An Enhanced Anti Phishing Approach Based on Threshold Value Differentiation

Komatla. Sasikala[1], P. Anitha Rani[2]

[1]Computer Science Engineering, Guntur Engineering College, Guntur, India

Abstract: ***Phishing is form of creating a similar look-a-like legal website and misleading the customers or users to user their identities or authentication keys such as usernames, passwords , pins to gain the control and then duping the customers by illegal activities such as interpreting data , financial accounts transfer etc mainly phishing is heavily seen in portals like banking, mails etc. This paper presents an anti phishing approach for detecting phishing attacks. Our approach combines a Threshold Value Differentiation approach with machine learning techniques. The Threshold is used as filter that blocks phish web pages used to imitate innocuous user behavior.***

Keywords: Phishing, Threshold, Differentiation, Machine learning techniques, Innocuous.

1. Introduction

According to the Anti Phishing Working Group (APWG) [1], phishing is a criminal mechanism employing both social engineering [2] and technical subterfuge to steal consumers' personal identification data, including financial account credentials. Usually, phishers trick users with spoofed e-mails which appear to be from a trusted source such as a bank or a reputable commerce agency.

There are two main classes of phishing attacks [3]: malware based phishing and deceptive phishing. Malware-based phishing methods install malicious software by exploiting security holes in the user's system. This software then records confidential and sensitive data and relays it to the phisher.

In deceptive phishing an attacker sends misleading e-mails which appear to come from trusted sources. These e-mails invite users to access a web link leading to a fake web site carefully designed to trick users into divulging targeted sensitive data. The phisher in this type of attack uses several techniques to dupe the user and Bergholz et al. [4] classified these approaches according to the following types:

- Social engineering, which includes all methods and scenarios invented by phishers to create a convincing context.
- Imitation, which consists of forging websites that look like legitimate ones.
- E-mail spoofing, which allows a phisher to spoof the source address of an e-mail.
- URL hiding, this enables phishers to mask the URL to which a user is redirected.

One popular anti-phishing solution is the "blacklist/whitelist" approach, where a blacklist [5], [6], [7] contains a list of URLs of websites known to be phishing sites. A webpage whose URL is present in this list is blocked from the user. Despite their high precision, the blacklists suffer from the "zero-day attack", a vulnerability window that exists before a phishing URL is recognized and added to the blacklist. A whitelist avoids this problem since it instead contains a list of legitimate URLs, webpages whose URL does not exist in the whitelist are qualified as suspect pages. The principal disadvantage of this solution is that it should ideally contain every legitimate site, which is impossible, and therefore leads to a large number of false positives. Several studies [8], [9], [10] propose improvements to the whitelist solution, such as personalized white lists that contain only the websites' URLs used by a particular user, thus avoiding the management and updating of large amounts of data. Despite this improvement, these customized whitelists suffer from the same false positive problems since they make a bold assumption: if a URL is not present in the list, it is considered suspicious and subsequently filtered.

We present a personalized whitelist based approach for automatic phishing webpage detection. Unlike previous work, our approach uses a whitelist in combination with a support vector machine (SVM) classifier. The phishing pages that are not blocked by the whitelist are passed to the SVM classifier. Our experimental results show that the proposed approach yields improvements compared to the existing methods.

The remainder of this paper is organized as follows: Section II presents the state-of-the-art of current anti-phishing solutions. Section III describes details about our proposed approach. Our experimental results are given in Section IV, and we conclude and provide some discussion in Section V.

2. State-of-the-Art

Several anti-phishing solutions have been proposed in the literature. We classify existing works according to the approach they use. We focus on the most pertinent approaches, citing related work in each category.

2.1. " Blacklist / Whitelist" Solutions

The blacklist solution is typically deployed as a toolbar or extension in web browsers. Examples of tools that implement this type of solution as Mozilla's Firefox [5], Google's safe browsing [6], and PhishTank [7].

Several studies have proposed improvements based on whitelisting. Cao et al. [8] propose an individual whitelist containing a user's usual login pages modelled as feature sets. Whenever a user attempts to establish a connection, the parameters of the connection are recorded from the login page are compared against those in the whitelist. The user is notified if no similarity is detected. For the whitelist construction, Cao and colleagues used a Bayesian classifier to identify the legitimate login pages. Reddy et al. [9] have proposed a whitelist solution in a similar context where the list contains the URLs of all Indian banks. A distance metric is calculated between the visited site's URL and the URLs of the whitelist using the Levenshtein edit distance algorithm [11]. If the distance is less than a threshold, an IP comparison is performed; if the IPs match then the page is considered legitimate. In [10], the whitelist is designed to be completed by the user and used only when the user tries to connect to a webpage by sending sensitive information.

2.2. Classification based solutions

This type of solution is based on artificial intelligence classification techniques. We categorize these solutions according to the classification subject, such as: classifications based on e-mails, URLs, or a webpage's content.

1) *E-mail-based classification:* This solution classifies emails in a similar manner as in spam filters. The e-mails in this case are classified as legitimate or phishing emails. This solution can be implemented at the level of the mail servers or at the client-side level. Khonji et al. [12] surveyed the most used features in this type of classification.

PILFER [13] is an example implementation of such an approach and uses a random forests algorithm. PILFER can detect 96% of phishing e-mails with only 0.1% rate of false positives. Salem et al.[14] propose an intelligent system based on fuzzy rules. The system classifies e-mails into three categories: safe e-mails, partially safe e-mails, and phishing e-mails. The proposed system was able to certify 95% of legitimate e-mails as safe and 5% as partially safe.

Abu-Nimeh et al. [15] present a comparative study between several machine learning classification techniques, namely: Logistic Regression Classification and Regression Trees, Bayesian Additive Regression Trees, SVMs, Random Forests and Neural Networks. This work used 43 features for a base model and 2889 e-mails consisting of 1171 phishing e-mails and 1718 legitimate e-mails. Except for a slight advantage from the Random Forest method, the results obtained during this student did not show a significant difference between the various methods.

2) *Content-based classification:* This type of solution classifies a webpage (legitimate or phishing) by evaluating its content. The most cited work for this approach is CANTINA [16]. This solution detects if a webpage is a phishing page by extracting a signature from the page content. This signature is composed of five words obtained by applying the TF-IDF algorithm (Term Frequency-Inverse Document Frequency). The signature is used as a keyword in a search engine query and if the page's URL is among the first results, the page is classified as legitimate. If not, it is classified as a phishing page. This method detects 90% of phishing pages, with a 1% rate of false positives. M. He et al. in [17] used 12 features with an SVM classifier. The approach detects 97% of the legitimate sites with a 4% rate of false negatives.

3) *URL-based classification*: This type of solution is similar to the preceding techniques where the only difference is that the classification features are based on the URL alone.

Garera et al. [18] studied URL characteristics of phishing sites and derived a set of features for use in an ULR classifier. They used a dataset of 2508 URLs composed of 1245 phishing URLs and 1263 legitimate URLs, 66% of URLs were used for training and the remaining 34% were used as the test set. This method can detect 95% of phishing site with a 1.2% rate of false positives. A very similar approach is proposed by Ma et al. [19] where three classification methods (naive Bayes, SVM and LR classifiers) were applied. Training and testing were conducted on four different datasets. The best score obtained was a 0.9% error rate and a 0.8% rate of false positives.

3. Our Approach

We propose a solution that combines a personalized whitelist and an SVM classifier. Figure I illustrate the main components of our proposed solution.

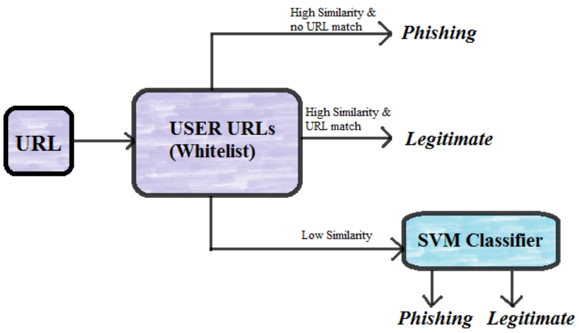


Figure 1: The Proposed Approach

Our solution is designed to be implemented as an extension to a web browser. When a user tries to access a webpage, a similarity metric is computed between the webpage and pages in the whitelist. Depending on the degree of similarity three cases can occur:

- Case 1: if there is a high similarity (sim > threshold) between the visited page and one of the whitelist pages, with different domain names, then the page is considered as a phishing site.
- Case 2: if there is a high similarity (sim > threshold) between the visited page and the pages of the whitelist with the same domain name, then the page is considered legitimate.
- Case 3: if there is a low similarity (sim < threshold) then the page is processed by the SVM classifier, which decides whether a page is legitimate or not.

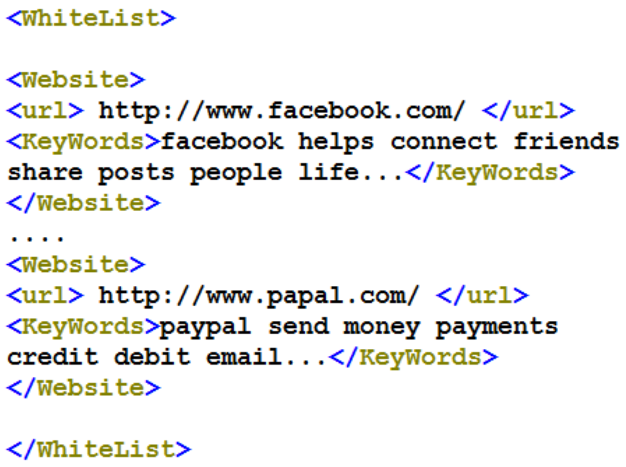
The rest of this section describes the structure of the whitelist as well as the features used in the SVM classifier.

3.1. Structure of the whitelist

The whitelist in our approach is an XML file that contains a user's login pages' URLs and a set of keywords (see Figure II). The key words are composed of the domain names of the page's URL and a set of terms from the Document Object Model (DOM) tree for the site. More precisely:

- The content of the "title" tag, ex: <title> text </title>.
- The content of the "meta keywords" tag, ex: <meta name ="keywords" content ="text"/>.
- The content of the "meta description" tag, ex: <meta name = "description" content ="text"/>.

These three tags provide a precise description of the webpage content. Stop words and punctuation symbols are omitted from the parsed field and the remaining words are concatenated and stored in the whitelist with the URL of the page. Figure II gives an example of the whitelist content.



```
<WhiteList>

<Website>
<url> http://www.facebook.com/ </url>
<KeyWords>facebook helps connect friends
share posts people life...</KeyWords>
</Website>
....
<Website>
<url> http://www.papal.com/ </url>
<KeyWords>paypal send money payments
credit debit email...</KeyWords>
</Website>

</WhiteList>
```

Figure 2: The structure of the whitelist.

The keywords are used to calculate similarity between a visited webpage and the pages of the whitelist. We used a *"bag of words"* model [20] for the construction of the keywords' frequency vector of each page and a *"cosine"* distance for similarity calculation. The similarity between a visited page "*Pv*" and a whitelist page "*Pl*" is calculated as follows:

Where: *fx,t* is the frequency of the term "*t*" in the set "*x*".
Two pages are similar if their cosine similarity is close to 1.

The whitelist contains the login pages of the top 10 most attacked sites (by means of phishing) [21], as well as the user's typically visited pages. This avoids any user intervention, which is often a source of error, and facilitates installation and use.

3.2. The Classification Features

If the visited web page has no similarity with the pages of the whitelist, a feature vector is constructed for the page and it is subsequently processed by our SVM classifier. We use eight features to represent a page P = <F1, F2, F3, F4, F5, F6, F7, F8 >. Some features are constructed according to the URL of the webpage, and others from its content.

- *Feature 1(F1): URL with IP address*

For cost minimization reasons, many phishing pages use an IP address instead a domain name, contrary to a legitimate site which is accessed most commonly through a hostname. A link that does not contain a domain name and requests sensitive information to users is probably a phishing site. The feature F1 is a binary feature: if the URL of a webpage contains an irresolvable IP address then $F1 = 1$, otherwise $F1 = 0$.

- *Feature 2(F2): special characters in the URL*

For this feature we tested the existence of the "@" character in the URL. The presence of this character in a URL forces characters before it to be treated as user login credentials to URL's intended site. Phishers often use this character to mislead users: for example, if a user encounters the URL http://paypal.com@www.phishpaypal.com, he may be fooled into thinking the site is a legitimate Paypal page. Our F2 feature is also binary, set to 1 if "@" is present in the URL and 0 otherwise.

- *Feature 3(F3): presence of a Secure Sockets Layer (SSL) certificate*

The majority of financial and commercial institutions have an SSL certificate for their websites, which is not typically the case for phishing sites. The binary feature F3 is set to 1 if an SSL certificate is present and 0 otherwise.

- *Feature 4(F4): whether the identity of the webpage conforms to its URL.*

The webpage identity is the most frequent base domain in the page's hyperlinks. For example, the identity of the page http://www.facebook.com/ is "facebook.com"; the majority of the page's hyperlinks should contain the base domain. In general, legitimate web pages have an identity that matches their URL domain despite the existence of links that point to a foreign domain. Phishing pages, on the other hand, behave different. A phishing page that imitates a legitimate page usually retains the same hyperlinks, thus leading to a mismatch between the page's identity and its URL base domain. The binary feature F4 is 1 if the webpage identity matches its URL domain and 0 otherwise.

- *Feature 5(F5): search engine*

The principle of this feature is to apply the algorithm TF-IDF (Term Frequency-Inverse Documents Frequency) on a webpage to extract a document signature (the most relevant words). This feature is used by [16] and [17]. The signature is used as a keyword in a search engine query. If the page's URL is among the first results, the page is defined as legitimate, if not it is classified as a phishing page. We use the same principle but, instead of applying the TF-IDF technique, we use the keywords extraction method used when generating the whitelist (see section III-A). The keywords of the search engine query are composed of the page identity and the four most frequent words among those resulting from the extraction phase. We used the "matacrawler1" search engine which aggregates and returns the relevant results from three search engines (Google, Yahoo and Bing). The binary F5 feature takes the value 1 if the URL of the page is among the top 20 search results and 0 otherwise.

- *Feature 6(F6): Nil anchors*

A nil anchor is an anchor that does not pointanywhere, e.g.: <a href="javascript::void(0)">, <a href="#">. Some phishing pages that imitate a legitimate page replace links to external pages with nil anchors. A high percentage of nil anchors are a likely sign of phishing. The (non-binary) feature F6 is calculated as follows:

$$F6 = \frac{L_N}{L_T} \text{ If } L_T > 0; \quad F6 = 0 \text{ If } L_T = 0. \qquad (2)$$

Where: *LN* is the number of nil anchors and *LT* is the total number of anchors.

- *Feature 7(F7): frequency of links*

Some phishing pages use images instead of html code to imitate a legitimate website's appearance, thus reducing the number of links pointing to other pages. Feature F7 models the frequency of links pointing to pages compared to links pointing to images or scripts and is calculated as follows:

$$F7 = \frac{L_P}{L_T} \text{ If } L_T > 0; \quad F7 = 0 \text{ If } L_T = 0. \qquad (3)$$

Where: *LP* is the number of links to pages and *LT* is the total number of links.

- *Feature 8(F8): action complies with the page identity*

A login page requests access information from users with a form that contains "input" fields, as represented in the following example:

```
<form method="post" action="action.jsp">
<input name="login" id="username" />
<input name="passwd" id="passwd" />
<button type="submit" > Connetion </button>
</form>
```

The information entered in the input fields are processed by the function whose URL is specified in the action field. Usually, phishing web pages claim a legitimate page identity but the action field contains a different URL compared to this identity. The trinary F8 feature models this behaviour as presented in the Algorithm 1.

4. Evaluation

To evaluate our approach, we will test the validity of the whitelist followed by the performance of the SVM classifier.

4.1. The whitelist

To evaluate the performance of the whitelist we used 400 pages, of which 200 were legitimate and 200 were phishing pages. The legitimate pages are composed of the following:

- 10 login pages of top targeted websites [21].
- 50 login pages of the most visited websites according to Alexa2.
- 140 pages from Yahoo 3.

All of the 200 phishing pages were collected from PhishTank4.

As mentioned earlier, the whitelist contains only the information about login pages from the top 10 websites targeted by phishers[21].

Table 1: Evaluation Results of the Whitelist.

Threshold	WebPages (WP)	Phish detected	Legitim detected
≥ 0.7	Legitimate WP (200)	02	10
	Phishing WP (200)	36	0
≥ 0.8	Legitimate WP (200)	01	10
	Phishing WP (200)	34	0
≥ 0.9	Legitimate WP (200)	00	10
	Phishing WP (200)	30	0

To choose an adequate threshold we have tested our whitelist with three threshold values: 0.7, 0.8 and 0.9. Table 1 summarizes the test results. The results show that the higher the threshold similarity is, the lower whitelist wrong decisions are. With a threshold of 0.7 the whitelist detects more phishing pages (36 pages against, 34 and 30 for 0.8 and 0.9 respectively), however there is also an increase in incorrect decisions (2 legitimate pages were classified as phishing pages, versus 1 and 0 for 0.8 and 0.9 respectively). As we seek to avoid any incorrect classifications in the whitelist level, we have adopted a value of 0.9 for our threshold similarity. With this value, the whitelist detected about 5% of all legitimate pages and 15% of phishing pages. Despite this low percentage, we note the absence of incorrect classifications (i.e., the rate of false positives and false negatives is 0%). Moreover, we note that pages with low similarity are not processed at the whitelist level.

Lastly, whitelist effectiveness will increase with growing use as a user's frequented pages are automatically added to the whitelist. This will reduce the risk of a user falling victim to a phishing page imitating one of his or her frequented pages.

4.2. The SVM classifier

If a page has a low similarity with the pages of the whitelist, this page is transformed into a feature vector to be classified. We used an SVM classifier [22], a binary classifier well adapted to our case, as we have only two classes (phish or legitimate).

We used a database of 850 pages, of which 400 are the ones used to evaluate the whitelist, 200 are legitimate pages from Yahoo Random, and 250 are phishing pages collected from PhishTank. We trained our classifier against 400 pages (200 legitimate and 200 phishing) and tested the classifier again the remaining 450 pages (200 legitimate and 250 phishing). Before transforming the testing dataset into feature vectors, we applied the whitelist as a filter. The whitelist filtered 41 phishing pages and 0 legitimate pages. The remaining 409 pages are transformed into feature vectors and passed to the SVM classifier.

We evaluate our classification model according to the percentage of correctly classified phishing pages or true positives rate (TP also called recall), legitimate sites wrongly classified as phishing or false positives rate (FP), precision

(P) that represent the degree to which pages identified as phishing sites are indeed malicious, and F-measure (FM), the harmonic mean between the precision and recall. These various metrics are calculated as follows:

$$TP(R) = \frac{P_P}{P_P + P_L} \quad (4)$$

$$FP = \frac{L_P}{L_P + L_L} \quad (5)$$

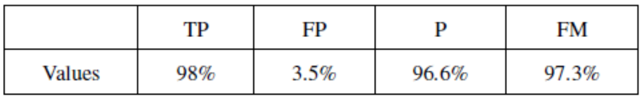
$$P = \frac{P_P}{P_P + L_P} \quad (6)$$

$$FM = 2 \times \frac{P \times R}{P + R} \quad (7)$$

Where *PP*, *PL*, *LP* , *LL* respectively represent: the number of correctly classified phishing web pages , the number of incorrectly classified phishing pages, the number of legitimate pages wrongly classified as phishing sites, and the number of correctly classified legitimate sites.

The following table summarizes our evaluation results on the aforementioned testing dataset.

Table 2: Evaluation results on the testing dataset (SVM classifier performances).



	TP	FP	P	FM
Values	98%	3.5%	96.6%	97.3%

We compared our classification model to those of CANTINA [16] and [17](the approach of M. He et al.).

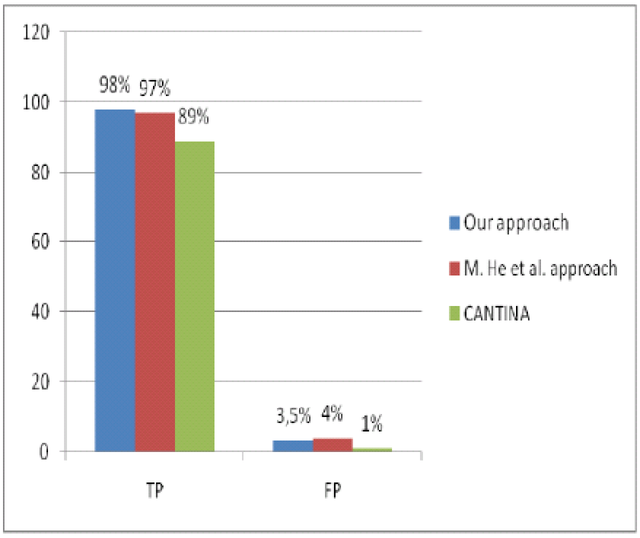
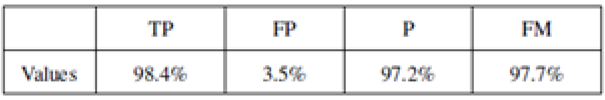


Figure 3: Comparison against previous works

Figure 3 shows a slight improvement of our model in terms of true positive rates: our model can detect 98% of phishing pages against 89% for [16] and 97% for [17]. If we count the 41 pages detected at the whitelist level (since the pages were correctly classified), results will be further improved as illustrated in Table 3.

Table 3: Evaluation results on the testing dataset (svm+whitelist).



	TP	FP	P	FM
Values	98.4%	3.5%	97.2%	97.7%

Our approach suffers from a high false positives rate (>3%); this high value is associated principally with the SVM classifier. As mentioned earlier, the application of the SVM classifier will decrease after the whitelist stabilization, as the whitelist is incrementally augmented by the user's surfing history; at this point, most classification decisions are made at the whitelist level and the risk of a successful phishing attack against the user decreases considerably since any phishing pages that attempt to imitate a page frequented by the user will be detected by the whitelist. If a phishing page imitates a non whitelisted page, it is unlikely that the user would provide the site with sensitive information and, in the unlikely event that the user does provide such information; the page will likely be detected as a phishing attempt by our SVM classifier.

5. Conclusion

In this work we have described an anti-phishing solution that combines a personalized whitelist and an automated classification engine. Our combined approach benefits from the advantages of both techniques without suffering from the drawbacks of each method.

We maintain the accuracy of whitelist solutions and eliminate the difficulty of managing and updating large amounts of data by using a personalized whitelist. False positives traditionally present in these solutions are eliminated since, if a page does not belong to the whitelist, it is not classified as a phishing page but is instead treated by our SVM classifier. Moreover, the proposed whitelist is designed to be automatically updated without user intervention, significantly reducing configuration errors and improving usability.

Despite these advantages, our proposed approach does suffer from some shortcomings. For example, our approach is unable to detect whether legitimate websites are attached by a DNS spoof. We can solve this shortcoming by adding the IP addresses of each page to the whitelist, since the IP address of the majority of targeted websites is often stable [8]. Also the dependence of one feature of the classification model on a search engine can affect the ease of use and responsiveness of the tool in the case of the search engine dysfunction.

Lastly, our approach's performance may be improved, classification features need to be tuned up to work better, while new relevant features may be discovered in the future to further differentiate the legitimate and phishing pages.

References

[1] Phishing Activity Trends Report, 1st Half 2011, http://www.antiphishing.org/

[2] http://www.social-engineer.org/, query date: January 2012.
[3] A. Emigh, "Phishing attacks: Information flow and chokepoints". In M. Jakobsson and S. Myers, editors, Phishing and Countermeasures, pages 31-64. Wiley, 2007.
[4] A. Bergholz, J.H. Chang, G. Paass, F. Reichartz and S. Strobel, "Improved Phishing Detection using Model-Based Features". CEAS 2008.
[5] Mozilla. Phishing protection, http://www.mozilla.com/enUS/firefox/phishingprotection/, query date: March 2011.
[6] google safe browsing : http://code.google.com/intl/fr/apis/safebrowsing/ , query date: March 2011.
[7] http://www.phishtank.com/ query date: March 2011.
[8] Y. Cao, W. Han and Y. Le, "Anti-phishing Based on Automated Individual Whitelist", DIM'08, October 31, 2008, Fairfax, Virginia, USA.
[9] V. P. Reddy, V. Radha and M. Jindal, "Client Side protection from Phishing attack", International journal of advanced engineering sciences and technologies Vol no. 3, Issue no. 1, 039 - 045, 2011.
[10] Y. Wang, R. Agrawal and B. Choi, "Light Weight Anti-Phishing with User Whitelisting in a Web Browser", IEEE Region 5 Conference, April 2008.
[11] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals". Soviet Physics Doklady 10 (1966).
[12] M. Khonji, A. Jonesy, Y. Iraqi, "A Brief Description of 47 Phishing Classification Features", http://khonji.org/upload/feature_desc. Accessed January 2012.
[13] I. Fette, N. Sadeh, A. Tomasic, "Learning to Detect Phishing Emails". Proceeding of International World Wide Web Conference (WWW 2007), Banff, Alberta, Canada, May 2007.
[14] O. Salem, A. Hossain, M. Kamala,"Awareness Program and AI based Tool to Reduce Risk of Phishing Attacks", CIT 2010, Bradford, UK, (2010).
[15] S.Abu-Nimeh, D.Nappa, X.Wang, and S. Nair, "A Comparison of Machine Learning Techniques for Phishing Detection", APWG eCrime Researchers Summit, Pittsburgh,PA, USA. (2007).
[16] Y. Zhang, J. Hong and L. Cranor, "CANTINA: A Content-Based Approach to Detecting Phishing Web Sites". Proceeding of International World Wide Web Conference (WWW 2007), Banff, Alberta, Canada, May 2007.
[17] M. He , S.J. Horng, P. Fan, M. K. Khan, R.S. Run, J.L. Lai, R.J. Chen and A. Sutanto, "An efficient phishing webpage detector", Journal Expert Systems with Applications (12018-12027): Volume 38 Issue 10, September, 2011.
[18] S. Garera, N. Provos, M. Chew, and A. D. Rubin, "A framework for detection and measurement of phishing attacks". In Proceedings of the WORM. (2007).
[19] J.Ma, L.K. Saul, S.Savage, GM. Voelker, "Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs", KDD 09.Paris, France (2009).
[20] G. Salton, "Mathematics and information retrieval". Journal of Documentation, 35(1),1–29, (1979).
[21] OpenDNS 2010 Report: Web Content Filtering and Phishing. http://www.opendns.com/pdf/opendns-report-2010.pdf.
[22] C. C. Chang and C.-J. Lin, "LIBSVM:a library for support vector machines". ACM Transactions on Intelligent Systems and Technology, 1–27 (2011).

Author Profile

Komatla. Sasikala Obtained the M.sc degree in Computer Science from Govt. College for Women, Guntur. At present persuing the M.Tech in Computer Science and Engineering (CSE) Department at Guntur Engineering College, Guntur.

P. Anitha Rani obtained the B.Tech Degree from Sri C.R. Reddy Engineering College and M. Tech (CSE) from JNTU, Hyderabad. She has 10 years of teaching experience and working in Computer Science and Engineering (CSE) Department at Guntur Engineering College, Guntur.