VBEx: A Browser Plug-In To Prevent DNS Based Phishing Attacks

Swapan Purkait*
Dr. S.K. De**

Abstract

Phishing has grown significantly in last decades or so, becoming the most common web threat today. To mitigate the widespread problem of phishing on the Internet, almost all modern day web browsers provide security indicators. Detecting and identifying any phishing websites in real-time, particularly for e-banking, is really a complex and dynamic problem involving many factors and criteria. This work is the extension of our earlier work on Virtual Browser (VB) (Purkait, 2012). In this study we propose Virtual Browser Extension (VBEx) a web browser plug-in, which checks for the authoritative name server and verifies the same for a visiting website. In addition to that it maintains a white list of websites that engage in online monetary transactions so that when a user requires accessing any of these, the default protocol should always be HTTPS, failing which it will prevent the web browser to load the requested website. We conducted series of lab test on the VBEx prototype and found that the VBEx is a promising approach.

Keywords: Phishing; Virtual Browser extension; Phishing countermeasures; Security indicators; Phishing toolbars

Introduction

Phishing has become the most common channel for thieves for acquiring personal information to aid them in identity theft (Brody et al., 2007; Anderson et al., 2008). Studies show a steady increase in phishing activities as well as its related costs. APWG in their annual report published in October 2012, reported 93,462 Phishing attacks from the period of January to June 2012 (APWG, 2012). PhishTank, the online website which collects data on websites engaged in phishing activities received 22,581 valid submissions of phishing websites solely for the month of July 2012 (PhishTank, 2012).

There are many techniques for phishing ranging from, code-based key-loggers (Gorling, 2007), DNS (Domain Name System) poisoning, search engine phishing to mass emailing (Forte, 2009). Most phishing attacks trick users into submitting their personal information using web forms. Even though using a web form for extracting sensitive information from users has been known for its vulnerability to phishing attacks yet most genuine sites continue this practice and billions of dollars are lost each year due to unsuspecting users entering personal information into fraudulent websites. As concluded by Parno et al. (2006) phishing is a significant and growing problem which threatens to impose increasing monetary losses on businesses and to shatter consumer confidence in e-commerce. As highlighted by Chou et

**Research Scholar, Vinod Gupta School of Management, Indian Institute of Technology, Kharagpur, India.*
***Professor, Vinod Gupta School of Management, Indian Institute of Technology, Kharagpur, India.*

Authors are grateful to the anonymous referees for comments and suggestions for improving the text and contents of the paper. Authors alone are responsible for any errors/mistakes still remaining in the paper.

al. (2004) criminals will become more active and their attacks will become more sophisticated, making user-based protection mechanisms fragile given the user population of non-experts.

In this work we focus on phishing attack which uses Internet vulnerabilities based on DNS server and DNS resolver to redirect users to an imitation of the original site. In this attack, the DNS infrastructure is compromised so that DNS queries for the victim's requested domain address return an attacker-controlled IP address. These kinds of attacks are more dangerous since a user may be unknowingly taken to a malicious website even if he types the correct web address. As mentioned by Karlof et al., (2007) these attacks are particularly devious because the browser's URL bar will display the domain name of the legitimate site, potentially fooling even the most meticulous users.

How DNS Works

The Internet is made of two principal name spaces: Internet Protocol addressing system (IP Address) and Domain Name System (DNS). A DNS server stores the DNS records which are nothing but a mapping of domain names with their corresponding IP addresses. When asked, these DNS servers resolve human readable domain names to IP addresses that are assigned to computer servers on the Internet. When a new domain is registered or booked with a domain name registrar, the domain administrator for the said domain provides a list of name servers that are authoritative for the zone corresponding to that domain. The registrar in turn conveys these server names to the domain registry for the TLD that is authoritative for the corresponding zone, e.g. .in for nettech.in. The domain registry updates its authoritative name servers to include the new domain information. VeriSign Inc., the trusted provider of Internet infrastructure services, reported in their October 2012 Domain Name Industry brief, a total of 240 million domain name registrations across all Top Level Domains (TLDs) in the world (Verisign, 2012).

Normally, when an Internet user wants to visit a website and types the web address in the address bar of the web browser, for example www.nettech.in, a local DNS stub resolver looks into the memory of the local machine to see if it has recently found and cached an IP address for the domain name in question. If it is not able to resolve the same locally it sends a DNS query to a Local DNS server or a recursive name server. A recursive name server is typically maintained by Internet Service Providers (ISPs) or by the local network administrator for a Local Area Network (LAN).

A recursive name server or a local DNS server performs domain name lookups on behalf of the client computers. The recursive name server maintains a local cache as well. If it has looked up that exact domain name before, it will find the answer in its cache and respond at once. If not, it will send its own DNS queries to one or more of the authoritative name servers responsible for the requested domain name and fetch the appropriate address record requested by the client computer. Thus, recursive name servers resolve any query they receive, by querying the authoritative name servers for the requested domain and passes the same to the client computer. In this way, the computer knows which server to contact when user types the domain name in the address bar of the web browser. If a user mistypes the domain, e.g. www.netttech.in instead of the original domain www.nettech.in of Nettech Private Limited, the DNS server fails to resolve the domain and the user gets an error message.

Most Internet users use the DNS servers of their LAN. Commonly a local DHCP (Dynamic Host Configuration Protocol) server provides the IP address of the DNS server to all computers who request for automatic IP settings from DHCP. A rogue DHCP server on local area network can provide an address of a rogue DNS server to these computers (Purkait, 2013). These DNS servers are set up by malicious third parties to translate certain domains to fallacious IP addresses. As a result, victims are redirected to possibly malicious websites without them noticing it. For example, if a user wants to view www.nettech.in, a rogue DNS server may resolve www.nettech.in to an IP address controlled by an unknown third party. If that third party creates pages that look exactly like those of Nettech, the user might think that he is browsing Nettech indeed, without noticing that he is actually visiting a website controlled by somebody other than Nettech. This may cause the user to give away sensitive information to third parties.

Phishing Techniques Based on Domain Name

Phishers apply a wide range of techniques to allure innocent Internet users to visit fake web pages and provide their personal information. A detailed discussion on these techniques can be found in Butler (2007) and Purkait (2012a). For this study, we look at the following phishing techniques based on domain name, detailed discussion on DNS based phishing attack can be found in Gastellier-Prevost et al, (2011a, 2011b).

a) **Completely different domain name.** Phisher relies on the end users ignorance or lack of awareness of the domain name of the enterprise they want to visit. Phisher can host the spoofed website in any random website of their choice and accept that the victim will not be looking at the URL bar completely. For example a spoofed onlinesbi.com can be hosted in abc.com.

b) **Sub domain or child domain name.** Phishing URLs contain some parts of original website URLs. For example, the legitimate website of Nettech Private Limited is www.nettech.in. A phisher uses URL name which contains a part of the legitimate URL such as www.nettech.in. demo.com.

c) **Misspelt domain name.** It is very easy to tamper with one or more letters of a legitimate URL and go unnoticed by a casual Internet user. For example, www.paypai.com can be used to spoof www.paypal.com. Or for example sbionline.com instead of the original domain name onlinesbi.com

d) **Different TLD used.** Phishers use a different TLD to host the spoofed site. For example a spoofed site of onlinesbi.com is hosted in onlinesbi.co.in. User who is not aware of the original domain name can easily be fooled with this technique.

e) **Same domain name.** Phisher hosts the spoofed website with the same domain name as of the original one and changes the host table of the victim's computer to point to the

spoofed domain's IP address. It might also direct client computer's DNS queries to a rogue recursive DNS server which sends wrong address record to the victim's computer. In this case though the user has typed the correct domain name but due to the rouge DNS server the user will be directed to the phishing website.

In this paper we present Virtual Browser Extension (VBEx) a web browser plug-in to deter the growing phishing attacks. This work is extension of our previous work on Virtual Browser (Purkait, 2012). Virtual Browser is an on-demand browsing service over internet where Internet user can remotely connect to a secure server and use pre-configured browsers to securely connect to any websites of their choice. Instead of detecting whether the site is legitimate or not, Virtual browser allows access only to the genuine sites. Our new solution (VBEx), is a web browser plug-in, which checks for the authoritative name server and verifies the same for a visiting website. In addition to that it maintains a white list of websites that engage in online monetary transactions so that when a user requires accessing any of these, the default protocol should always be HTTPS, failing which it will prevent the web browser to load the requested website. Every time a user tries to login to a particular site, the site information will be verified with the maintained white list, if the site is not present in a white list, user will be warned for a possible attack (Kang and Lee, 2007; Cao et al., 2008; Wang, 2008; Sengar and Kumar, 2010).

Related Work

HTTP basic authentication protocol is vulnerable to phishing attacks because a client needs to reveal his password to the server that he wants to login. Most users have multiple password protected accounts over the Internet. To avoid the headache in remembering and managing a long list of different and unrelated passwords, most users simply use the same password for multiple accounts. A phisher can effectively steal users' passwords for high-security servers, such as an online banking website by setting up a malicious server or breaking into a low-security server, such as a high-school alumni website. Gouda, et al., (2007) proposed anti-phishing single password protocol. Proposed protocol allows a client to securely use a single password across multiple servers and also prevents phishing attacks. The protocol achieves client authentication without the client revealing his password to the server at any point. Therefore, a compromised server cannot steal a client's password and replay it to another server.

Although password is one of the most commonly adopted means to protect user accounts, most users are used to giving away the same very easily. Most users disregard the security functionality; they do not have the knowledge and/or the motivation to configure or to use the existing security functions correctly. Some software based protections in the client computer can help in user password management. Whenever a user wants to submit login credential into any of the phishing sites, these tools can be useful in preventing such an incident. In the recent past various client side and browser based tools have been proposed as solutions to phishing attacks. Jendricke et al. (2000) proposed Identity Manager, a security tool which offers a user interface for security functionality that is compatible with all Internet applications; so that even inexperienced users can configure and negotiate their security needs in a convenient way. In another work, Wu et al. (2006a) proposed the Web Wallet, which is a browser sidebar. It detects phishing attacks by determining where users intend to submit their information and suggests an alternative safe path to their intended site if the current site does not match it. It integrates security questions into the user's workflow so that its protection cannot be ignored by the user. Authors conducted a user study on the Web Wallet prototype and found that it significantly decreased the spoof rate of typical phishing attacks from 63% to 7%, and it effectively prevented all the phishing attacks as long as it was being used. In their concluding remarks they also pointed out that there is the possibility of web wallet getting spoofed itself in addition to the fact that there is always a human factor involved in understanding the security warnings given by these toolbars.

Kirda and Kruegel (2005); Kirda and Kruegel (2006) in their paper presented a browser extension named AntiPhish, that aimed to protect end users against spoofed website based phishing attacks. AntiPhish generates warnings whenever the user attempts to submit login credential to any of the un-trusted or spoofed websites. But in any case if the users go ahead without heeding the warnings and login to a spoofed website, the job of AntiPhish remains limited. In a similar work Ronda et al., (2008) presented iTrustPage, a FireFox extension - that does not rely completely on automation to detect phishing. Instead, iTrustPage relies on user input and external repositories of information to prevent users from filling out phishing Web forms.

Passpet, a tool that improves both the convenience and security was proposed by Yee and Sitaker (2006). It used a combination of techniques such as password hashing and user-assigned site labels to help users identify the secure sites in case of spoofed attacks. But as concluded by the author, the tool may not be of much help in case of pharming attacks or for non-SSL sites. In another work, Herzberg and Jbara (2008) proposed TrustBar, a browser extension that allows users to assign a name or logo to identify SSL/TLS-protected sites; if users did not assign a name or logo then TrustBar would identify protected sites by their own name or logo, and by the certificate authority (CA) who identified the site.

Halderman et al. (2005) in their paper proposed Password Multiplier, an implementation in the form of an extension to the Mozilla Firefox web browser. They proposed a novel technique that uses a strengthened cryptographic hash function to compute secure passwords for arbitrarily many accounts while requiring the user to memorize only a single short password or master password. This mechanism functions entirely on the clientside; no server-side changes are required. As discussed by the authors, the password multiplier has a shortcoming with respect to account password changes. There are sites which requires user to change their login password frequently, the same will not be possible with the current application and they proposed to add the same feature in their next release.

In another work Jammalamadaka et al. (2005) presented Pvault which allowed users to outsource personal data to a server which is not trusted. Data confidentiality and Integrity were preserved using cryptographic techniques. Pvault system gave its users seamless mobile access to their personal data. Pvault's auto fill feature fill out passwords or other information on websites, thereby relieving the users from filling up their details manually.

It also prevents online scams such as pharming and phishing. They also listed some of the drawbacks of the proposed system, which included the requirement of installation of the Pvault client software in all remote machines from where users need to perform online web activity. Another problem is that all the Pvault entries are guarded by one master password. If the master password is compromised, all the Pvault entries will be easily known to the adversary. It is important that the users choose a strong password as the master password.

PwdHash, a browser extension that transparently produces a different password for each site was proposed by Ross et al. (2005). It helped in improving web password security and defence against password phishing and other attacks. This browser extension applies a cryptographic hash function to a combination of the plaintext password entered by the user, data associated with the web site, and a private salt stored on the client machine. Theft of a password phished from one site will not yield a password that is useful at another site. As noted by the authors this approach may not be effective against a pharming or DNS attack or against any spyware or key logger.

Chou et al. (2004) introduced a browser plug-in called Spoof Guard. This plug-in monitors a user's Internet activity, computes a spoof index, and warns the user if the index exceeds a level selected by the user. The proposed solution uses a combination of stateless page evaluation (URL check, Image Check, Link check, Password check), stateful page evaluation (Domain check, Referring page, Image-domain association) and examination of outgoing post data to compute a spoof index. When a user enters a username and password on a spoof site that contains some combination of suspicious URL, misleading domain name, images from an honest site, and a username and password that have previously been used at an honest site, it will intercept the post and warn the user with a pop-up that foils the attack. But the proposed solution can have a very high false alarm rate because of user using the same password in different site or visiting a site for the first time. The frequent false alarm can de-motivate users to use the solution or to give proper attention to the positive notifications.

These different approaches are all preventive by nature. In a recent usability studies (Dhamija et al., 2006; Egelman et al., 2008; Wu et al., 2006b, Schechter et al., 2007; Zhang et al., 2007, Purkait, 2012b) researches have demonstrated that neither server-side security indicators nor client-side toolbars and warnings are successful in preventing vulnerable users from being deceived. This is mainly because:

a. Phishers can convincingly imitate the appearance of legitimate web sites.
b. Users tend to ignore security indicators of warnings.
c. Users do not necessarily interpret security cues appropriately.

However these approaches cannot completely foil phishing attacks, and will take long time to be effective on a global scale.

Virtual Browser Extension (VBEx)

After carefull study of existing anti-phishing tools and available counter measures and approaches we came to conclusion that Phishing is a Social engineering attack, our approach to counter that will not be the same, and we can not have a solution which will be user dependent. Our solution has to be purely technology oriented and should be difficult to breach.

Design Principles

VBEx is designed keeping in mind that human users are the weakest link in the security chain. VBEx is a method where the user's intention is obvious; hence nothing is left for the user to decide. Our design principal is that, user experience should be preserved and solution should work without involvement of user. VBEx works on binary principle of legitimate or phishing and provides a trusted path between browser and the legitimate website (Ye et al., 2005), unlike others it does not provide any user interface to accept any wrong certificate or allow user to bypass the tool. As suggested by Parno et al. (2006), VBEx will ensure that the user's data should go to the intended recipient only, it will also make sure that attacker or any other unauthorized user should not have any access to the user's data. VBEx will work is the background and allow access to the white listed domains after checking proper authoritative name servers in addition it will also ensure that concern site maintains a HTTPS connection always.

Design Assumption

VBEx follows the same design assumption as of the Virtual Browser, they are as follows:

- It is the security that matters and not the speed.
- VBEx is not meant for regular Internet browsing, as it will allow only the white listed domains.
- During registration/configuration at VB website, the user will provide the name of the institution/sites which he intends to visit with the help of VBEx (the user will have the option to add or delete from the wish list later also).

Technology

The Mozilla application framework (Mozilla, 2013) is used to create a user-installable .xpi (XPInstall) file to install the VBEx plugin in Mozilla Firefox (see Figure 1). The following are the various components of the framework:

- **CSS:** Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation semantics (the look and formatting) of a document written in a markup language.
- **DOM:** The Document Object Model (DOM) is a cross-platform and language-independent convention for representing and interacting with objects in HTML, XHTML and XML documents.
- **Gecko:** Gecko is a free and open source layout engine designed for performance and portability.
- **JavaScript:** JavaScript (JS) is the primary language of Mozilla browsers. It is an interpreted computer programming language. It was originally implemented as part of web browsers so that client-side scripts could interact with the user, control the browser, communicate asynchronously, and alter the document content that was displayed.
- **XPI:** A XPI (XPInstall) file typically contains the resources to be installed and an install script that guides the installation process.
- **XUL:** XUL, which stands for XML User Interface Language, is the basis of user interface. It is an application of XML

(Extensible Markup Language) that defines various user interfaces elements, mostly widgets, control elements, template, etc. It is similar in many ways to HTML.

b) **NSlookup module:** This module compares the answers for 'A' and 'NS' record of the requested web site with local DNS answers and that of the authoritative name server of the requested domain. Authoritative name server lists are kept and managed for all the white listed website in VB. If all answers are not same then it kills the page loading process for the requested website and displays the pre-designed website block page (See Figure 3).

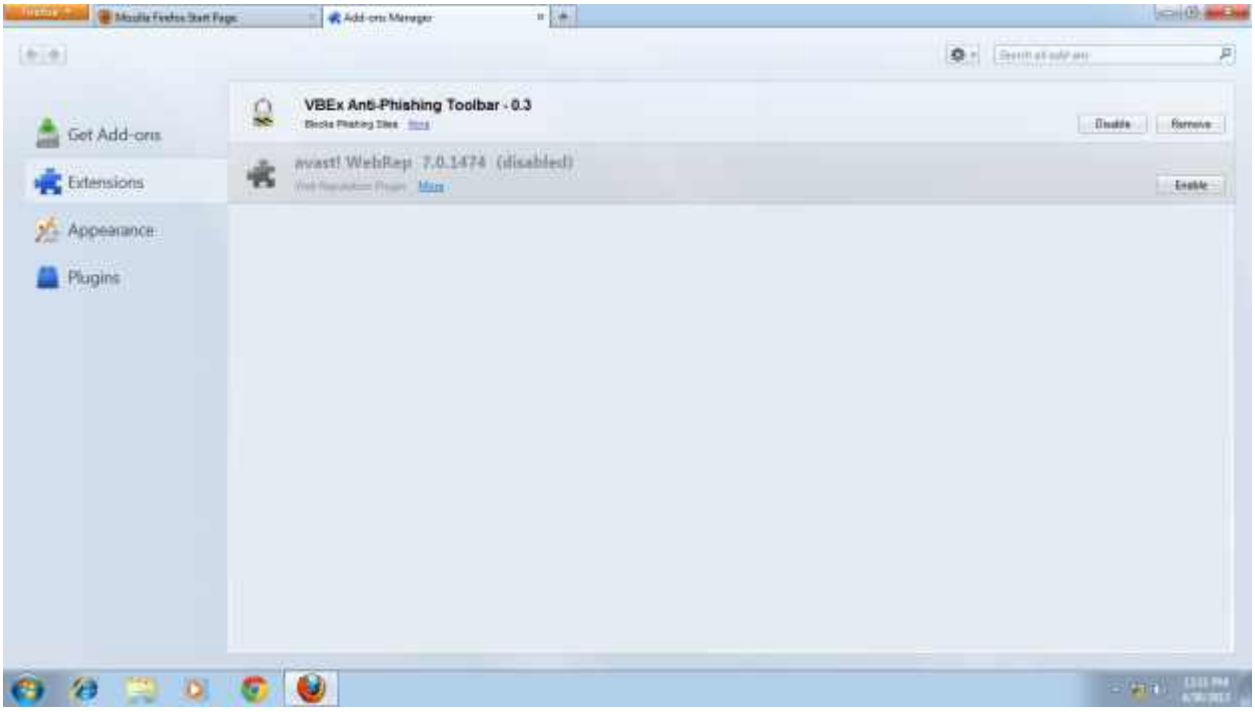

Figure 1: VBEx Extensions in Firefox Add-ons Manager

VBEx Functionality

VBEx has three different modules, All these modules work at the background when a user enters a web address on the browser address bar.

a) **White list module:** When a user requests a URL, this module becomes active and performs a domain look up in white list stored in the local computer. If the domain is found in the white list the URL is passed to the next module for a Name server lookup. If the domain URL is not found on white list, it kills the page loading process and loads pre-designed website block page (See Figure 2). It also periodically checks and updates the locally stored white list with the central VB server. It will also provide an interface which will allow user to request for inclusion of a specific website in the white list.

c) HTTPS detection module: In this last module, we check whether https is used for the requested website or not if so, whether the certificate check succeeded or failed. If the protocol use is https and the certificate is valid for the domain, the website will be loaded else the loading process will be terminated and a pre-designed website block page will be loaded (See Figure 5 and 6). Sample code for HTTPS detection for the site www.google.com is shown in Figure 4.

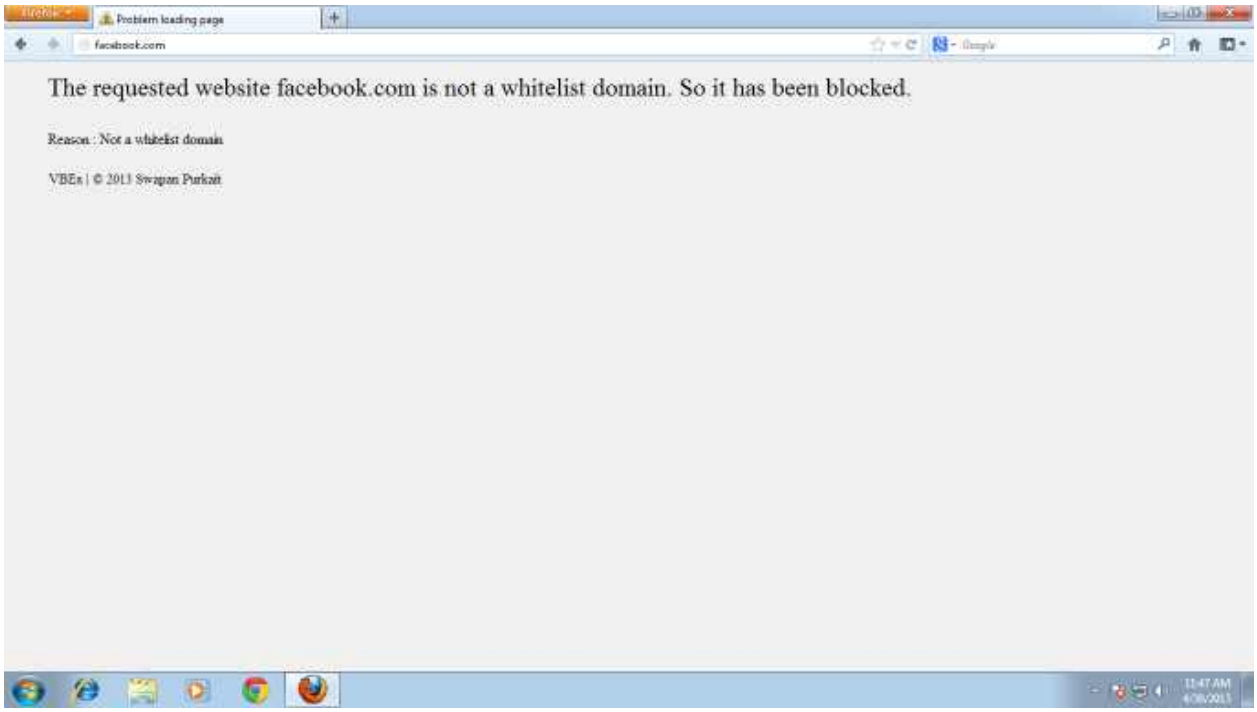

Figure 2: Website blocked as it is not a white listed domain

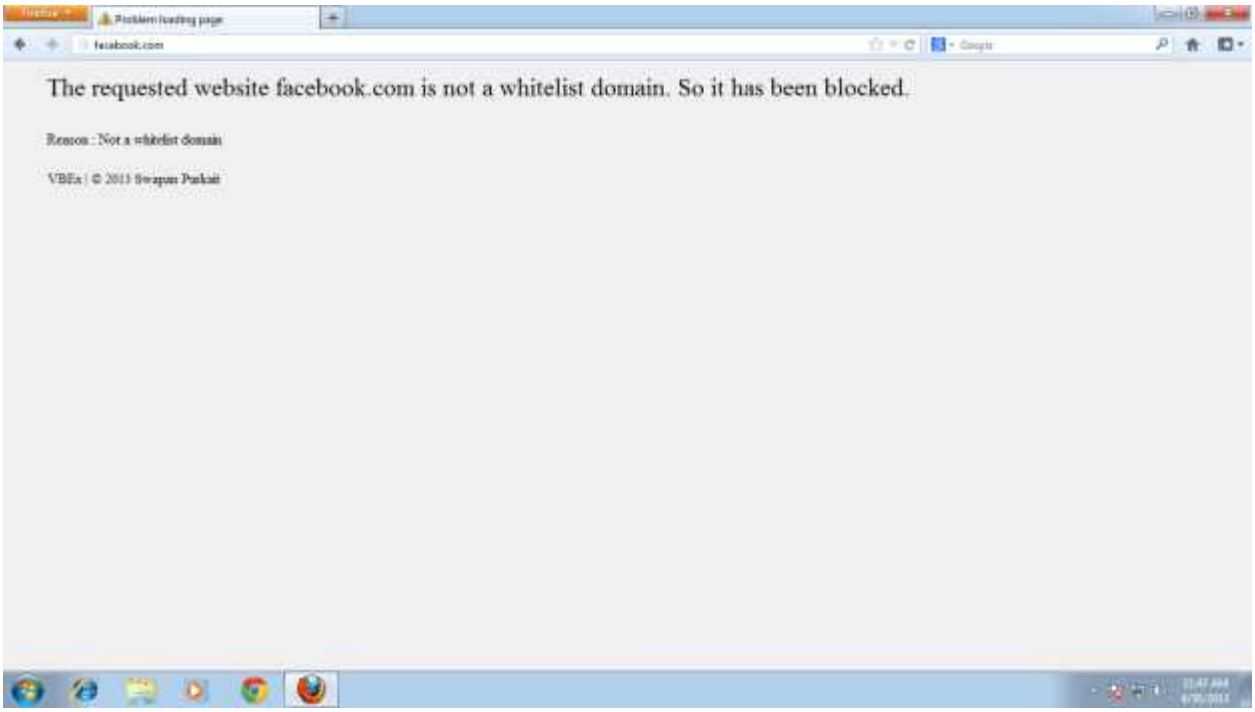

Figure 3: Website blocked due to DNS records mismatch

```
function CheckSSL(channel) {
    try {
        const Ci = Components.interfaces;
        var secInfo = channel.securityInfo;
        if (secInfo instanceof Ci.nsITransportSecurityInfo) {
            secInfo.QueryInterface(Ci.nsITransportSecurityInfo);
        }
        if (secInfo instanceof Ci.nsISSLStatusProvider) {
            var cert = secInfo.QueryInterface(Ci.nsISSLStatusProvider).
            SSLStatus.QueryInterface(Ci.nsISSLStatus).serverCert;
            var verificationResult = cert.verifyForUsage(Ci.nsIX509Cert.CERT_USAGE_SSLServer);
            switch (verificationResult) {
                case Ci.nsIX509Cert.VERIFIED_OK:
                    break;
                case Ci.nsIX509Cert.NOT_VERIFIED_UNKNOWN:
                    document.content.body='This website is a fraud and so blocked';
            }
        }
    } catch(err) {
        alert(err);
    }
}
var pageMod = require("page-mod");
pageMod.PageMod({
  include: ["http://www.google.*", "http://www.google.com/*"],
  contentScript: 'CheckSSL(document.location.url)',
    contentStyle: ["body { color:red}", "h1 { font-size:100px;color:red;}"
  ]
});
```

Figure 4: Sample code for HTTPS detection for the site www.google.com

Figure 5: Website blocked as SSL certificate expired

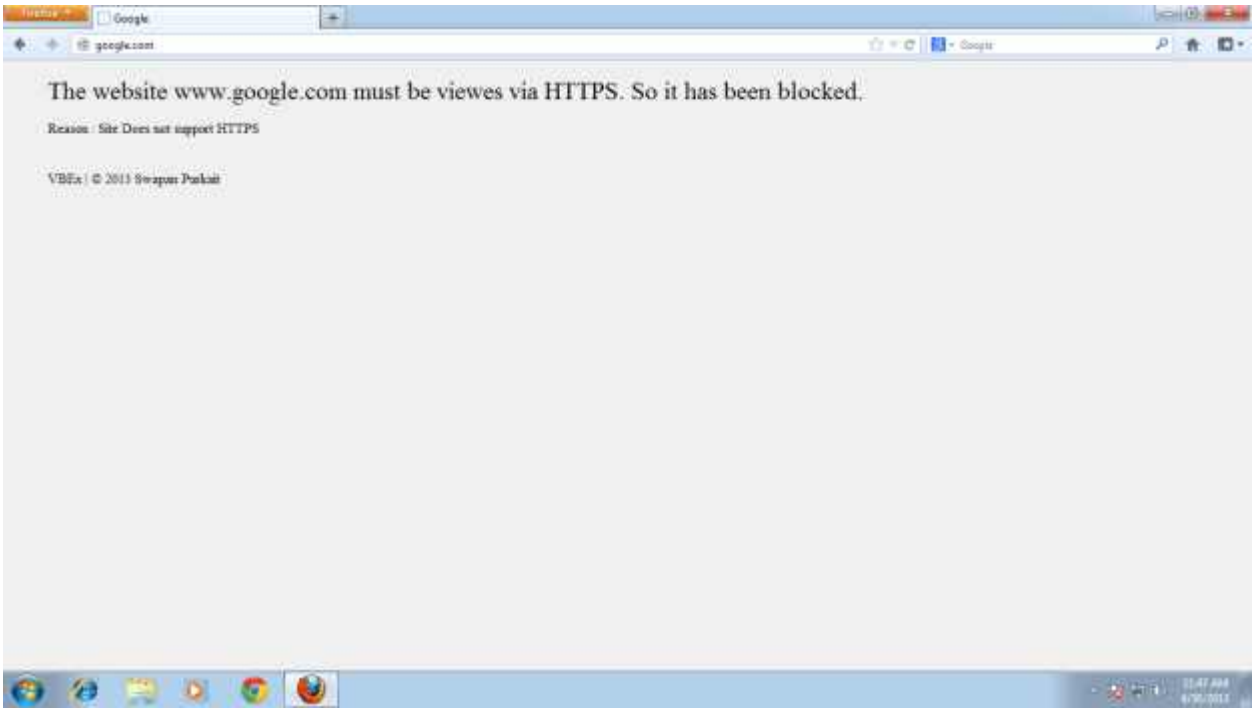

Figure 6: Website blocked as requested site does not support HTTPS

VBEx Evaluation

a) Selection of website

For evaluation purpose, one search engine site, one web based email service provider, one social site and two websites of well known Indian banks were selected. The selection list is as follows:

a) google.com
b) gmail.com
c) orkut.com
d) icicibank.com
e) onlinesbi.com

b) Test Preparation

In a single DELL laptop (6 GB RAM) two Red Hat Enterprise Linux servers were installed. A copy of the original website was needed to create a spoof of the same. Source pages of all the sites listed above were copied and saved locally. Copying only the source code without the images and script would cause distortion of the page as the same will display broken link for all images. To prevent that all the required images and JAVA and CSS scripts were also copied manually. All links were modified to use relative URL link such a way that the spoofed website should not send any resource request to the legitimate website. The action tag on the login page was modified and was directed to a "submit.php" page. PHP's file handling functions were used in "submit.php" to store the personal information of the user such as UserID and Password in a "pass.txt" file.

Out of the two server installed server-1 was the DNS and server-2 worked as the Web server. For the web server configuration, virtual web hosting was used to host all spoofed websites on port 80 (HTTP protocol) on server-2 (see Table I, for Basic server configuration details). Server-1 had DNS entries for all domains, which resolved all domain queries (A records) to the server 2's IP address. One of the network interfaces of server-1 was connected to the Internet and the same was configured for packet forwarding and worked as a gateway to the client computer. For the evaluation purpose, VBEx were installed as extension in Mozilla Firefox in a Windows 7 computer, with the following IP configuration.

IP Address	: 10.100.1.100
Subnet Mask	: 255.0.0.0
Default Gateway	: 10.100.1.1
DNS Servers	: 10.100.1.1

c) Execution and Analysis

After connecting the server laptop with the LAN network, all the required services were started. When a client computer requests for any website the DNS request will go to server-1. If the requested website is among any of the website hosted locally, the server-1 will send back the answer with the resolved IP address back to the client computer. After getting the required IP address the client computer will ask for the pages from the server-2 which is the web server. It will send the spoofed website to the user's browser and complete the process. The phishing webpage's were clever replicas of the legitimate site.

All four sites hosted locally were accessed from the Firefox which had VBEx installed. As the 'A' record address fetched by the local DNS server was not the same as the record fetched from the authoritative name server the page loading process was terminated, and VBEx loaded the pre-defined blocked page as shown in Figure 3.

For the evaluation of the white list module, we typed facebook.com in the address bar, as the same was not in the white list. The VBEx terminated the page loading process and showed the block page as shown in Figure 2.

For the third test for https detection we had to deactivate the NSLookup module as it was anyway blocking all sites hosted locally. After the deactivation when we opened google.com in the web browser, the white list module pass the page loading but the HTTPS detection module stopped the page loading as it detected that the site was not running on HTTPS and showed error as shown in Figure 6.

For the SSL certificate validity test, we removed the local DNS and the local web server. The ISP DNS address was used in the client computer for DNS resolving. We changed the system date of the client computer to future date in 2013. As the SSL certificate for the domain google.com was valid till 30-12-2013, VBEx stopped the page loading process and showed the error page as shown in Figure 5.

legitimate or phishing and provides a trusted path between browser and the legitimate website, unlike others it does not provide any user interface to accept any wrong certificate or allow user to bypass the tool. It will only allow a legitimate site to open and prevents all others to load on the web browser. We conducted series of lab test on the VBEx prototype and found that VBEx is a promising approach.

More functionality can be added to VBEx for protecting user against key-loggers and screen-grabbers and client side scripting attacks. New interface can be added where to user can submit request for addition of a new website in the white list. Further studies are required to determine the user acceptance and motivation to use VBEx. The current plug-in is designed for Mozilla Firefox, the same can be designed for other web

Table I: Basic server configurations for VBEx Evaluation

Settings	Server 1	Server 2
OS	Red Hat Linux	Red Hat Linux
Services Installed	DNS	Web
Basic IP Address	10.100.1.1/8	10.100.1.2/8
Gateway	Default gateway for the computer lab in the LAN network	10.100.1.1
DNS	Default DNS server for the Computer lab in the LAN network	10.100.1.1
DNS Entries for A Record	All sites had their A record as 10.100.1.2 (IP address of Server 2)	Not applicable
Web site Hosted	Not applicable	google.com gmail.com orkut.com icicibank.com onlinesbi.com

Conclusions and Future Work

The rapid growth in phishing attacks and its sophistication has spurred up the calls for its solutions. A number of different countermeasures have been proposed, ranging from quick fix changes to more substantial redesigns. These different approaches are all preventive by nature. Recent studies have demonstrated that neither server-side security indicators nor client-side toolbars and warnings are successful in preventing vulnerable users from being deceived. In this study we proposed a new phishing countermeasures called Virtual Browser Extension (VBEx) a web browser plug-in, which checks for the authoritative name server and verifies the same for a requested website. VBEx can be downloaded from the virtual browser site on the Internet after user authentication. The plug-in comes with individual white list for the user depending upon the sites selected or registered with VB. Registered user can download the plug-in multiple times in different computers depending upon their requirement. VBEx searches for the authoritative name server responsible for the domain and queries for the IP address for the domain, thus it prevents any kind of misuse of DNS entries on the host files of the local computer or on the local DNS server. In addition to that it maintains a white list of websites that engage in online monetary transactions so that when a user requires accessing any of these, the default protocol should always be HTTPS, failing which it will prevent the web browser to load the requested website. VBEx works on binary principle of browsers such as Internet Explorer, Chrome and Opera. Further work is required for encrypting and securing the white list maintained locally by the plug-in. Additional work is required for usage of SSL or any other secure communication between the plug-in and the central server for regular update.

Reference

1. *APWG (2012), "Global Phishing Survey: Trends and Domain Name Use in 1H2012", Anti-Phishing Working Group (APWG), http://docs.apwg.org/reports/APWG_GlobalPhishingSurvey_1H2012.pdf (accessed April 2013).*
2. *Anderson, K. B., Durbin, E., and Salinger, M. A., (2008), "Identity Theft", Journal of Economic Perspectives, Volume 22, Number 2. Spring 2008, pp. 171-192.*
3. *Brody, R.G., Mulig, E. and Kimball, V. (2007), "Phishing, Pharming and Identity Theft", Academy of Accounting and Financial Studies Journal, Volume 11, pp. 43-56.*
4. *Butler, R. (2007), "A Framework of anti-phishing measures aimed at protecting the online consumer's identity", The Electronic Library, Volume 25, Issue 5, pp. 517-533.*
5. *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, Virginia, USA.*
6. *Chou, N., Ledesma, R., Teraguchi, Y. and Mitchell, J. C. (2004), "Client-side defence against web-based identity theft",*

Proceedings of the Network and Distributed System Security Symposium, NDSS 2004, San Diego, California, USA.

7. *Dhamija, R., Tygar, J. D. and Hearst, M. (2006), "Why phishing works". Proceedings of the SIGCHI conference on Human Factors in computing systems, New York, NY, USA, ACM Press, pp. 581-590.*
8. *Forte, D. (2009), "Anatomy of a phishing attack: A high-level overview", Network Security, April, pp. 17-19.*
9. *Egelman, S., Cranor, L. F. and Hong, J. (2008), "You've been warned: an empirical study of the effectiveness of web browser phishing warnings", In CHI '08: Proceeding of the twenty sixth annual SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM, pp. 1065-1074.*
10. *Gastellier-Prevost, S., Granadillo, G.G. and Laurent, M. (2011a), "A Dual Approach to Detect Pharming Attacks at the Client-Side", Proceeding of 4th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Paris.*
11. *Gastellier-Prevost, S., Granadillo, G.G. and Laurent, M. (2011b), "Decisive Heuristics to Differentiate Legitimate from Phishing Sites", Proceedings of the Network and Information Systems Security (SAR-SSI), La Rochelle.*
12. *Gorling, S. (2007), "An Overview Of The Sender Policy Framework (SPF) As An Anti-Phishing Mechanism", Internet Research, Vol. 17 No. 2, 2007, pp. 169-179.*
13. *Gouda, M. G., Liu, A. L., Leung, L. M. and Alam, M. A. (2007), "SPP: An anti-phishing single password protocol". Computer Networks, Volume 51, Issue 13, 12 September 2007, pp. 3715-3726.*
14. *Halderman, J. A., Waters, B. and Felten, E.W. (2005), "A convenient method for securely managing passwords". Proceedings of the International World Wide Web Conference (WWW), pp. 471-479.*
15. *Herzberg, A. and Jbara, A. (2008), "Security and Identification Indicators for Browsers against Spoofing and Phishing Attacks". ACM Transactions on Internet Technology, Vol. 8, No. 4, Article 16.*
16. *Jammalamadaka, R. C., Mehrotra, S. and Venkatasubramanian, N. (2005), "Pvault: A Client Server System Providing Mobile Access to Personal Data". Proceedings of the 2005 ACM International Workshop on Storage Security and Survivability. StorageSS 2005, Fairfax, VA, USA, pp. 123-129.*
17. *Jendricke, U. and Markotten, D. G. (2000), "Usability meets security - the Identity-Manager as your personal security assistant for the Internet". Proceedings of the 16th Annual Computer Security Applications Conference, pp. 344-353.*
18. *Karlof, C.,Tygar, J.D., Wanger, D., and Shankar, U., (2007), "Dynamic Pharming Attacks and Locked Same-origin Policies for Web Browsers", Proceedings of the 14th ACM conference on Computer and communications security. October 28-31, 2007, Alexandria, Virginia, USA. pp. 58-71 .*
19. *Kang, J. and Lee, D. (2007), "Advanced White List Approach for Preventing Access to Phishing Sites", Proceedings of International Conference on Convergence Information Technology, Gyeongju, pp. 491-496.*
20. *Kirda, E. and C. Kruegel. (2005), "Protecting Users against Phishing Attacks with AntiPhish", Proceedings of the 29th Annual International Conference on Computer Software and Applications. COMPSAC 2005, Edinburgh, Scotland, pp. 517-524.*
21. *Kirda, E. and C. Kruegel. (2006), "Protecting users against phishing attacks". The Computer Journal, Volume 49, Issue 5, pp. 554–561.*
22. *Mozilla (2013), "Building an extension, Mozilla Developer Network" https://developer.mozilla.org/en-US/docs/Building_an_Extension, (accessed May 2013).*
23. *Parno, B., Kuo, C. and Perrig, A. (2006), "Phoolproof Phishing Prevention", Financial Cryptography and Data Security Lecture Notes in Computer Science, Volume 4107, pp. 1 -19.*
24. *PhishTank (2012), "PhishTank > Statistics about phishing activity and PhishTank usage > July 2012", available at: http://www.phishtank.com/stats/2012/07/(accessed April 2013).*
25. *Purkait, S. (2012), "Virtual Browser: An On-Demand Service to Prevent Phishing Attacks", The IUP Journal of Information Technology, Vol. VIII, No. 2, pp. 7-23.*
26. *Purkait, S. (2012a),"Phishing counter measures and their effectiveness - literature review", Information Management & Computer Security, Vol. 20, Issue 5, pp. 382-420.*
27. *Purkait, S. (2012b), "Exploring the factors that influence an internet user's ability to correctly identify phishing websites", The IUP Journal of Information Technology, Vol. VIII, No. 3, pp. 7-38.*
28. *Purkait, S. (2013), "DHCP-Enabled LAN Prone to Phishing Attacks", The IUP Journal of Information Technology, Vol. IX, No. 1, pp. 24-40.*
29. *Ronda, T., Saroiu, S. and Wolman, A. (2008), "iTrustPage: A User-Assisted Anti-Phishing Tool", Proceedings of the 3rd ACM SIGOPS/EuroSys European Conference on Computer Systems 2008, pp. 261-272.*
30. *Ross, B., Jackson, C., Miyake, N., Boneh, D. and Mitchell, J. C. (2005), "Stronger password authentication using browser extensions". Proceedings of the USENIX Security Symposium, pp. 17–32.*
31. *Schechter, S. E., Dhamija, R., Ozment, A. and Fischer, I. (2007), "The emperor's new security indicators", SP '07 Proceedings of the 2007 IEEE Symposium on Security and Privacy, pp. 51-65.*
32. *Sengar, P. K. and Kumar, V. (2010), "Client-Side Defense Against Phishing With Pagesafe", International Journal of Computer Applications, Volume 4, Number 4 - Article 2, pp. 6 - 10.*
33. *Verisign (2012), "The Domain Name Industry Brief", Volume 9, Issue 3, October 2012; http://www.verisigninc.com/assets/domain-name-brief-oct2012.pdf (accessed April 2013).*
34. *Wang, Y., Agrawal, R. and Choi, B.Y. (2008), "Light Weight Anti-Phishing with User Whitelisting in a Web Browser", Proceedings of the IEEE Region 5 Conference, Kansas City, MO.*
35. *Wu, M., Miller, R. C. and Little, G. (2006a), "Web Wallet: Preventing Phishing Attacks by Revealing User Intentions", In SOUPS '06: Proceedings of the second symposium on Usable privacy and security, Pittsburgh, PA, USA, pp. 102-113.*
36. *Wu, M., Miller, R. C. and Garfinkel, S. L. (2006b), "Do security toolbars actually prevent phishing attacks?", In CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems, New York, NY, USA, ACM Press, pp. 601-610.*
37. *Ye, Z., Smith, S. and Anthony, D. (2005), "Trusted Paths for Browsers". ACM Transactions on Information and System Security, Vol. 8, No. 2, May 2005, pp. 153–186.*
38. *Yee, K. P. and Sitaker, K. (2006), "Passpet: Convenient Password Management and Phishing Protection", In SOUPS '06: Proceedings of the second symposium on Usable privacy and security, Pittsburgh, PA, USA, pp. 32-43.*
39. *Zhang, Y., Egelman, S., Cranor, L. and Hong, J. (2007), "Phinding Phish: An Evaluation of Anti-Phishing Tools", Proceedings of the ISOC Symposium on Network and Distributed System Security, Internet Society.*