A Middleware based Anti-Phishing Architecture

A.A Orunsolu
Department of Computer Science
Moshood Abiola Polytechnic
Abeokuta, Nigeria
orunsolu.abdul@mapoly.edu.ng

A.S Sodiya
Department of Computer Science
Federal University of Agriculture,
Abeokuta, Nigeria
sodiyaas@funaab.edu.ng

A.T Akinwale
Department of Computer Science
Federal University of Agriculture,
Abeokuta, Nigeria

ABSTRACT

Phishing attacks are becoming an everyday threat to the ever growing cyber community. Regrettably, most online users do not understand some of the simplest indicators of a typical phishing scam. In addition, the sophistication of some of the newest phishing defeat most of the current software-based against phishing attacks.

countermeasure and anti-phishing education. In this work, a new paradigm-shift architecture is proposed after extensive survey of current client/server-based anti-phishing techniques. Although the architecture is at implementation stage, we present this paper to communicate the state of anti-phishing research to support the efficiency of the new approach in the fight

CCS Concepts

Computers and Society →Electronic Commerce – security, payment schemes, electronic data interchange (EDI)

Keywords

Attacks, E-Commerce, Middleware, Phishing, Internet

1. INTRODUCTION

The prevalence of e-services in today's digital world has opened a door for various cyber-crimes that threatened the acceptability of such services. Hackers have continuously managed a host of online black markets which discourage stakeholders' confidence in the usability of internet services [13]. This range of criminal enterprises includes spam-advertised commerce, botnet attacks, and a vector for propagating malware [4]. Among all the cybercrimes targeting e-services, phishing attacks have become a significant security threat which causes tremendously losses every day to both experienced and unwary internet users [5]. This is mostly due to the unhealthy disclosure of user's credentials to a phishing-related sites, chats, SMS or e-mail. Thus, these crimes have subjected the popular advantages of Internet to debate as businesses, government, individuals and financial institutions recorded millions of dollars in losses and espionage.

Phishing is e-communication criminal act which uses social engineering and technical subterfuge to exploit unwary internet users and acquire their confidential data such as credit card number, PIN, password, answer to security questions etc. Social engineering-based phishing techniques use spoofed emails, chat or SMS to lead internet users to fake agents, websites etc. On the other hand, technical subterfuge-based phishing scheme plant crime ware unto computers to steal sensitive data. Recently, phishers develop "*ransomware*" which executes a cryptovirology attack that adversely affects computing resources and demands a ransom payment to restore the resources to original state. According to an online report by CSO, 93% of phishing emails are now "*ransomware*". The report observed that most victims tend to pay quickly because of the sensitive nature of their resources [3].

Basically, a typical phishing attack begins with unauthenticated message crafted by phishers. These messages arrived at the client or user's machine in the form of email, e-advert, SMS, websites etc. with brand logos and call center number of a known company. One of the core features of these messages is their deceptive view which may not be easily identified even to an experienced IT-expert [5, 8]. The user falls for a phish by actively following the instruction in the message through performing a click action or download action. In the end, the user's actions result to the execution of phishers' payload. A payload is the functional part of a phisher's code where their malicious intention is achieved. Figure 1 presents the life cycle of a phishing attack.

Ravaged by unhealthy reality of phishing attacks, researchers proposed a number of countermeasures ranging from user-education to software enhancements. In spite of the existence of various anti-phishing measures, the frequency of phishing incidences continues to increase [14, 29]. For instance, RSA's online fraud report showed estimated losses of over $4.6 billion by global organizations in 2015. In a similar vein, the Central Bank of Nigeria White paper estimated that about $250 million was lost to cybercrime in 2013 [15].

To this end, we report the survey of anti-phishing researches and examine their weaknesses. After survey of relevant extant literature, we provide a brief discussion on a new approach that will effectively mitigate the weakness of the current approaches. This is a very important milestone in harnessing diverse anti-phishing defense system in one study to provide the basis for evaluating the proposed paradigm-shift approach.

The rest of the paper is organized as follows: Section 2 presents related works on why phishing works. The overview of the current anti-phishing defense architecture is examined in Section 3. In Section 4, we present the proposed paradigm-shift architecture to address current challenges. Section 5 presents our conclusions.

2. WHY PHISHING WORKS?

A number of studies have examined the reasons that people fall for phishing attacks. For instance, Dhamija et al. [5] identified lack of computer system knowledge, lack of knowledge of security and security indicators, visual deception and bounded attention. The authors further showed that a large number of people cannot differentiate between legitimate and phishing web sites, even when they are made aware that their ability to identify phishing attacks are being tested. In another related work, Down et al. [6] conducted a research in which 20 non-expert computer users revealed their strategies and understanding when faced with possible suspicious e-mails.

CoRI'16, Sept 7–9, 2016, Ibadan, Nigeria.

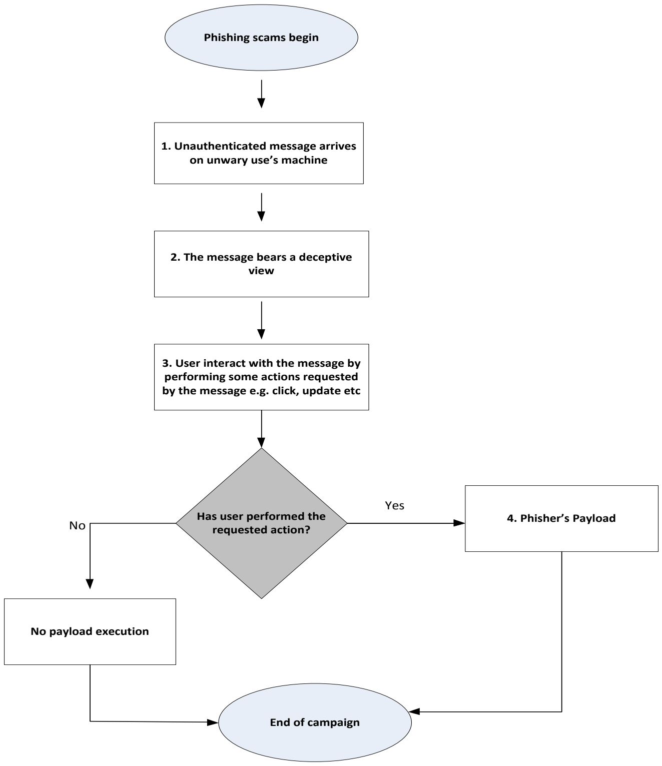
Phishing scams begin
1. Unauthenticated message arrives on unwary use's machine
2. The message bears a deceptive view
3. User interact with the message by performing some actions requested by the message e.g. click, update etc
Has user performed the requested action?
Yes
No
4. Phisher's Payload
No payload execution
End of campaign

Figure 1. Life cycle of a phishing attack

The investigation showed that participants used basic, often incorrect heuristics in deciding how to respond to email messages. In another development, Sheng et al [31] and Jakobsson et al. [11] provided useful insights on why phishing works using demographic data. While Sheng et al [31] revealed that women are more vulnerable than men due to their less exposure to technical knowledge, Jakobsson et al [12] revealed users' sensitivity to variety of common trust indicators such as logos, padlock icons etc. when navigating web pages. Jagatic et al. [11] researched into a more sophisticated spear-phishing attacks in which the attackers use specific knowledge of individuals and their organizations to conduct attack. Their investigation showed that people were 4.5 times more likely to fall for phish sent from an existing contact over standard phishing attacks. This is why social networking sites like Facebook are now more patronized by phishers.

Appealing to people's sense of greed is an ancient technique now adapted to the digital world especially in phishing scams [10]. This kind of phishing scam may look like online survey in which unsuspicious users are promised some financial returns for participating in the survey exercise. In a similar vein, phishers might pose as relief agency asking for help with recent natural disasters to appeal to people's sense of emotion [10]. Most unsuspecting users may not suspect anything negative even when asked to provide their financial details because of some gory pictures that usually accompanied such campaigns.

In a more recent study, Mohammed et al. conducted user study with the use of eye tracker to obtain objective quantitative data on user judgment of phishing sites. Their results indicated that users detected 53% of phishing sites even when primed to identify them with little attention on security indicators [20].

3. THE CURRENT COUNTERMEASURES

In this section, we considered the state of current countermeasures against phishing attack from software enhancement perspective. Software enhancement techniques are computer programs that are designed to defeat or mitigate phishing attacks. These software approaches use techniques such as list-based, machine learning, visual similarity and multi-channel authentication algorithms. They are either deployed on the client side or server side.

3.1 Client-side Anti-phishing approaches

PhishNet [26] proposed an active blacklist approach in which new malicious URLs can be effectively predicted from the existing blacklist entries. This is achieved by processing blacklisted URLs and producing multiple variations of the same URL using IP address equivalence, query string substitution, brand name equivalence, directory structure similarity and top level domain replacement. In this way, multiple variations of the same URL called children are obtained. In order to filter non-existent children URLs, the system performed DNS query, TCP connect, HTTP header response and content similarity. The approach achieved remarkable results during real-time blacklist feeds against new malicious URLs. However, the problem of false positives still exists.

PhishZoo [1] built profiles of trusted websites based on fuzzy hashing techniques in a whitelisted based approach. The approach also used blacklisting and heuristics approaches to warn users about malicious sites. This approach compared the stored profile of authentic sites with the content of sites under investigation. The approach achieved significant accuracy rate of about 96% with the possibility of defeating zero-day attack. However, there is lack of generalization to new phishing due to human interventions.

Cao et al [4] developed an Automated Individual White-List (AIWL) in which the record of well-known benign sites visited by users is kept. In this way, AIWL maintains a record of every URL along with its Login User Interface information where the user input his or her details to prevent unhealthy disclosure of confidential information to malicious sites. The LUI information maintains by AIWL for any suspicious website include the URL, the Input Area and the IPs. The URL refers to the Unified Resource Locator of the website. The input area includes the form username path and password path. The IPs is a list of legitimate IP addresses mapping to a URL. This method is very effective against pharming and dynamic phishing attacks. However, the problem of new login can result in false alert

In the work of Downs et al [6], a behavior-based phishing detection system (UBPD) which monitor submission of user credentials by building binding relationship between users and web pages was proposed. This is done by constructing a personal whitelist for the user by adding web sites the user has visited more than three times. UBPD consists of three components namely the user profile, the monitor and the detection engine. The user profile contains data to describe the user's binding relationships and the user's personal whitelist. The monitor collects the data the user intends to submit and the identity of the destination websites. The detection engine uses the data provided by the monitor to detect phishing websites and update the user profile if necessary. The approach can be effectively applied to static authentication credentials such as user name, password, security questions etc. However, zero day attack is possible since prediction is only applied to websites that user once visited.

Gowtham et al [8] presented a dynamic defense approach in which direct and indirect links in associated with a malicious page is generated. In this way, the target domain set is constructed as input into Target Identification algorithm to recognize a phishing page. Using DNS lookup and IP address resolution, the suspicious page can be predicted without the use of machine learning algorithms or existing restriction lists. The accuracy rate of this approach was 99.62%. However, the prediction of this approach is largely dependent on the TF-IDF algorithms, search engine speed and DNS lookup. The unavailability of any of these, defeat the efficacy of this approach.

A model to test the trustworthiness of suspected phishing page was developed in [30] by checking if the response of websites matches with the known behavior of phishing or legitimate sites. The model used the notion of Finite State Machine to capture the submission of forms with random inputs and then their corresponding responses to describe the website's behavior. The experimental results showed zero false negative and positive rates. The ability to detect advanced XSS-based attacks is another plus for this method. However, the approach cannot handle phishing attacks where images are employed.

PhishAri [2] detects phishing on Twitter in real-time. The approach uses Twitter specific features along with URL features to detect whether a tweet posted with a URL is phishing or not. The features used in this approach are classified into URL based, Tweet-based, WHOIS-based and Network-based. The approach is implemented as a Chrome browser extension which makes a call to a developed API (called RESTful API) and accordingly shows an indicator next to each tweet indicating whether the tweet is phishing or not. Experimental result shows that the system achieves 92.52% accuracy. The system detection speed can be improved with presence of external database repositories. However, XSS attack is still possible

In another work, Islam et al [13] proposed a multi-stage methodology that employed natural language processing and machine learning algorithm to detect phishing attack and discover the organization that the attackers impersonated during phishing attacks. The approach first discovered named entities and hidden topics in a suspected message using Conditional Random Field and Latent Dirichlet Allocation after parsing the message with the Multipart Internet Mail Extension Parser and HTML parser. In the next stage, utilizing topics and named entities as features, each message was classified as phishing or non-phishing using AdaBoost. In the final stage, the approach discovered the impersonated organization using CRF. The approach ensured automatic discovery of an impersonated entity, which help the legitimate organization to take necessary action against the offending site. The problem of scalability, false positives and the requirement of an efficient parser still exist.

The work of Maurer et al [21] focused on URL similarity for detecting phishing pages by extracting and verifying different terms of a URL using search engine recommendation. The authors developed algorithms to detect possible search terms that were worth checking using basename, subdomains, pathdomain and brand name. Top Level Domain was used to extract the base domain that was used with the search engine. The approach was evaluated with a large set of 8730 URLs from online phishing website database. The approach is effective against software toolkits that launch a large number of phishing pages using different URLs. However, high false positive rates affect the efficiency of this approach. In addition, significant performance issues like high overhead resulted as the system relies on consecutively querying search engines to identify legitimate domain.

An offensive approach in which a large number of bogus credentials are transparently fed into a suspected phishing page was proposed in [30]. In this way, the victim's real credential is concealed among bogus credentials thereby increasing the overhead on phishers' side in discerning the real credentials. BogusBiter consists of four main modules: information extraction, bogus credential generation, request submission and response process. The information extraction module extracts the username and password pair and its corresponding form element on a login page. The bogus credential generation module generates bogus credential based on an original credentials. The request submission is responsible for spawing and submitting multiple HTTP requests. The response process module determines the legitimacy of a website based on its response to HTTP requests. The approach is not bound to any specific phishing detection scheme and can be incrementally deployed over the internet. However, this approach can result in increased bandwidth overhead and it can be blocked by phisher since the bogus credentials is being submitted by a dedicated IP address.

3.2 Server-side Anti-phishing Approaches

The deployment of server-side anti-phishing defense system is not very popular as client side solutions. One of such server-side based solution is a practical authentication service in which the need for preset user password is eliminated during information flow between the client and the server [16]. This is achieved through the use of one-time passwords delivered on demand via a reliable secondary communication channel. On the receipt of the OTP, the user can login before the password expires. The proposed approach involves two processes namely a registration process and a login process with four participating entities: websites, instant messaging service provider, users and phishers. In the registration process, a user choose a unique account name, select a login password, fill in all the required information fields, complete an additional IM account registration and provide at least one type of personal contact information. In the login process, the registered user can log in with the OTP assigned by the website. The approach does not suffer from client side vulnerabilities and cost of deployment is low which increases the practicability of this method. The approach cannot detected XSS attacks and phishing sites hosted on compromised domains

In another approach, Chen et al [7] proposed an image based anti-phishing strategy that measure suspicious pages' similarity to actual sites based on discriminative key point features in web pages is proposed. The approach defined three aspects of visual similarity consisting of block-level similarity, layout similarity and overall style similarity to compare pages during detection process. Their invariant content descriptor, which uses the contrast context histogram, computes the similarity degree between suspicious and authentic pages. The proposed method takes a snapshot of a suspected page and treats it as an image throughout the detection process. It uses CCH to capture invariant information around discriminative key points on the suspect page and then match the descriptors with those of authentic pages that are often targeted by phishers. However, the approach cannot detect phishing pages in which phisher use images to mimic their target.

The concept of dynamic security skins that allow humans to distinguish one computer from another was proposed in [17]. Dynamic security skins allow a remote web server to prove its identity in a way that is easy for human user to verify and difficult for attackers to spoof. This approach assigns each user a random personalized photographic image that will always appear in the password window. However, the ability of user to recall this image is a subject of debate. In addition, it is difficult to convince web master to apply these rules in web page creation.

3.3 Summary of problems with the existing countermeasures

In this subsection, we itemized the summary of the problems with the existing anti-phishing system.

a. The inability of most existing anti-phishing countermeasures to efficiently detect newer phishing scams i.e. possibility of zero-day attacks which a type of attack mounted using hosts that are not blacklisted or using techniques that evade known approaches to phishing detection [25, 20]
b. Most of the existing countermeasures consider small set of heuristics features in their approach and most browsers' plugins anti-phishing solutions are susceptible to java vulnerabilities [25, 27]
c. Although there has been substantial performance improvement in detecting phishing, the foremost drawback of methods currently in use, in particular for classification based methods using statistical learning algorithms, continue to be the false positive problem [13]
d. High computational overhead of most classification-based anti-phishing countermeasures [13]
e. Lack of consensus and problems of coverage of most blacklist techniques. In addition, the blacklist method cannot adapt the filter to identify emerging rule changes in the intruders' attacks [19]
f. Intensive configuration and lack of users' proper attention with most client-side solutions [22]
g. Absence of holistic countermeasures that detect, prevent and disrupt phishing scams. Most existing anti-phishing system either focuses on phishing email or phishing website detection [6,7,13,19]

4. THE PROPOSED APPROACH

The phishing problem has been and still is very important, and the current detection and warning approach taken to address the problem is not enough. Motivated by this challenge, we proposed a paradigm-shift based architecture (Fig.3) based on middleware technology. The middleware technology is one of the viable alternatives to the challenges of client/server anti-phishing techniques. The primary advantage of MT is that it leverages the benefits of software as a service model. That is, software solution or design remains external to their system and is accessible and executable by a large numbers of individuals. The approach has potentially great benefits to anti-phishing design: MT is able to always keep the system up to date (fully maintained) as administration is under the control of service provider, ensure the anti-phishing service remains efficient (by automatically adding new filter rules as required), interacts with a large volume of data traffic which can be collated and analyzed for improved security coverage in the fight against phishing, provide a suitable basis for anomaly detection technology and the transparency it offers to both the client and server. Nevertheless, the MT technology raises the issue of scalability especially in a user intensive environment like internet; but with the emergence of cloud computing infrastructure this challenge can be easily leveraged [22].

In this paradigm-shift approach, we shall employ Map Reduce algorithm to aggregate web streams into different jobs as suggested [27]. Map Reduce is a programming model and software framework intended to facilitate and simplify the processing of vast amount of data in parallel on large clusters. The aggregation of tasks results into non-computational and computational classes. In the non-computational class (PhishDetect C1), phishing detection is done using list-based approach to reduce the unnecessary computation within the system. If a phishing attack cannot be detected by non-computational class, the computational class is invoked to complete the detection process. In this case, the extracted features from the suspected sites are compared with trained feature vectors from a hybrid classifier (NB-SVM). The proposed system will be implemented and evaluated using datasets from research sources such as PhishTank, APWG etc. The overview of the algorithm for the proposed scheme is presented in Figure 3

```
Get Web Document (webpages, email message, e-chat)
    Sort web document using Map-Reduce algorithms
    Generate the Mapper and the Reducer Function
    For each Mapper and Reducer Function, invoke non-computational Class
        If detection is accurately performed, the exit
        Else send uncompleted task to computational Class
            Trained NB-SVM classifier on feature class on the uncompleted task
            Classify the task and exit
```

Figure 2. Pseudo code of the proposed scheme

Stage 1: The first stage of the architecture is where client transactions are captured before being forwarded to the phishing detection manager (PDM). When a user opens a page in the web browser, the extension module accesses the DOM tree of the downloaded page from web browser's IFrame. Document Object Model, is a World Wide Web consortium standard, that allows programmers and scripts to dynamically access and update the content, structure and style of documents. After the construction of DOM, the transaction is also parsed to extract any hyperlinks present in the body of the transaction or a webpage

At the same time, the transaction is also tokenizes in an attempt to identify the named entities such as organization and hidden topics that the phishers is trying to deceive the unsuspecting users. Named entities are proper names that are names of people, organization, location etc. in the body of a document. The robust Conditional Random Field (CRF) which is an information retrieval task that seeks to locate and classify elements in text documents as one of these proper names is employed. The second task of the tokenizer is to discover the hidden topics in a transaction which is achieved by employing the Latent Dirichlet Allocation. LDA is sensitive to changes in feature usage which make it good at handling synonym. It is also robust to polysemy, features with different meaning in different context. In addition, it can discover threatening theme in a message and intentionally misspelled features and conjoined features. The most powerful feature of LDA is its ability to discover multiple topics from a single document.

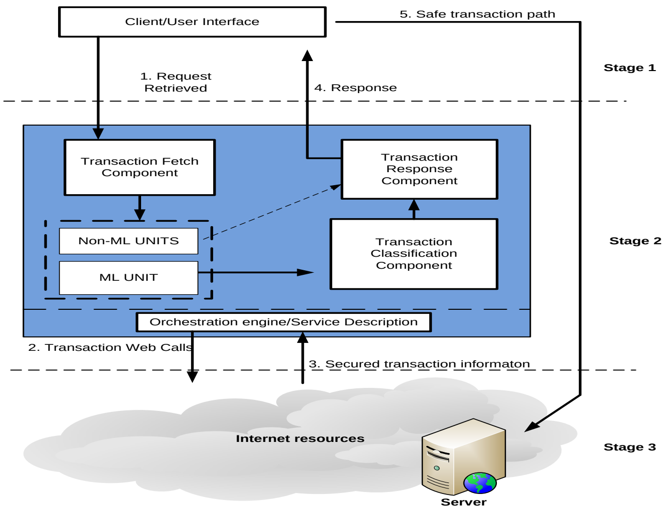
Client/User Interface
5. Safe transaction path
Stage 1
1. Request Retrieved
4. Response
Transaction Fetch Component
Transaction Response Component
Non-ML UNITS
ML UNIT
Transaction Classification Component
Stage 2
Orchestration engine/Service Description
2. Transaction Web Calls
3. Secured transaction informaton
Internet resources
Server
Stage 3

Figure 2. The Middleware-based anti-phishing architecture

Stage 2: This contains the core component of the proposed technique that detect the phishing label of a transaction. The phishing detection manager provides the link between the client interface and the Secured Server Side Transaction (Stage 3). The Orchestration Engine (OE) is responsible for managing communication between the Machine Learning Detection units and Non-Machine Learning Detection units. In this way, exception management, transaction management, resource management and components management are easily coordinated. The Phishing Detection Manager offers methods for all the basic tasks associated with the construction and interaction of the phishing detection process. The core components of the phishing detection manager are:

a. Transaction Fetch Component (TFC)
b. Transaction Preliminary Filter Component (TPFC)
c. Transaction Classification Component (TCC)
d. Transaction Response Component (TRC)

These four components are integrated into a Middleware system as anti-phishing scheme using service model architecture. That is, the anti-phishing scheme remains external to their system (i.e. the server and client). In addition, the system is accessible and executable by a large number of client machines irrespective of the browser type.

Transaction Fetch Component (TFC)

The Transaction Fetch Component represents the entry point of web requests into the Anti-Phishing System where billions of user transactions are aggregated after the DOM construction and tokenization for onward generation of phishing label with a cost efficient response. The task of aggregation is made computationally less expensive with the employment of Map Reduce framework. This is consistent with the suggestion of [27]. Map Reduce is a programming model and software framework intended to facilitate and simplify the processing of vast amounts of data in parallel on large clusters such as Internet Web Streams (IWS). The Map Reduce framework consists of a single master JobTracker and one slave Task Tracker per cluster-node. The master is responsible for scheduling the jobs' component tasks on the slaves, monitoring them and re-executing the failed tasks. The slaves execute the tasks as directed by the master. The core idea behind Map Reduce is mapping your data set into a collection of <key, value> pairs, and then reducing overall pairs with the same key. However, it can be more efficient to sort data once during insertion than sort them for each Map Reduce query. In the light of this, an insertion sorting technique is adopted to increase the efficiency of Map Reduce capability of TFC

Transaction Preliminary Filter Component (TPFC)

The output of Map Reduce algorithms provides the input into the Transaction Preliminary Filter Components which involves the following tasks:

1. Preliminary Transaction Filtering Module (PTFM) which determine phishiness of a transaction without learning algorithms using Anti-Phishing Dictionary with Customized Source Code Scanner, Anti-Phishing Authentication System with ability to detect abnormally in the login form and Phishing Toolkit Analyzer using Phishing Toolkit Corpus. The rationale for the introduction of this module is to reduce the system computation and enhances efficient memory usage in a time-critical scenario like web scape. This module is especially suited for preapproved sites and sites with known popularity.

2. Feature Selection Module (FSM) which determines efficient feature for classification in a Feature Generator Process using efficient feature selection approach. The main attraction of this module is to select most informative Comprehensive Anti-Phishing Feature for efficient classification. FSM takes advantage of the factors embedded within or surrounding a message (called heuristic cues) such as its source, format, length, and subject, to quickly make a validity assessment.

3.Cached Internet Resource Module which provides for faster lookup and speed up the phishing label of a transaction using data from WHOIS properties, Phish Tank, Crawling Instances etc. This is to reduce superfluous computation on already labeled suspicious webpage or transaction.

Transaction Classification Component

Given an identity and a set of features, the task of determining the genuineness of a transaction is executed by a classification algorithm. A classification algorithm automatically learns how to make accurate predictions based on past or trained observations. The Transaction Classification Component of HAPS uses a hybrid classifier approach to provide an efficient status of a transaction.

Naïve Bayes and Support Vector Machine are combined as hierarchical hybrid system model (NB-SVM) to maximize detection accuracy and minimize computational complexity. The NB is a relatively accurate classifier especially for large dimensional dataset like web streams. However, capacity control and generalization remains an issue. The main problem associated with using SVM as classifier is the computational overhead needed to transform text data into numerical data which is sometimes termed as “vectorization”. Generally in PDM, the features of a web transaction are directly vectorized by transforming the text documents into numerical format using SVM. Thus, NB is used as a pre-processor for selected features in the front end of the SVM to vectorize corpus before the actual training and classification are carried out. The motivations for the adoption of this hybrid classifier approach are:

i. Improve the generalization of the overall system
ii. Maintain a comparatively feasible training time and categorization time
iii. Overcome the limitations of list-based methods (e.g. blacklist approach) by dynamically updating the training patterns whenever there is new pattern during classification
iv. Ignore serious deficiencies in underlying algorithms of both classifiers
v. Produces a simple computationally effective and highly accurate classifier

Transaction Response Component

The Transaction Response Component provides a cost efficient response to a classified transaction based on the severity of attack as computed by the Threat Identification Module. The Transaction Identification Module measures and identifies the threat severity associated with a classified transaction. With classified transactions, a TIM is proposed to proactively predict the level of seriousness of the attack. This is necessary in advancing the notion of HAPS to a high level especially for accessing the severity of phishing campaign. Consider the TIM algorithm that assign a threat score, 0 ≤t_i ≤1, to the ith transaction upon the occurrence of the jth classification by PDM. The threat scores may qualitatively identify the threat level upon classification as compromised if t_i=1, threatened if 0<t_i<1, and unthreatened if t_i=0.

Stage 3: The third stage of the architecture ensures that only safe transaction are forward or return to client for the completion of the initiated task after necessary anti-phishing computation have been performed. The orchestration engine of the PDM also makes web calls into this stage when there is need for external sources of data in validating a transaction under investigation. All transactions are directed to benign server while malicious servers are bypassed.

5. CONCLUSIONS AND FUTURE WORK

As the rapid explosion of e-commerce witnessed unprecedented adoption by online communities, phishing activities continue to wreak havoc on unsuspecting users who access the e-commerce services. In the process, both users and the service providers have suffered millions of dollars in losses compare to any form of cybercrime. Therefore, phishing has become a plague that threatens stakeholders' confidence in the security of online product and services. Considerable researches have been done towards protecting users from phishing attacks. Despite the efforts by the research community, the industry, and law enforcement to develop solutions to tackle the problem, phishing has shown no sign of abating (Basnet et al. 2012) as each of these existing techniques suffers from such major challenges. In this paper, we provide survey of relevant literature from client/server-side perspective anti-phishing defense systems. We illustrated some open problems with the current counter strategy and make a case for a paradigm-shift defense system for a middleware-based approach. The middleware-based approach overcomes some inherent challenges of client and server-side approach through provision of enhanced security, ease of configuration, optimization of load-balancing, management of connections etc. In addition, we present a working architecture of the proposed method. Future works will consider the implementation of the proposed architecture on real-time phishing data corpus as well as benign data corpus.

6. REFERENCES

[1] Afroz, A., & Greenstadt, R. (2011). PhishZoo detecting phishing websites by looking at them. *In Proceedings of IEEE fifth international conference on semantic computing (pp. 368–375).*

[2] Aggarwaly, A., Rajadesingan, A., Kumaraguru, P. (2012). PhishAri: Automatic realtime phishing detection on twitter. *In Seventh IEEE APWG eCrime researchers summit (eCRS). Las Croabas, Puerto Rico, 22–25*

[3] CSO Online report on phishing activities. Accessed 2016 *(www.csoonline.com/articles)*

[4] Cao, Y., Han, W. and Le, Y. 2008. Anti-phishing based on automated individual white-list. *Proceedings of the 4th ACM Workshop on Digital Identity Management, Alexandria, USA.*

[5] Dhamija, R., Tygar, J.D. and Hearst, M. 2006.Why phishing works. *Proc. of the IGCHI Conference on Human Factors in Computing Systems, ACM Press, pp. 581-90.*

[6] Downs, J.S., M.B. Holbrook, and L.F. Cranor(2006). Decision strategies and susceptibility to phishing. *In Proceedings of the Second Symposium on Usable Privacy and Security (SOUPS 2006). pp. 79-90.*

[7] Chen, K., Chen, J., Huang, C. and Chen, C. (2009), "Fighting phishing with discriminative keypoint features", *IEEE Internet Computing, Vol. 13 No. 3, pp. 56-63.*

[8] Gowtham R, Krishnamurthi I. 2014. An efficacious method for detecting phishing webpages through target domain identification. *Journal of Decision Support Systems. Elsevier Press*

[9] Han W, Cao Y, Bertino E and Yong J. 2012.Using automated individual white-list to protect web digital identities. *Expert Systems with Applications.*

[10] Hong J. (2012). The state of phishing attacks. *Contributed Articles in the Communication of the ACM. Vol 55 No 1*

[11] Jagatic, T., Johnson, N., Jakobsson, M. and Menczer, F. (2007). Social phishing. *Communications of the ACM, Vol. 50*

[12] Jakobsson, M. and Myers S. A. (2007). Phishing and countermeasures: Understanding the increasing problem of identity theft. *Introduction to Phishing (Eds.), (pp. 1– 2). New York: John Wiley & Sons, Inc.*

[13] Islam R and Abawajy J. 2013. Multi-tier phishing detection and filtering approach. *Journal of Network and Computer Applications.*

[14] Kathryn P., Agata M., Malcolm P., Marcus B and Cate J. 2015. The design of phishing studies: Challenges for researchers. *Journal of Computers and Security.*

[15] Longe T. 2014. Ensuring Information Security Assurance through Policy Framework. *Proc. of First National Cyber Security Forum. Lagos. Nigeria*

[16] Huang C., Ma S and Chen K., (2011). Using one-time passwords to prevent password phishing attacks. *Journal of Network and Computer Applications. Elsevier Press..*

[17] Dhamija, R. and Tygar, J. D. (2005). The battle against phishing: Dynamic security skins. *In Proceedings of the Symposium on Usable Privacy and Security (SOUPS). 77–88.*

[18] Lovet, G. 2009.Fighting cybercrime: technical, juridical and ethical challenges. *Proceedings of the Virus Bulletin Conference.*

[19] Moghimi M and Varjani A.Y. (2016). New rule-based phishing detection method. *Journal of Expert Systems with Applications. Vol 53 pp. 231-242.*

[20] Mohammed A., Furkan A., and Sonia C. 2015. Why phishing still works: User strategies for combating phishing attacks. *International Journal of Human-Computer Studies. Volume 82. pp. 70-82. Elsevier Press*

[21] Maurer M and Hofer L. (2012). Sophisticated Phishers Make More Spelling Mistakes: Using URL Similarity Against Phishing. *Springer.*

[22] Ofuonye E and Miller J. (2013). Securing web-clients with instrumented code and dynamic runtime monitoring. *Journal of Systems and Software.*

[23] Pan Y and Ding X. 2006. Anomaly based web phishing page detection. *Proc. of the 22nd annual computer security applications conference.*

[24] Parno, B., Kuo, C. and Perrig, A. 2006. Phoolproof phishing prevention. Financial Cryptography and Data Security, *Lecture Notes in Computer Science, Vol. 4107, Springer, Berlin.*

[25] Purkait S. 2012. Phishing counter measures and their effectiveness- literature review. *Information Management and Computer Security Vol. 20 No. 5.*

[26] Prakash, P., Kumar, M., Kompella, R.R. and Gupta, M. (2010). Phishnet: predictive blacklisting to detect phishing attacks. *Proceedings of the 29th Conference on Information Communications, San Diego, CA, USA, pp. 346-50.*

[27] Ramanathan V and Wechsler H. 2013. Phishing detection and impersonated entity discovery using Conditional Random Field and Latent Dirichlet Allocation. *Journal of Computers and Security.*

[28] Ralf K, Peter F, and Wolfgang N. 2009. Latent Dirichlet Allocation for Tag Recommendation. *Proc. of RecSys ACM.*

[29] RSA Anti-Fraud Command Center. RSA monthly online fraud report, 2014.

[30] Shahriar H, Zulkernine M. 2011. Trustworthiness testing of phishing websites: a behavior model-based approach. *Future Generation Computer Systems.*

[31] Sheng, S., Holbrook, M., Kumaraguru, P., Cranor, L.F. and Downs, J. 2010. Who falls for phish? A demographic analysis of phishing susceptibility and effectiveness of interventions. *Proc. of the 28th International Conference on Human Factors in Computing Systems, USA.*

[32] Xiang, G., Hong, J., Rose, C.P. and Cranor, L. 2011 CANTINA+: a feature-rich machine learning framework for detecting phishing web sites. *ACM Transactions on Information and System Security*

[33] Yue, C. and Wang, H. 2010. BogusBiter: a transparent protection against phishing attacks. *ACM Transactions on Internet Technology, Vol. 10 No. 2, pp. 1-31*

[34] Zhang Y., Egelman S., Cranor L. and Hong J. 2007. Phishing Phish: Evaluating Anti-Phishing Tools. *Proc. of Network and Distributed Systems Security Symposium (NDSS)*

[35] Zhang, H., Liu, G., Chow, T.W.S. and Liu, W. 2011. Textual and visual content-based anti-phishing: a Bayesian approach. *IEEE Transactions on Neural Networks, Vol. 2*