RUNTIME DETECTION OF PHISING ATTACK USING SYNTACTIC VERIFICATION THROUGH WEB SERVICES

V. Shanmuganeethi NITTTR

Chennai India

ABSTRACT: *Providing a secure service in web applications is a growing concern and real challenge in web security. Among the various types of web application attacks, phishing is the most common type of attack. It often direct the users to enter details at a fake website whose look and feel are almost identical to the legitimate site. Present tools are cannot completely detect the phishing attacks, that leverage vulnerabilities in trusted web applications. This paper attributes to identify phishing web sites by analyzing and validating the Uniform Resource Locator (URL), Hyperlink in web pages and syntactic verification of Hyperlink. As URLs are following the common standard RFC 1738, we have developed a schema for converting the URL into XML for verifying the URL. The detection of Phishing web sites implemented by means of two layered web services. Our web services are an independent layered module in a web application and detect and prevent the phishing attacks.*

KEYWORDS: Phishing, Web Service, URL, XML Schema, Input Validation, Hyperlink, IP Address, Scammer

INTRODUCTION

Most of the web application attacks are initiated from the software vulnerabilities and flaw in the design of the software solutions. Vulnerabilities in software allow attackers to steal the personal information of another person or another system. Web application vulnerability control mechanism is one of the most important parts of the web application security. There are some standard strategies to protect such vulnerabilities like protect the client's personal information from the attackers, ensure the credibility of the website to the client and by providing a secured transaction or communication via the web applications. But, these strategies are very general in nature and they are preventing only well known web application threats.

Phishing Attack

Phishing attack is a way of attempting to steal sensitive information by masquerading as a trustworthy entity in an electronic communication [1]. It is typically carried out by the URL, hyperlinks and e-mails. Phishers are using different techniques to get the victims' personal information, but most of the time the final purpose is to steal money from their bank account. Indeed, most of the recent attempts were targeting the customers of banks and the users of online payment. Very often, phishers send a mail to bank clients or online payment users to inform them of a technical problem with their account or with their data. This mail seemed to be sent by legitimate online organizations, or ISPs. In these e-mails, attackers will make-up some causes, e.g. the password of the credit card had been wrongly entered for many times or to providing upgrading services, to allure the user to visit attacker's web application to conform or modify your account number and password through the hyperlink provided in the e-mail. User will then be linked to a counterfeited web application after clicking those links.

According to the Anti-phishing Working Group(APWG 2014) [1], approximately 125,215 unique phishing websites were reported. The recent report from APWG is shown in figure 1.

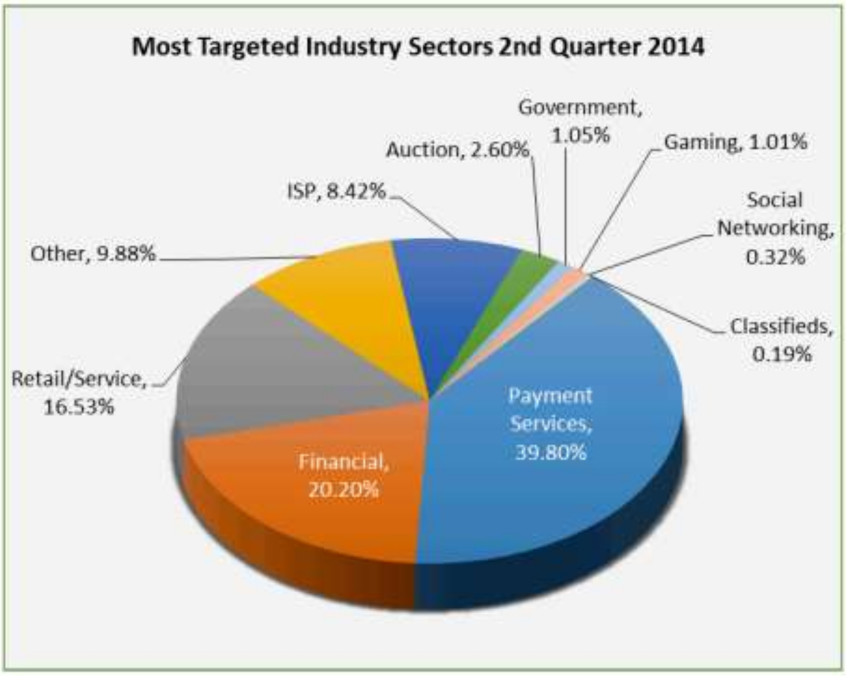


Figure 1. Classified Ads Sector Breaks Out in Q2 as Rapidly Expanding Phishing Vector

With the phenomenal development of social website such as MySpace or Facebook very popular among students all over the world, phishers tend to orient their attempts towards social networking. They know that a lot of personal data can be stolen. Social network phishing attempts get a success of more than 70%, as it was shown by experiments. Indeed, young people feel that they are among friends and that they can trust everybody.

Main variant of phishing attacks

Over the years, many different types of phishing attacks have emerged, and continue in common use today. In order to be on the alert for these scams, it is important to be aware of the several common strategies used as part of these attacks. The main variants of phishing attacks are [2] deceptive phishing, Malware-Based Phishing, Keyloggers and Screenloggers, Session Hijacking, Web Trojans, Hosts File Poisoning, System Reconfiguration Attacks, Data Theft, DNS-Based Phishing ("Pharming"), Content-Injection Phishing, Man-in-the-Middle Phishing and Search Engine Phishing

LITERATURE SURVEY

Many works have been done in the area of phishing related attacks in web applications. Anti Phishing Working Groups (APWG) [1] describes the concepts of phishing attack, explores the attack vectors, and cites examples of preventative best practice in Web applications. Lance James, [2] describes the nature of phishing attacks, various frequencies in which they occur and methods of detecting and correcting them. Mitesh Bargadiya et al [3] discuss a method to avoid the phishing attack during the web transaction by the technique of mutual authentication. Guang Xiang [4] describes the detection method for phishing by using key word retrieval mechanism. Lorrie Cranor et al [5] describe a technique to detect the phishing attack using the tool CANTINA. This tool having two type of features such as URL based features and HTML based features. URL base features will count the number of dots in the url, number of dashes in the url, whether the domain name is hot coded or not and check any sensitive word in the URL. The HTML based feature find the sensitive keywords in the web page, check the action atrribute value in the form tag and compare the most frequent keyword in the html link and the web site brand name. John Yearwood [6] describes to identify the phishing emails by analysis the hyperlink along with Domain Name Server (DNS). Yingjie Fuet al [7] describes to identify the phishing web sites using the visual similarity of the web sites. Brad Wardman et al[8] describe to identify the phishing web sites by analyzing the common sub string in the phishing URLs. Juan Chen and Chuanxiong Guo [9] describe the prevention mechanism by analyzing the hyperlink of the web site. Maher Aburrous, et al [10] describe to identify the phishing web site by using fuzzy data mining in order to secure the applications from phishing attacks and protect the client's sensitive information from the attackers.

Our approach circumvents the problems existing in the present prevention tools through a methodology independent of phishing signatures and specific implementations, and thus it is able to handle new phishing variants quickly. Moreover, our approach implemented by means of a layered web services which is independent to web application and it doesn't demand the change in web application when our service is deployed.

Proposed System Architecture

Our proposed system architecture is as shown in figure 2. It consisting of layered modules to detect phishing URL. The modules are URL validator, Black list verifier, Source code verifier with Black list and hyper link validator. Every requested web site is passed to entire layer for checking its legitimacy. If the requested page found as a phishing site in a module, then no need to pass to the remaining modules. For syntactic verification, a XML file will be generated from the suspected phishing URL or hyperlink presented in the invoking web page. This XML file will be validated against to a XML schema which we have proposed in our methodology to verify the genuineness of the URL.

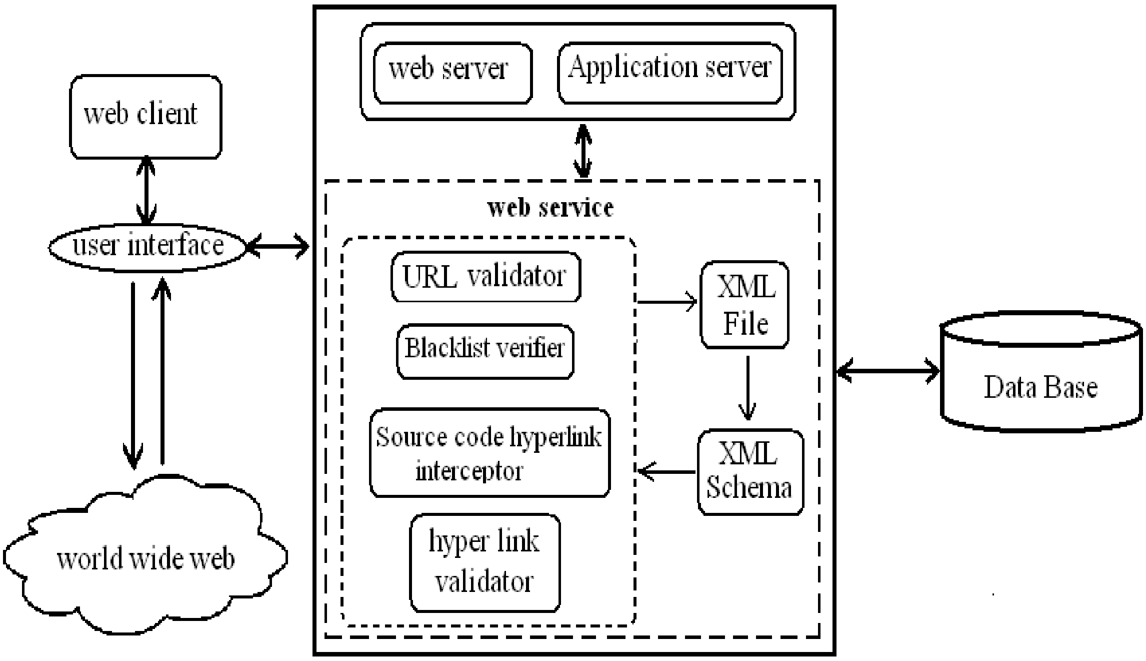


Figure 2. Proposed System Architecture Diagram

When a user requested a web page, the URL of the web page would be sent to our web service as input to validate the URL. Then, the URL passed for checking its legitimacy and finally based on the result (true / false), the user will be directed to requested page or warning as phishing page. The XML file is a generated file from the input URL and XML schema is our own schema representing the URL format. The Database consists of list of phishing URLs as blacklist.

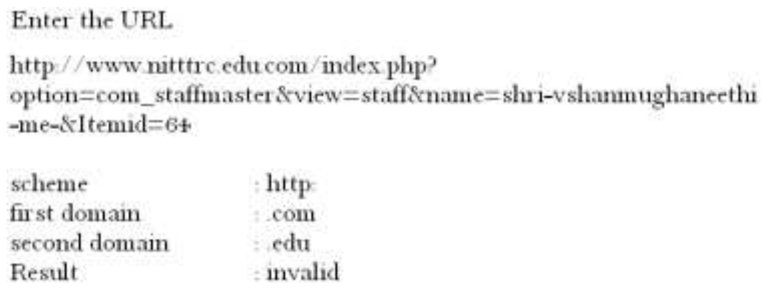
The URL Validator and Black list verifier

The URL format is based on UNIX file path syntax, where forward slashes are used to separate directory or folder and file or resource names. For example The syntax is *scheme://domain:port/path?query_string#fragment_id*

The scheme name defines the namespace, purpose, and the syntax of the remaining part of the URL. For example, a web browser will usually dereference the URL http://example.org:80 by performing an HTTP request to the host at example.org, using port number 80. The URL mailto:shanneethi@nitttrc.ac.in may start an e-mail composer with the address shanneethi@nitttrc.ac.in in the *To* field.

In a URL validator, when a user requests a URL, the requested URL is intercepted and tokenized for structure verification. The tokenized list is classified with a scheme, domain name with port number, file name with path and optional fragment identifiers. According to the URL standard, there should be an only one scheme, domain name with port number and path in the URL structure. If the tokenized list is consisting of more than one standard parts of the URL, then the request will be not to the legitimate website. Hence, If the structure of the URL is not followed the standard structure, this module blocks the HTTP or other scheme request to phishing site and sends a warning message back to user's page. For example, *http://www.nitttrc.edu.com/index.php?option=com_staffmaster&view=staff&name=shri–v shanmugha neethi me-&Itemid=64*

In the above URL is tokenized and analyzed by the URL validator is shown in figure 3. It results, there is more than one domain namely .com and .edu. Due to two Top level domain names are presented which would be resulted as invalid link.



```
Enter the URL

http://www.nitttrc.edu.com/index.php?
option=com_staffmaster&view=staff&name=shri-vshanmughaneethi
-me-&Itemid=64

scheme              : http
first domain        : .com
second domain       : .edu
Result              : invalid
```

Figure 3. Query tokenizer

Hence, this type of URL would be prevented to load in a web browser. But http://www.paypai.com is a URL which is in the standard structure but it is not a legitimate URL. This type of phishing sites would be prevented by a blacklist URLs. It is a commercial managed service maintained considerably larger than most unmaintained blacklists [11]. It consist bulk entries of phishing URL with add and remove option. These add and remove options are used to update phishing sites and remove accidentally added sites in the phishing entries. This commercial provider didn’t give any warranty or guarantee of the black list service. But these entries are key component for most of the phishing verification tools. These entries are very much helpful to indentify the phishing web sites in World Wide Web. We obtained those bulk entries and categorized based on the domain name to easy verification of the phishing site shown in figure 4. This category based structure results faster response in real time verification of phishing URL. Our module obtained the user requested URL and matched against with the black list URLs based on the domain name. If match not found, then this URL will be sent to next module for syntactic verification. Because this entries are not provide 100% support to prevent phishing sites.

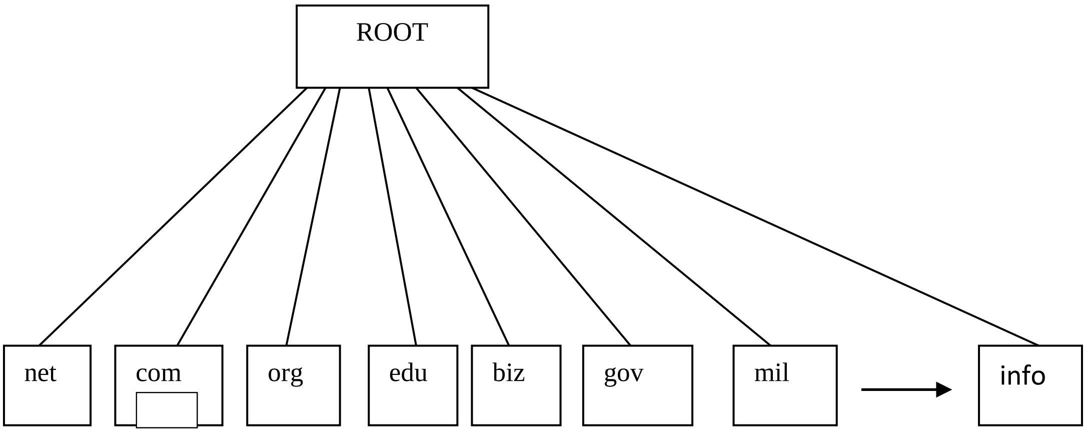
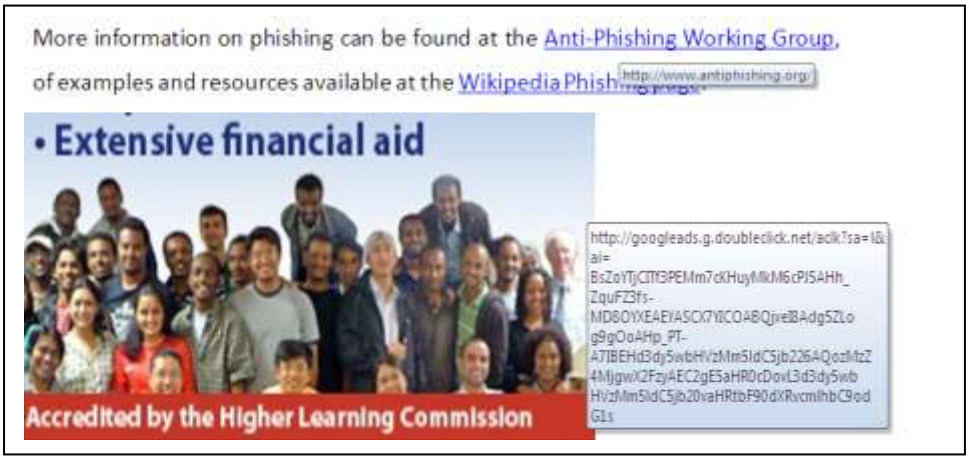


Figure 4. Sample TLD category

Otherwise, the requested URL rejected from this module and customized warning message is returned to the client.

Source code hyperlink interceptor

As one of phishing variant, phishing hyperlinks could be embed by the phisher into the source code. These hyperlinks will be loaded immediately in the browser, when the user clicks the particular link. These types of links are not prevented from URL validator and Black list verifier as it a hyperlink in the source code. Such links must be prevented from the client to secure the client information. To detect these hyperlinks, a source code hyperlink interceptor is integrated in our approach.



More information on phishing can be found at the Anti-Phishing Working Group,

of examples and resources available at the Wikipedia Phish

http://www.antiphishing.org/

http://googleads.g.doubleclick.net/aclk?sa=l&ai=BsZoYTjCITf3PEMm7cKHuyMkM6cPJ5AHh_ZquFZ3fs-MD8OYXEAEYASCX7YICOABQjveBAdg5ZLog9gOoAHp_PT-A7IBEHd3dy5wbHVzMm5ldC5jb226AQozMzZ4MjgwX2FzyAEC2gE5aHR0cDovL3d3dy5wbHVzMm5ldC5jb20vaHRtbF90dXRvcmlhbC9odG1s

• Extensive financial aid
Accredited by the Higher Learning Commission

Figure 5. Hyperlink interception

Reference with the URL, the entire source code of the web page will be read and intercept all the hyperlinks presented in the particular page. The hyperlinks may be in the form of anchored text as well as anchored image as shown in figure 6, which are used to redirect the client to the other web servers. The intercepted URLs are analyzed as, if any intercepted URLs is belongs and pointing to the same web site, then those links are ignored. Such links cannot be phishing link because all are towards to legitimate current web site. All other links are newly explored links from the current web page, among those links any one of its hyperlink may be phishing link which can be injected by the phishers while communication. Hence, there may be a possibility of phishing link in the particular web site. So, the explored links again verified by the blacklist verifier to validate its legitimacy and protect further loading of web page if phishing link was found.

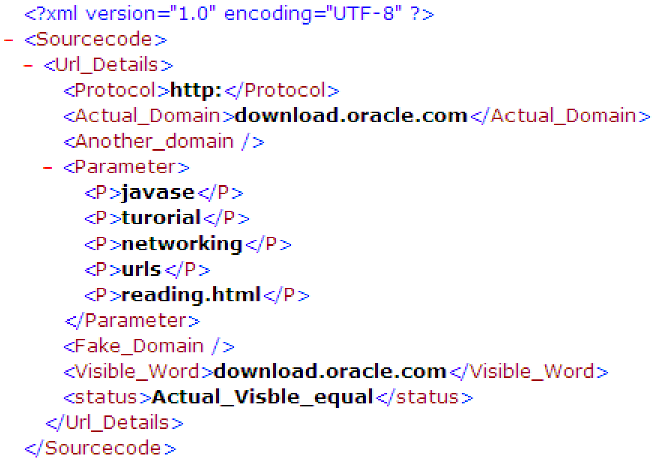
Hyper Link Validator

This module is a new approach to detect the phishing URL and hyperlink which are not prevented from the URL validator and Source code hyperlink interceptor due to its standard structure and not placed in the Black list. This module collected the suspected URL from the source code hyperlink interceptor. The suspected link would be parsed and analyzed based on the scheme, domain name, path and fragment identifier to create a XML file since all the links are standard in structure. For example A legitimate code in a web page is *<a href="http://download.oracle.com/javase/tutorial/networking/urls/reading.html"> download.oracle.com/javase/tutorial/networking/urls/reading.html</a>*

In this code the visible link and actual links are exactly same. So, when a user clicks the visible link the user redirected to the intended web page

<a href=“http://download.oracle.com/javase/tutorial/networking/urls/reading.html”> .

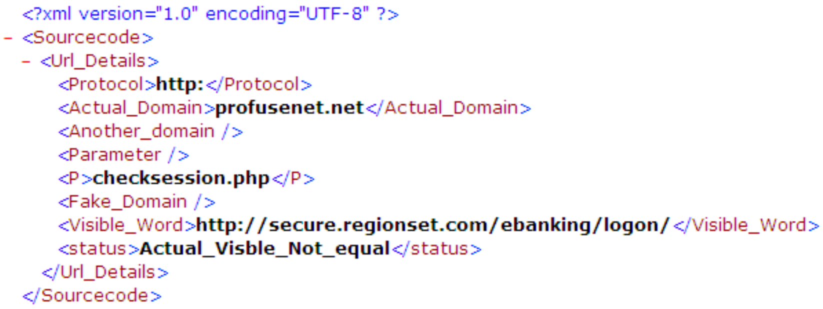
When this link is parsed by this module, the XML file would be generated which is shown in figure 6.



```
  <?xml version="1.0" encoding="UTF-8" ?>
- <Sourcecode>
  - <Url_Details>
      <Protocol>http:</Protocol>
      <Actual_Domain>download.oracle.com</Actual_Domain>
      <Another_domain />
    - <Parameter>
        <P>javase</P>
        <P>turorial</P>
        <P>networking</P>
        <P>urls</P>
        <P>reading.html</P>
      </Parameter>
      <Fake_Domain />
      <Visible_Word>download.oracle.com</Visible_Word>
      <status>Actual_Visble_equal</status>
    </Url_Details>
  </Sourcecode>
```

Figure 6. XML structure for legitimate link

The XML file is standard in structure, which describes the scheme called protocol, actual domain with visible link and corresponding path and parameter for actual domain. The suspected hyper links are categorized under three types, which are left out from the previous modules. The first type of suspected phishing link has the difference in visible link and actual link location. The second type, there is more than one domain in the actual link but in a visible link looks a genuine word. This cannot be protected through URL validator since, this link presented in the source code not in the address bar and the last suspected category links consists fake domain in actual hyperlink and genuine word in the visible web page. First type, a suspected link will be <a href ="http://www.profusenet.net/checksession.php"> https://secure.regionset.com/EBanking/logon/ </a> the actual link and visible link locations are not same. This type of link is highly venomous to lead phishing site. For the above suspected link the XML file is shown in figure 7.

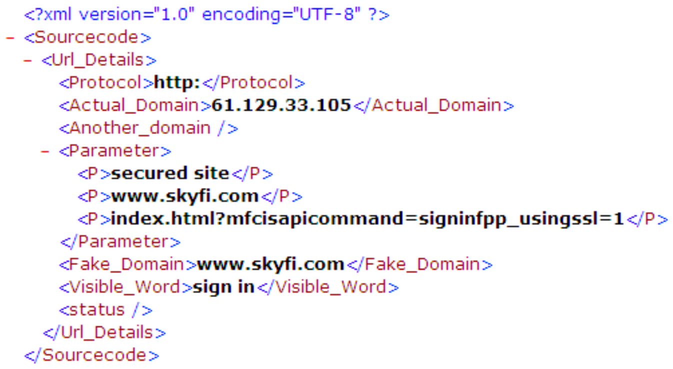


```
  <?xml version="1.0" encoding="UTF-8" ?>
- <Sourcecode>
  - <Url_Details>
      <Protocol>http:</Protocol>
      <Actual_Domain>profusenet.net</Actual_Domain>
      <Another_domain />
      <Parameter />
      <P>checksession.php</P>
      <Fake_Domain />
      <Visible_Word>http://secure.regionset.com/ebanking/logon/</Visible_Word>
      <status>Actual_Visble_Not_equal</status>
    </Url_Details>
  </Sourcecode>
```

Figure 7. XML structure for phishing link – Type 1

In this XML structure, the exact difference is clearly identified with the tag value in <Actual_Domain> and <Visible_Word>. Second type, a suspected link will be *<a href="http://61.129.33.105/secured site/www.skyfi.com/index.html?MfcISAPICommand=SignInFPP&UsingSSL=1"> SIGN IN </a>*

Here, the actual link designed with more than one domain. For this type of phishing link, the XML file is shown in figure 8.

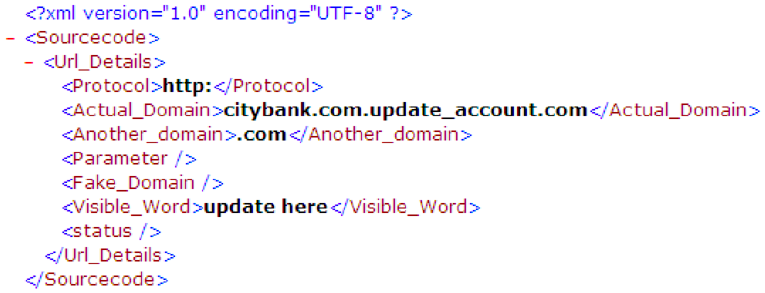


```
  <?xml version="1.0" encoding="UTF-8" ?>
- <Sourcecode>
  - <Url_Details>
      <Protocol>http:</Protocol>
      <Actual_Domain>61.129.33.105</Actual_Domain>
      <Another_domain />
    - <Parameter>
        <P>secured site</P>
        <P>www.skyfi.com</P>
        <P>index.html?mfcisapicommand=signinfpp_usingssl=1</P>
      </Parameter>
      <Fake_Domain>www.skyfi.com</Fake_Domain>
      <Visible_Word>sign in</Visible_Word>
      <status />
    </Url_Details>
  </Sourcecode>
```

Figure 8. XML structure for phishing link – Type 2

In this XML structure, tag value in <Actual_Domain> and <Visible_Word> are differed. Third type, the suspected hyperlink will be *<a href="http://citybank.com.update_account.com">update here </a>*

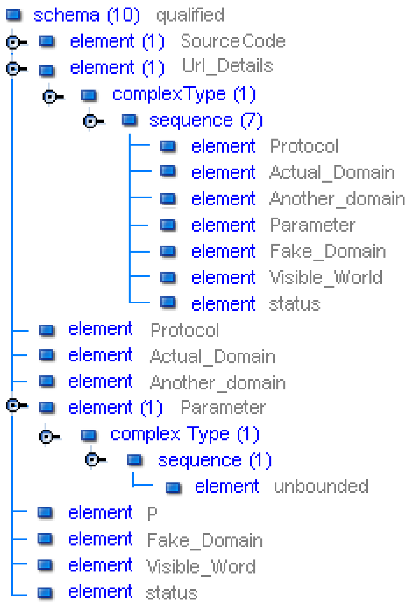
This link leads to a web site, but that site is not being a legitimate site. In this link there are two domains like citybank.com and it is appended with update_account.com. The XML file for the above link is shown in figure 9.



```
  <?xml version="1.0" encoding="UTF-8" ?>
- <Sourcecode>
  - <Url_Details>
      <Protocol>http:</Protocol>
      <Actual_Domain>citybank.com.update_account.com</Actual_Domain>
      <Another_domain>.com</Another_domain>
      <Parameter />
      <Fake_Domain />
      <Visible_Word>update here</Visible_Word>
      <status />
    </Url_Details>
  </Sourcecode>
```

Figure 9. XML structure for phishing link – Type 3

In this XML file, <Actual_Domain> tag value consists of more than one TLD value. So, all the types of phishing attack could be filtered by our approach by validating the generated XML file with our proposed XML Schema shown in figure 10. An XML schema is a description of a type of XML document, typically expressed in terms of constraints on the structure and content of documents of that type, above and beyond the basic syntactical constraints imposed by XML itself. These constraints are generally expressed using some combination of grammatical rules governing the order of elements, Boolean predicates, data types of elements and attributes, and more specialized rules such as uniqueness and referential integrity constraints. We have proposed a very general standard XML schema which satisfies all genuine hyperlinks which it is in the form of XML file.



```
schema (10)  qualified
  element (1)  SourceCode
  element (1)  Url_Details
    complexType (1)
      sequence (7)
        element  Protocol
        element  Actual_Domain
        element  Another_domain
        element  Parameter
        element  Fake_Domain
        element  Visible_World
        element  status
  element  Protocol
  element  Actual_Domain
  element  Another_domain
  element (1)  Parameter
    complex Type (1)
      sequence (1)
        element  unbounded
  element  P
  element  Fake_Domain
  element  Visible_Word
  element  status
```

Figure 10. Tree structure of XML schema

To prevent a phishing link, the hyperlink would be converted as XML file with the defined tag names and the values were filled for the tag. The generated XML file validated against with the XML schema. If the XML file parsed successfully, then the hyperlink is not suspicious. Otherwise, the hyperlink would be leads to a phishing site.

RESULTS AND DISCUSSIONS

To evaluate the proposed approach, we have taken sample URL entries which consists genuine and leads to phishing site. The sample entries were tested with DyPhishDec versus without DyPhishDec and the response time measured. The same entries were tested with all the layers in step-by-step process. We analyze the performance of the web application based on the response time in each module. Each response time evaluated independently and the response time and the way of response (allowed/not allowed) are tabulated. The following tables are shows the response time results of our approach.

In the table 1 the response time is very less but it allows client to visit the phishing web sites. Hence, we included our modules one by one for evaluating our proposed approach to prevent phishing web sites. The table 2 shows the data by inclusion of URL validator to check the phishing sites.

Table -1 Response time and phishing Detection

URL	Time taken to response (ms)	URL type	Status
www.gmail.com	855	Legitimate URL	Allowed
87.193.226.99	922	Phishing IP in hyper link	Allowed
www.paypai.com	795	Phishing URL	Allowed
www.convert.money.net	871	Having phishing hyper link	Allowed
www.google.com.net	958	Phishing URL	Allowed

Table -2 Response time and phishing Detection with URL Validator

URL	Time taken to response (ms)	URL type	Status
www.gmail.com	879	Legitimate URL	Allowed
87.193.226.99	925	Phishing IP in hyper link	Allowed
www.paypai.com	800	Phishing URL	Allowed
www.convert.money.net	879	Having phishing hyper link	Allowed
www.google.com.net	529	Phishing URL	**Not Allowed**

In table 2 the response time is just increased compare to the previous table i.e.) without our proposed approach in terms of milliseconds but it will protect the basic errors in the URLs. But some of the sites which are in the category of phishing sites are allowed only with the URL validator. By filtering with commercial blacklist along URL validator results are shown in table 3.

Table -3 Response time and Phishing Detection with URL Validator and Black List verification

URL	Time taken to response (ms)	URL type	Status
www.gmail.com	988	Legitimate URL	Allowed
87.193.226.99	999	Phishing IP in hyper link	Allowed
www.paypai.com	923	Phishing URL	**Not Allowed**
www.convert.money.net	1012	Having phishing hyper link	Allowed
www.google.com.net	569	Phishing URL	**Not Allowed**

In table 3 the response time is more than compare to only URL validator along with Blacklist Filter, but the site paypai.com is a phishing site that is prevented to load in the client browser. The paypai.com is exactly following the standard structure of the URL but that is placed in the blacklist. But, the sites other than gmail.com are leads to phishing. This module doesn't prevent those URL with commercial blacklist along URL validator. Hence, the next layered module hyperlink validator is included along the previous modules. The response time of entire approach is shown in table 4.

Table- 4 Response time and Phishing Detection with URL Validator, Black List verification and Hyperlink validator

URL	Time taken to response (ms)	URL type	Status
www.gmail.com	1010	Legitimate	Allowed
www.google.com.net	1032	Phishing URL	**Not Allowed**
www.paypai.com	1025	Phishing URL	**Not Allowed**
www.convert.money.net	1055	Having Phishing hyperlink	**Not Allowed**
150.101.116.140	590	Phishing IP in hyperlink	**Not Allowed**

In table 4, all the URL which leads to phishing are detected and prevented to load in the client browser. The time difference with our approach and without our approach is negligible when comparing the consequences of phishing. Hence, our approach detects all the phishing sites which are in different forms and category. Moreover, our approach is completely free from false negatives.

CONCLUSION AND FUTURE WORK

In World Wide Web huge number of web sites has vulnerabilities, which can be hacked or attacked by phishing attack technique. Phishers can create a fake web site which is similar to the legitimate and forcibly persuade the client to visit that fake websites and ask them to enter the sensational information in order to steal the client's personal information. In this paper, we have proposed web service approach for run time detecting and preventing the phishing attack by validating the url and hyper link of the web sites. Comparing to the existing work the web service approach is platform independent and there is no load in the client side like a separate tool. We analyze the web application with the developed web service and found that the response time of the web application result set and error set. For future work, we intend to analyze only the URL, which is given as input to the web browser by a user. The independent analyze of the URL and hyper links gives the greater performance to protect phishing. If URL, hyperlink and the content of the emails is properly analyzed, we could protect phishing better way.

REFERENCES

[1] The Anti-Phishing Working Group (APWG), http://www.antiphishing.org/

[2] Lance James, Secure Science Corporation "Phishing Exposed - Uncover secrets from the Dark side" Syngress Publishing, Inc. ISBN: 159749030X, 2005

[3] Mitesh Bargadiya, Vijay Chaudhari, Mohd. Ilyas khan, Bhupendra Verma, "Anti-Phishing Design Using Mutual Authentication Approach". International journal of computer applications and Information Technologies Volume 1 Issue 3 pp. 175-178, July 2010

[4] Guang Xiang, "A Hybrid Phish Detection Approach by Identity Discovery and Keywords Retrieval" World Wide Web Conference 2009, Madrid, Spain, April 20–24, 2009

[5] Guang Xiang, Jason Hong, Lorrie Cranor, Carolyn P. Rose, Cantina: A Feature-rich Machine Learning Framework for Detecting Phishing Web Sites.

[6] John Yearwood, Musa Mammadov, Arunava Banerjee "Profiling Phishing Emails Based on Hyperlink Information", In 2010 International Conference on Advances in Social Networks Analysis and Mining, pp. 120-127,August 2010 ,

[7] Yingjie Fu, Liu Wenyin, Xiaotie Deng "EMD based Visual Similarity for Detection of Phishing Webpages". IEEE Transactions on Dependable and Secure Computing, Vol. 3 pp 301 - 311 Oct -Dec .2006

[8] Wardman, Gaurang Shukla, and Gary Warner, "Identifying Vulnerable Websites by Analysis of Common Strings in Phishing URLs", Conference on eCrime Researchers Summit, pp 1 – 13, 2009.

[9] Juan Chen and Chuanxiong Guo, "Online Detection and Prevention of Phishing Attacks" International conference on Communications and Networking in China, PP. 1-7, 2006

[10] Maher Ragheb Aburrous, Alamgir Hossain, Keshav Dahal, Fadi Thabatah "Intelligent phishing detection system for e-banking using fuzzy data mining", Journal of Expert Systems with Application, Volume 37 Issue 12, December, 2010

[11] "URL Black List", http://urlblacklist.com/?sec=download

[12] "Phishing Site Checking submission – phishtank"http://www.phishtank.com/ phish_search.php? page=1 & valid=y&Search=Search

[13] APWG, Issues in Using DNS Who is Data for Phishing Site Take Down, journal May 2007.

[14] Laurie A. Werner Jill Courte, “Analysis of an Anti-Phishing Lab Activity”, Journal of Information science and Education, Vol.8, Num11, pp 3 – 8, April 2010

[15] Yue Zhang, Serge Egelman, Lorrie Cranor, and Jason Hong, Carnegie Mellon University, “Phinding Phish: Evaluating Anti-Phishing Tools” Proceedings of the 14th Annual Network and Distributed System Security Symposium,2007

[16] Ye Cao, Weili Han and Yueran Le, Ye Cao Software School, Fudan University 825#, zhangheng Road Shanghai, P. R. China, Anti-phishing Based on Automated Individual White-List, IEEE conference 2008.

[17] Holz, T.Marechal, S.Raynal, F. Mannheim Univ. “New Threats and Attacks on the World Wide Web, Security & Privacy, IEEE, pp 72-75,2006,

[18] Lorrie Faith Cranor, Jason Hong, “An Empirical Analysis of Phishing Blacklists”, CEAS 2009 Sixth Conference on Email and AntiSpam ,Mountain View, California USA, July 2009