. Moore and R. Clayton, “Examining the impact of website take-down on phishing,” in eCrime ’07: Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit. New York, NY, USA: ACM, 2007, pp. 1–13.

[23] S. Egelman, L. F. 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, ser. CHI ’08. New York, NY, USA: ACM, 2008, pp. 1065–1074.

[24] 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, ser. CHI ’06, New York, NY, USA, 2006, pp. 601–610.

[25] S. Sheng, B. Wardman, G. Warner, L. F. Cranor, J. Hong, and C. Zhang, “An empirical analysis of phishing blacklists,” in Proceedings of the 6th Conference in Email and Anti-Spam, ser. CEAS’09, Mountain view, CA, July 2009.

[26] Google, “Google safe browsing API,” http://code.google.com/apis/ safebrowsing/, accessed Oct 2011.

[27] Google, “Protocolv2Spec,” http://code.google.com/p/ google-safe-browsing/wiki/Protocolv2Spec, accessed Oct 2011.

[28] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, “Phishnet: predictive blacklisting to detect phishing attacks,” in INFOCOM’10: Proceedings of the 29th conference on Information communications. Piscataway, NJ, USA: IEEE Press, 2010, pp. 346–350.

[29] Y. Cao, W. Han, and Y. Le, “Anti-phishing based on automated individual white-list,” in DIM ’08: Proceedings of the 4th ACM workshop on Digital identity management. New York, NY, USA: ACM, 2008, pp. 51–60.

[30] N. Chou, R. Ledesma, Y. Teraguchi, and J. C. Mitchell, “Client-side defense against web-based identity theft,” in NDSS. The Internet Society, 2004.

[31] T. Holz, C. Gorecki, K. Rieck, and F. C. Freiling, “Measuring and Detecting Fast-Flux Service Networks,” in Proceedings of the Network and Distributed System Security Symposium (NDSS), 2008.

[32] P. Likarish, D. Dunbar, and T. E. Hansen, “Phishguard: A browser plug-in for protection from phishing,” in 2 nd International Conference on Internet Multimedia Services Architecture and Applications, 2008. IMSAA 2008, 2008, pp. 1 – 6.

[33] D. L. Cook, V. K. Gurbani, and M. Daniluk, “Phishwish: A stateless phishing filter using minimal rules,” in Financial Cryptography and Data Security, G. Tsudik, Ed. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 182–186.

[34] Y. Zhang, J. I. Hong, and L. F. Cranor, “Cantina: a content-based approach to detecting phishing web sites,” in Proceedings of the 16th international conference on World Wide Web, ser. WWW ’07. New York, NY, USA: ACM, 2007, pp. 639–648.

[35] T. A. Phelps and R. Wilensky, “Robust Hyperlinks and Locations,” DLib Magazine, vol. 6, no. 7/8, Jul. 2000.

[36] M. Sharifi and S. H. Siadati, “A phishing sites blacklist generator,” in Proceedings of the 2008 IEEE/ACS International Conference on Computer Systems and Applications, ser. AICCSA ’08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 840–843.

[37] K.-T. Chen, J.-Y. Chen, C.-R. Huang, and C.-S. Chen, “Fighting phishing with discriminative keypoint features,” Internet Computing, IEEE, vol. 13, no. 3, pp. 56 –63, may-june 2009.

[38] C.-R. Huang, C.-S. Chen, and P.-C. Chung, “Contrast context histograman efficient discriminating local descriptor for object recognition and image matching,” Pattern Recogn., vol. 41, pp. 3071–3077, October 2008.

[39] M. Hara, A. Yamada, and Y. Miyake, “Visual similarity-based phishing detection without victim site information,” in IEEE Symposium on Computational Intelligence in Cyber Security, 2009. CICS ’09, 2009, pp. 30 – 36.

[40] G. Liu, B. Qiu, and L. Wenyin, “Automatic detection of phishing target from phishing webpage,” in Pattern Recognition (ICPR), 2010 20th International Conference on, aug. 2010, pp. 4153 –4156.

[41] H. Kim and J. Huh, “Detecting dns-poisoning-based phishing attacks from their network performance characteristics,” Electronics Letters, vol. 47, no. 11, pp. 656 –658, 26 2011.

[42] H. Zhang, G. Liu, T. Chow, and W. Liu, “Textual and visual contentbased anti-phishing: A bayesian approach,” IEEE Transactions on Neural Networks, vol. 22, no. 10, pp. 1532 –1546, oct. 2011.

[43] A. Y. Fu, L. Wenyin, and X. Deng, “Detecting phishing web pages with visual similarity assessment based on earth mover’s distance (emd),” IEEE Trans. Dependable Secur. Comput., vol. 3, no. 4, pp. 301–311, Oct. 2006.

[44] L. Ma, B. Ofoghi, P. Watters, and S. Brown, “Detecting phishing emails using hybrid features,” in Proceedings of the 2009 Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing, ser. UIC-ATC ’09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 493–497.

[45] P. Likarish, D. Dunbar, and T. E. Hansen, “B-apt: Bayesian anti-phishing toolbar,” in IEEE International Conference on Communications, 2008. ICC ’08, 2008, pp. 1745 – 1749.

[46] A. Bergholz, J. De Beer, S. Glahn, M.-F. Moens, G. Paaß, and S. Strobel, “New filtering approaches for phishing email,” J. Comput. Secur., vol. 18, pp. 7–35, January 2010.

[47] F. Toolan and J. Carthy, “Phishing detection using classifier ensembles,” in eCrime Researchers Summit, 2009. eCRIME ’09., 20 2009-oct. 21 2009, pp. 1 –9.

[48] “How to recognize phishing email messages or links,” http: //www.microsoft.com/security/online-privacy/phishing-symptoms.aspx, accessed March 2011.
[49] “How not to get hooked by a phishing scam,” http://www.onguardonline. gov/topics/phishing.aspx, accessed March 2011.

[50] “Avoiding getting hooked by phishers,” http://www.fraud.org/tips/ internet/phishing.htm, accessed March 2011.

[51] S. Sheng, B. Magnien, P. Kumaraguru, A. Acquisti, L. F. Cranor, J. Hong, and E. Nunge, “Anti-phishing phil: the design and evaluation of a game that teaches people not to fall for phish,” in Proceedings of the 3rd symposium on Usable privacy and security, ser. SOUPS ’07. New York, NY, USA: ACM, 2007, pp. 88–99.

[52] C. V. Zhou, C. Leckie, S. Karunasekera, and T. Peng, “A self-healing, self-protecting collaborative intrusion detection architecture to traceback fast-flux phishing domains,” in NOMS Workshops 2008. IEEE Network Operations and Management Symposium Workshops, 2008, 2008, pp. 321 – 327.

[53] L. Cranor, S. Egelman, J. Hong, and Y. Zhang, “Phinding phish: An evaluation of anti-phishing toolbars,” www.cylab.cmu.edu/files/ cmucylab06018.pdf, 2006, accessed Oct 2011.

[54] W. D. Yu, S. Nargundkar, and N. Tiruthani, “Phishcatch – a phishing detection tool,” in 33rd Annual IEEE International on Computer Software and