**LITERATURE SURVEY – WEB PHISHING DETECTION**

**ABSTRACT:**

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 defence, correction, and prevention, which we believe is critical to present where the phishing detection techniques fit in the overall mitigation process.

**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.

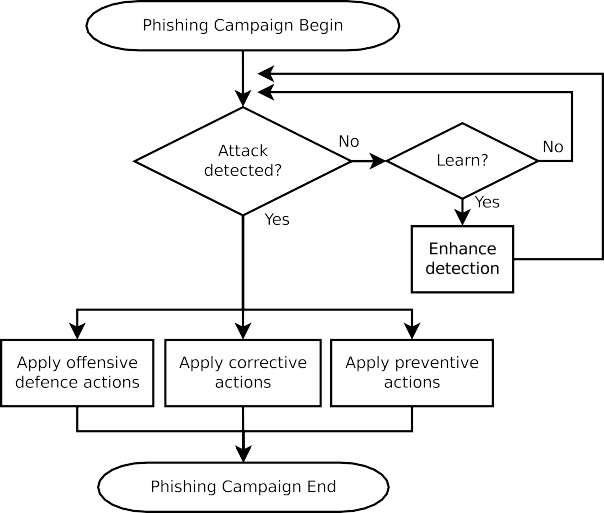
**DEFINITION:**

“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 (although no accurate studies have been made to reliably quantify.)

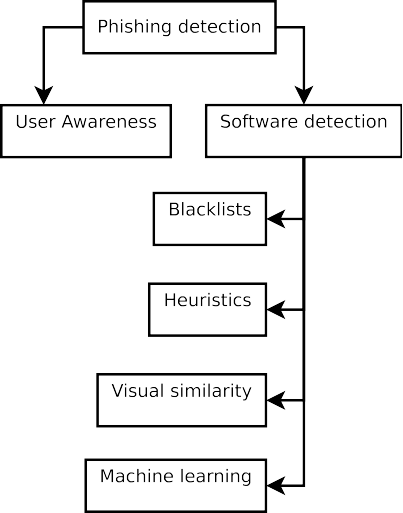
**MITIGATION OF PHISHING ATTACKS: AN OVERVIEW:**

Due to the broad nature of the phishing problem, we find important to visualise the life-cycle of the phishing attacks, and based on that categorise 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. 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.



**Detection Approaches**

In this survey, we consider any anti-phishing solution that aims to identify or classify phishing attacks as detection solutions. This includesUser 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 cate- gorization in which 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 categorise them under “detection”.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 user- training may not be feasible in some scenarios (such as when the user base is huge, e.g. PayPal, eBay, etc. . . ).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 classifi- cation 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 honeypotsto 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.The primary focus of this survey is covering the detection phase of phishing attacks. Figure [4](https://docs.google.com/document/d/1E4HGZo49aGUja5MtubiMW3k9WLQkBQXs/edit#heading=h.4i7ojhp) depicts an overview of phishing detection techniques that are covered in the subse- quent sections of this survey.Monitored network decoys that can be used to distract attackers from attacking valuable resources, provide early warnings about new attack trends, or enable in-depth analysis of the performed attacks.For example, all of the mitigation techniques, such as *correction*, *prevention* and *offensive defence* depend on a functional and accurate *detection* phase.



**Offensive Defence Approaches:**

Offensive defence solutions aim to render phishing cam- paigns useless for the attackers by disrupting the phishing campaigns. This is often achieved by flooding phishing web- sites with fake credentials so that the attacker would have a difficult time to find the real credentials.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 end- user 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 Bo- gusBiter.
    - The empirical effectiveness of this solution is not accurately measured.

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 defence approaches can theoretically make the attacker's 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 defence approaches are not accurately evaluated and can be questioned.

**Correction Approaches:**

Once a phishing campaign is detected, the correction pro- cess 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 work- stations in a botnet.

E-mail messages — could be sent from a variety of sources, such as: free Email 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)).A botnet is a number of infected computers controlled by attackers for function, and the way many VoIP provider systems are configured, spoofing Caller IDs are used by attackers as tools to increase their persuasion.In order to correct such behaviour, responsible parties (e.g. service providers) attempt to take the resources down. For example:

* Removal of phishing content from websites, or suspen- sion of hosting services.
* Suspension of email accounts, SMTP relays, VoIP ser- vices
* 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 organisations that provide brand protection services to their clients, which may include banking and financial companies that are possible vic- tims of phishing attacks. When phishing campaigns are iden- tified, 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.

**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, lawsuits and penalties against attackers by Law Enforcement Agencies (LEAs) are considered as prevention techniques.

In this survey, whenever the keyword “prevention” is men- tioned, it refers to the second previous item which is minimiz- ing 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 com- mon 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 equipments used to convey the attacks.

**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 vic- tims 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 [.](https://docs.google.com/document/d/1E4HGZo49aGUja5MtubiMW3k9WLQkBQXs/edit#heading=h.3dhjn8m) This section will present and discuss some of the work contributed in the field of user training in relation to phishing attacks.

**PHISHING VICTIMS**

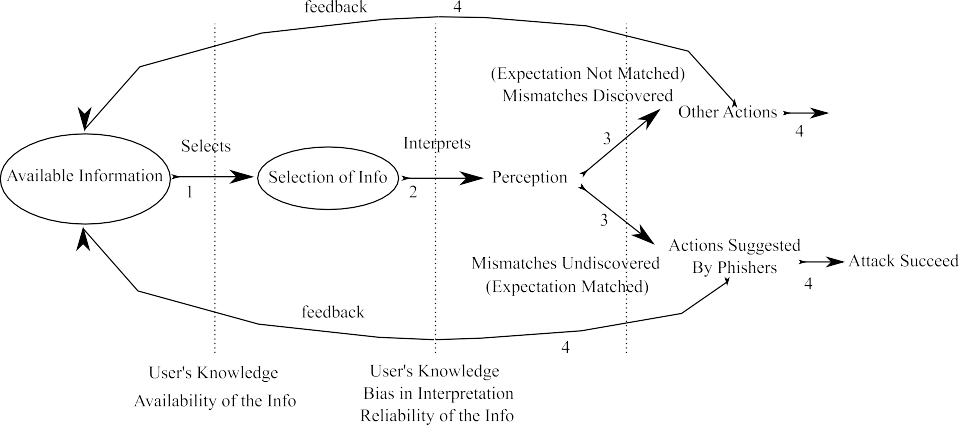
Julie S. Downs et al surveyed 232 computer users to study what are the criteria that can predict the susceptibility of a user to fall victims for phishing emails. The survey was formed in a role play where each user was expected to analyze emails as well as answering a number of questions. The outcome of the study was that those who had a good knowl- edge 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]](https://docs.google.com/document/d/1E4HGZo49aGUja5MtubiMW3k9WLQkBQXs/edit#heading=h.l7a3n9) was made by Huajun Huang et. al. [[21],](https://docs.google.com/document/d/1E4HGZo49aGUja5MtubiMW3k9WLQkBQXs/edit#heading=h.356xmb2) 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.

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



Generation of possible solutions: users usually find solu- tions 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 reflect 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 labelled “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 pro- cess of users through providing *external information* through the UI. The user interface provides two data sets (see Figure [6):](https://docs.google.com/document/d/1E4HGZo49aGUja5MtubiMW3k9WLQkBQXs/edit#heading=h.3tbugp1)

* + - 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 *metadata* and *content data* are legitimate.

Users may use *meta data* to decide whether an email message is legitimate. Phishers may also spoof *metadata* in order to further trick the users. As stated in[,](https://docs.google.com/document/d/1E4HGZo49aGUja5MtubiMW3k9WLQkBQXs/edit#heading=h.3ls5o66) the solution to the meta-dataintegrity 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 *metadata* 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.

**Service Policies:**

Although different methods have been developed to deliver warnings and notifications to end-users, *them* having a knowl- edge 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. . . ). Organisations should have strict policies against the distri- bution 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 chance of detecting inconsistencies within a phishing message if the *content data*, for example, asked for confidential infor- mation.

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[,](https://docs.google.com/document/d/1E4HGZo49aGUja5MtubiMW3k9WLQkBQXs/edit#heading=h.1kc7wiv) 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. Thun- derbird 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 behaviour could break the trust model of Public Key Infrastructure (PKI), making Hypertext Transfer Protocol Secure (HTTPS) or Secure/Mul- tipurpose 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 behaviour. Due to the generic nature of the challenge, a highly predictable solution remains difficult to achieve, if not impossible by today’s scien- tific advancements in understanding human mental functions and behaviour.

**Passive and Active Warnings:**

User interfaces can show security warnings based on trig- gered actions, such as viewing phishing web pages, as com- monly 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 depicted in Figure [7.](https://docs.google.com/document/d/1E4HGZo49aGUja5MtubiMW3k9WLQkBQXs/edit#heading=h.37m2jsg)
  + 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 depicted

The advantage of this embedded training method is that it teaches the end-user in the most receptive moment — which is when he falls victim to a fake phishing attack.

Kumaraguru et. al.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 ex- plain the causes; warning messages can fail if they have too much textual data.
  + The warning message should clearly and concisely ex- plain proper actions to be taken by the end-user to enhance his/her security.
  + The warning should not be delayed, but be shown im- mediately following the moment when the user falls 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 the 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 phishing- like messages, and the proposed solution’s performance is heavily based on the performance of the administrator who in turn is a variable as not all administrators are equivalent.

**PhishNet: Predictive Blacklisting:**

Any changes to a Phishing URL would result in no match. PhishNet [[28]](https://docs.google.com/document/d/1E4HGZo49aGUja5MtubiMW3k9WLQkBQXs/edit#heading=h.4hr1b5p) addresses the *exact match* limitation found in blacklists.

To solve this, PhishNet processes blacklisted URLs (par- ents) 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.](http://www.abc.com/online/ebay.html)
    - and [http://www.xyz.com/online.paypal.com.](http://www.xyz.com/online.paypal.com)

would result into forking the following children URLs:

* + - [http://www.xyz.com/online/ebay.html.](http://www.xyz.com/online/ebay.html)
    - and [http://www.abc.com/online.paypal.com.](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.](http://www.abc.com/online/ebay.php?ABC)
    - and [http://www.abc.com/online.paypal.com?XYZ.](http://www.abc.com/online.paypal.com?XYZ) would result into forking the following children URLs:
    - [http://www.abc.com/online/ebay.php?XYZ.](http://www.abc.com/online/ebay.php?XYZ)
    - and [http://www.abc.com/online.paypal.com?ABC.](http://www.abc.com/online.paypal.com?ABC)
  + Brand name equivalence: Multiple children URLs with same URL, but with different brand names. For example [http://www.abc.com/online/paypal.html,](http://www.abc.com/online/paypal.html) would also fork <http://www.abc.com/online/ebay.html> as a child (since they are often attacked).

The heuristics above are designed such that the URL varia- tions (as made by the attackers) are detected. For example, the *brand name equivalence* heuristic aims at detecting a common URL alteration that was performed by a phishing gang, namely RockPhish, as mentioned in [[28].](https://docs.google.com/document/d/1E4HGZo49aGUja5MtubiMW3k9WLQkBQXs/edit#heading=h.4hr1b5p)

Many of the forked children URLs may not exist or may be innocent irrelevant pages. So in order to filter/remove non- existent children URLs, the following tests are performed:

* + 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).
  + Content similarity: is the content similar to parent Phish- ing attack? If the page is different, it might be an innocent page. An external tool was used to measure the content similarity.

Google Safe Browsing API enables client applications to validate whether a given URL exists in blacklists that are con- stantly updated by Google [[26].](https://docs.google.com/document/d/1E4HGZo49aGUja5MtubiMW3k9WLQkBQXs/edit#heading=h.3im3ia3) Although the protocol is still experimental, it is used by Google Chrome and Mozilla Fire- fox. The current implementation of the protocol is provided by Google, and only consists of two blacklists named *goog-phish- shavar* and *goog-malware-shavar*, for phishing and malware respectively.