Suraksha: A FireFox Addon

Hilay Khatri and Pratik Bosamiya
{hmkhatri,pvbosami}@ncsu.edu

December 2, 2011

Abstract

Hundreds of new websites are developed on a daily basis and sometimes it becomes difficult to analyze which websites are authentic. Security related research in Web had gained significant interest from last few years since the birth of e-commerce, since everything can be done online. Hence, it is important to protect privacy and prevent misuse of such valuable services. To protect user's privacy, we propose a Firefox add-on called SURAKSHA which helps in preventing phishing attack, detects abrupt changes in the websites certificate and alerts user about un-encrypted transfer of credentials. The user will be informed about the authenticity of the web-site, even before the website is loaded by taking pro-active steps.

1 Introduction

Since the birth of e-commerce, research related to Web security had increased manifold. Today, from shopping, blogging, social engineering to stock investments, banking, booking tickets can be done online. As banking, online shopping, stocks investments and such services require access to credit card information it becomes necessary to protect the information provided by the user to the website. The E-Commerce field has been infected with many problems with the growth of phishing. As a result people are losing their trust in E-Commerce. Megaw and Flowerday [35] explain how phishing is slowly hampering the growth of E-Commerce and suggest a proper balance between Trust and Controls need to be maintained. Phishing websites looks exactly the same as the legit website, however they steal username, password and other credentials.

Identify theft is one of the most glaring threat to the E-Commerce sector. Detecting a websites authenticity is a primary task for a user, as it is used to make payments or providing credentials to use the services provided by the website. Users need to be made aware of threats posed by Phishing websites and MITM (Man-in-the-Middle) attacks where ones secure credentials are put at the risk of being stolen. MITM attack is where third party user intercepts the communication between two parties i.e. it establishes independent connections with the victims and relays messages between them making them believe that they are talking to each other, whereas the entire conversation is controlled by the attacker. Web servers now-a-days tend to use un-secure connections for authentication purpose only to provide a faster and a smoother browsing experience; however credentials of the customer are put at risk due to this. Also, there have been incidences of government forcing certificate authorities to issue SSL certificates for surveillance and monitoring purpose [37]. In this paper, we present a solution to the above addressed problems via a Firefox add-on called Suraksha.

SSL/TLS ensures that communication between client and the server is secured, thereby ensuring privacy of users credentials. The add-on monitors SSL certificates provided by servers for abrupt changes, i.e. changes in the Hash, name of the CA or country of the CA which may be suggestions of an attack in place. It also makes sure that user credentials are transmitted in a secure manner to the server by imposing SSL/TLS encryption while submitting secure credentials. For protecting against phishing websites, a user is asked to assign a secret pass code for each website. Each time the user visits the website the pass code is verified on the basis of URL and the passcode helps in detecting the phishing websites. The add-on ensures safe and secure browsing. It also ensures that the user interaction is kept at the minimal level.

The organization of this paper is as follows: In Section 2, we review related work done to prevent phishing attacks, man in the middle attacks and transmission of data over unsecured connection. In Section 3, we discuss about our proposed solution. In Section 4, we explain our Design model and Graphical User Interface (GUI) of our addon. In Section 5, we discuss implementation details. In section 6, evaluate our addon by conducting different experiments to test the effectiveness our approach. In Section 7, we discuss limitation of the addon. In Section 8, we discuss about possible future research work on design patterns. Finally, in Section 9, we conclude by summarizing the contents of the paper.

2 Related Work

According to the study of Zhang et al. [41], past work of anti-phishing falls into the following four categories : studies of why people fall for phishing attacks; methods of educating people about phishing attacks; the development of better user interfaces for anti-phishing tools; and automated tools to detect phishing. Various approaches have been used for phishing website detection, including heuristics [8, 28]. blacklists [28], whitelists [5] and classifications because a large number of people cannot differentiate between legitimate and phishing web sites [12], even when they are made aware that their ability to identify phishing attacks is being tested [25].

Anti-phishing tools use two major methods for detecting phishing sites [8]. The first is to use heuristics to judge whether a page has phishing characteristics. For example, some heuristics used by the SpoofGuard toolbar include checking the host name, checking the URL for common spoofing techniques, and checking against previously seen images. Some toolbars, such as Netcraft [28]. seem to use a combination of heuristics and blacklisted URLs that are verified by paid employees. The second method is to use a blacklist that lists reported phishing URLs. For example, Cloudmark [21] relies on user ratings to maintain their blacklist. In contrast, Y. Cao et al. [5] developed Automated Individual White-List (AIWL) that automatically maintains a list of familiar Login User Interfaces (LUIs) of web sites. Once a user tries to submit confidential information to LUI that is not in the white-list, AIWL will alert the user of the possible attack. W. Yu et al developed PhishCatch [40] using a heuristic algorithm that is focused on detecting phishing links and building an extensive data warehouse containing a wealth of information about phishing which can be further used to analyze trends in phishing. CANTINA [41] is a content-based approach to detecting phishing web sites, based on the Frequency/Inverse Document Frequency (TF-IDF) information retrieval algorithm. However, it does not include a dictionary for languages other than English. Chen and Guo [7] proposed a new end-host based anti-phishing algorithm, called LinkGuard, that utilizes the generic characteristics of the hyperlinks in phishing attacks that are derived by analyzing the phishing data archive provided by the anti-phishing working group (APWG). Due to its generic characteristics, LinkGuard can also detect unknown phishing attacks. Researchers [25] had also tried to classify security risk elements for evaluating the security risk of a website against the phishing attacks.

A large number of tools have been developed to prevent phishing. EBays Account Guard [14] uses colored icons to indicate site authenticity. A green icon indicates that current site belongs to eBay or PayPal, a red icon indicates blacklisted phishing site and a gray icon for all other sites. Joshi et al. [24] developed an algorithm used in browser plug-in to detect phishing attacks where false credentials are submitted to the Phishers and depending upon the response from the Web-Server the authenticity of the Website is determined. Ronda et al. [31] also presented an anti-phishing tool as downloadable extension to Firefox which relies on user input and external repositories of information to prevent users from filling out Phishing Web forms. However, a study by M. Wu et. al showed that security toolbars are still susceptible to high-quality phishing attacks [38].

Researchers even conducted surveys to examine the efficiency of security warnings on users. Schechter et al. [34] contributed a study design to observe participants as they log into security-critical web sites with their own authentication credentials. Using this design, they measured the efficacy of security indicators (HTTPS indicators and site-authentication images). Sunshine [37] conducted a survey of over 400 Internet users to examine their reactions to and understanding of current SSL warnings and found that far too many people exhibited dangerous behavior in warning conditions. P. Kumaraguru et al. [26] started an awareness program using training email system that taught people about phishing during their normal use of email. With the help of lab experiments they found that embedded training works better than the current practice of sending security notices. PHONEY [6] is a phishing email detection system that tries to detect phishing emails by mimicking user responses and providing fake information to suspicious web sites that request critical information. The web sites responses are forwarded to the decision engine for further analysis.

The ideas underlying usable security were first discussed in 1975 by Saltzer and Schroeder, who included psychological acceptability in their list of essential principles for information protection systems [33]. To make sure that a user can easily detect phishing attack, techniques based on visualization have also been developed. Dynamic Security Skins [11] proposed to use a randomly generated visual hash to customize the browser window or web form elements to indicate the successfully authenticated sites. PassMark [27] includes a personalized image in a web page to indicate that the user has setup an account with the site. Rosiello et al. proposed detecting a phishing page by leveraging layout similarity information to distinguish between malicious and legitimate web pages [32]. However as HTML can be described by various methods this method cannot help distinguish malicious from legitimate web pages well. [18] M. Hara et al developed a phishing detection mechanism based on visual similarity among phishing sites that mimic the same victim site. Dunlop et al. [13] proposed a scheme that protects against zero-day phishing attacks. The scheme captures the image of the page and uses optical character to

convert it into text, thereby leveraging Google Page Rank algorithm to decide the validity of the website. However, this approach places the burden on users to notice the visual differences between a good site and a phishing site and then correctly infer that a phishing attack is underway.

Often at times, a user ignores the security information of the website as delivered by the phishing detection tool. The Web Wallet [39] detects phishing attacks by determining where users intend to submit their information. It integrates security questions in to the users work flow so that its protection cannot be ignored by the user. The E-Commerce field has been plagued with many problems with the exponential growth of Phishing and thereby people are gradually losing their trust in E-Commerce. Megaw and Flowerday [29] explain how phishing is slowly hampering the growth of E-Commerce and suggest a proper balance between Trust and Controls need to be maintained.

In the recent years, many people in the security community have commented on the state of the SSL public key infrastructure, and the significant trust placed in the CAs [17, 3, 16]. Crispo and Lomas also proposed a certification scheme designed to detect rogue CAs [10], while the Monkeysphere project has created a system that replaces the CA architecture with the OpenPGP web of trust [1]. Sagoian and Stamm [35] introduced Compelled certificate creation attack whereby government agencies force Certificate Authorities to issue false SSL certificates to covertly intercept an individuals connection. They also discuss the design of a browser add-on that detects and prevents such attacks.

Several browser-based tools have been created to help protect users against SSL related attacks. Kai Engert [15] created Conspiracy, a Firefox add-on that provides country-level CA information to end-users in order to protect them from compelled certificate creation attacks. The Conspiracy tool displays the flag of the country of each CA in the chain of trust in the browser's status bar. However, according to Herley [19], this is an unreasonable burden to place upon end-users. Similarly there exists addons where users recognize CA by associating phrases or logos with them. Herzberg and Jbara created TrustBar, a Firefox add-on designed to help users detect spoofed websites. Trustbar makes use of SSL, by highlighting the logos of the website and its certificate authority (CA) also allowing the user to assign a per-site name or logo, to be displayed when they revisit to each site [20]. Tyler Close created Petname Tool, a Firefox add-on, that caches SSL certificates, and allows users to assign a per-site phrase that is displayed each time they re-visit the site in the future. In the event that a user visits a spoofed website, or a site with the same URL that presents a certificate from a different CA, the user's specified phrase will not be displayed [9].

Instances were reported to Certificate Authorities about the circulation of rogue certificates. Anka developed an add-on for the Firefox browser to detect and warn users about certificate chains that use the MD5 algorithm for RSA signatures [2]. The add-on was an outcome of Stevens et al. [36] demonstration that flaws in the MD5 algorithm could be used to create rogue SSL certificates (without the knowledge or assistance of the CA). Attackers avoid using SSL certificates from obscure CAs because using a CA that is not known to the browser will trigger a warning and thus might raise the users suspicion. Also, Ford and Howard [4] showed its possible to attack Web-based connections secured via HTTPS by exploiting some properties of common LANs as well as typical behaviors of inexperienced users. Younan et al. [30] discussed about a new attack against the common usage of SSL surfaced, SSL stripping. Using a man-in-the-middle attack one can suppress such messages and provide the user with "stripped" versions of the requested website forcing him to communicate over an insecure channel.

Users almost never request secure pages explicitly but rather rely on the servers, to redirect them to the appropriate secure version of a particular website. Jackson and Barth devised the ForceHTTPS system to protect users who visits SSL protected websites, but are vulnerable to man in the middle attacks due to the fact that they do not type in the https:// component of the URL [22]]. This system has since been formalized into the Strict Transport Security (STS) standard proposal [23], to which multiple browsers are in the process of adding support. While this system is designed to enable a website to hint to the browser that future visits should always occur via a HTTPS connection.

Bank of america had implemented a security system to prevent phishing attacks by associated an image with each user on login screen. Only when the user identifies the image and the text describing it, should the user enter the credentials. This approach is very useful as the image and the describtion would be private to the user and it would be difficult for the phishing website to copy those secure information.

We propose to create a knowledge base phishing detection system where a user initially authenticates a new website manually thereby updating the database of trusted URLs.We plan to extend the approach used by Bank of America, where we will be associating small piece of information as obtained from userwith each website's login page. Inititally the database will contain list of most trusted URL's. Also, whenever a user tries to enter his credentials on an unsecured website, an audio-visual warning will be triggered by the tool to alert the user of data being trasnsmitted in an unsecured manner. Use of audio for alerting users will help us effectively handle problems

faced by inexperienced users. Our tool will not only prevent Phishing attacks but also alerts users in case fake SSL certificates are presented to the users not issued by a trusted CA.

3 Our Solution

We developed a solution called 'Suraksha', which has been implemented as a Firefox Add-on. It is developed using XUL, a XML User Interface Language, which provides powerful APIs that allow developers to create add-ons for Firefox using HTML, CSS and JavaScript. XUL helps in building cross-platform applications which can run connected or disconnected from the internet. In order to preserve data, we have used SQLite Manager to manage out database. We proposed three different functionalities that help in detecting and preventing; (1) Unsecure transfer of credentials; (2) Man in the middle attack and; (3) Phishing attack.

In order to detect transfer of unsecure-credentials, our add-on detects the protocol used for transferring the information. If the protocol uses unsecure connection, it will notify the user about it. Most of the tools developed generally informs user about unsecure connection, once you enter your username and password and hit enter, as seen Firefox. However, our tool detects the protocol used for connection even before the webpage is loaded. This will alert user about the unsecure connection and will alert him to proceed with caution. The add-on will automatically redirect the user to secure connection.

We detect phishing attack by creating a user-base knowledge containing passcodes as entered by the user. The "passcode" can be anything from a small number of characters to large string. We have initialize knowledge base with some of the most used websites based on ALEXA ranking. This websites are genuine and are initialized by us. When a user visits a website, if its data is there in the knowledge-base, it will prompt user to enter a passcode to associate some piece of information with the website. As user will associate some information with the website, that passcode itself becomes the visual identity of that website and hence a user can trust the website as it is genuine. Whenever a user-visits a phishing website, its data would not be there in the knowledge-base and it will alert user about websites authenticity. Whenever a user re-visits that same website, the add-on will fetch details from the knowledge-base and will display it to the user. Through these details, the user will come to know that this website is not a phishing website as he had already visited this site before.

Immediate changes in SSL certificates are signs are active surveillances carried out by spying or government services [37]. The government compels CA to create legitimate certificate which can be monitored by them. As the certificates are legitimate, the browser would not detect any suspicious activity however, the connection is being monitored. In order to detect this type of man in the middle attack occurred as result of compelled certificate creation, we plan to monitor SSL certificates of websites, i.e. storing details of certificates in a database and then check them for changes when a user revisits the website. Any abrupt changes in the system are immediately reported to the user so that the user proceeds with caution.

4 Design and Graphical User Interface (GUI)

4.1 Design

Major web browsers are still sometimes vulnerable to MITM attack caused by compelling CAs to produce legitimate certificates for surveillance purpose. Here we introduce a light weight browser add-on that will prevent the impact of the attack on the users. In this section motivations of the design are outlined along with the threat model of the system. Next section is concerned with the implementation details.

4.1.1 Motivations

An MITM attack caused by forcing CAs to produce legitimate certificates is an example of very low probability attack. Although being less probable the implications of the attack are very severe, so it is necessary that users are made aware of it at the right instant. As for all detection techniques it is must that the false positive rate should be as low as possible else users will refrain from using the enhancement.

Since novice users are very unlikely to know of existence of these attacks it is must that the system requires no configuration and maintenance from the user end. The enhancement should perform appropriate background processing and alert the users when suspicious activities are detected. The solution at the same time should be self contained not compromising the privacy of browsers clients. We also believe that an average user has no knowledge of what an SSL is, existence of CA and MITM attacks so it is unreasonable to expect users to learn them, however to some extent our system requires users to make some decisions as that they may not well equipped to take. Consumers are nowadays advised to look for the lock icon the address bar to ensure privacy and secure transmission of their credentials, so it becomes obliging that we keep user interaction to the minimum.

Phishing attacks which have been on the rise in the last decade are well known to average users, but building a

self contained system in such case seemed difficult since user has to in some extent figure out details of the website. The interaction can be reduced by allowing users to assign a short pass-code, but that too would require interaction in some extent from the user end but it will be at its minimal.

We do not imply that users are completely clueless about these attacks, but its the browsers that handle the handle such critical decisions so that user experience is smooth.

4.1.2 Threat Model

The most important aspect while designing a security system is the threat model. Threat model is basically a collection of attackers abilities. An attacker performing an MITM attack has the capabilities to perform masquerading, eavesdropping and message manipulation. These threats have very severe implications and can exploit even the smallest vulnerabilities in your system. A powerful attacker in case of MITM possesses ability to not only read and modify all messages but also generate messages on the secure communication channel. Phishing websites also provide attackers with similar capabilities leading to instances of identity thefts, whereby bank details, passwords and last four digits of an SSN are compromised. The threat model is one of the most important design consideration and shapes the implementation of the system.

4.2 GUI

4.2.1 Detection of Phishing Websites

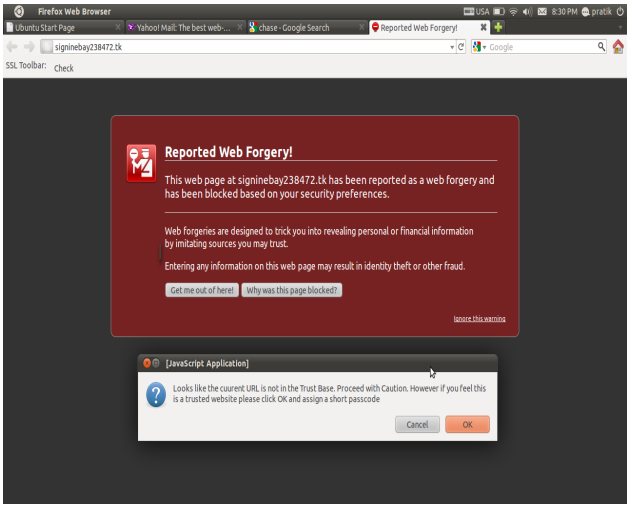

Figure 1: Phishing website dectection

We observed that phishing pages are quite similar to legitimate ones from the point of layout, however discrepancies were found in the url and structure of the document, so we thought of associating a passcode with each website which was indirectly linked to websites URL.

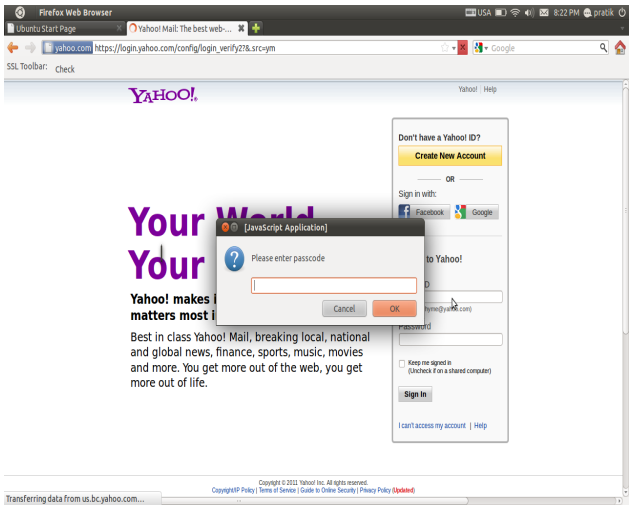

Figure 2: User enters a passcode to associate with website

In short, Suraksha does the following: (1) As soon a new website is opened, the add-on performs a check in the database of trusted URL, in case the entry is not found the user is asked to associate a pass-code with it if the user trusts the website; (2) The pass-code is displayed each time the user visits the website (can be annoying sometimes) and; (3) In case of any discrepancies like change in URL or structure, user is alerted and asked to verify the authenticity of website

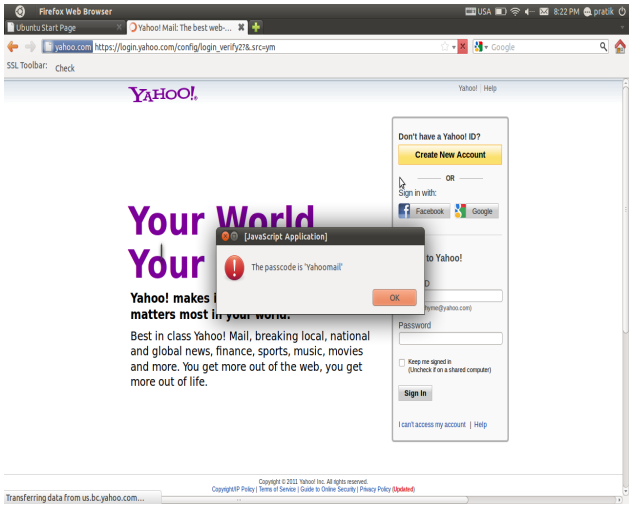

Figure 3: Displaying passcode associated with website

As seen in the Figure 1, whenever a user visits a websites not existent in the database of trusted URL's a warning is displayed to the usr for certifying the authenticity of the website. As one can see, the firefox also shows that

the website is not to be trusted. Hence, our addon detects phishing websites successfully. Figure 2 depicts that as per user input either a pass-code is entered and an entry for the website in the database or the user is redirected back to the home page. Figure 3 shows the message displayed whenever the user visits a website already present in the knowledge base.

4.2.2 Avoiding MITM attacks

The firefox browser already retains some website data, we have slightly modified it to retains more information using a SQLite Database for that purpose. Thus for each SSL protected website that a user visits the add-on caches additional data: (1) URL of the website; (2) Hash of the certificates; (3) Name of the Certificate Authority and; (4) Country of the Certificate Authority.

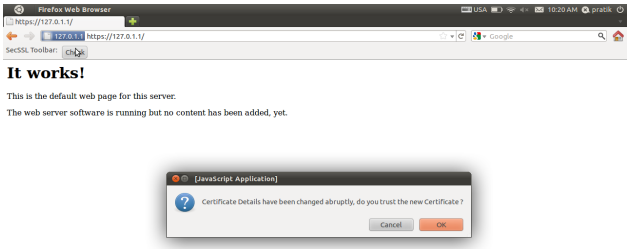

Figure 4: Alerting abrupt changes

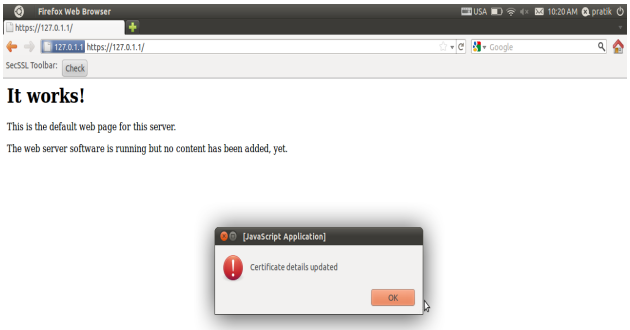

Figure 5: Certificate details updated

Whenever a user re-visits the SSL protected website, it Hash, name of CA and country of CA is compared to the already cached entry, in case of any changes the cached time-stamp is compared with new certificates details, if any abrupt change is detected like an overnight change in the name of CA or country of CA, the user is immediately notified and asked to proceed with caution. The Figure 4, displays an instance of our test bed where changes were made to the details of the self signed certificate, whereby the add-on displays a warning message asking the user to verify whether the new certificate is trustworthy or not. As per user input either the cached entry is updated or the user is redirected to the homepage. The Figure 5 shows an instance of update made to the cached entry.

4.2.3 Redirection

We have seen instances of popular websites using unsecure version of websites for transmission of secure credentials like password, credit card details etc. in order to provide faster and quicker responses.

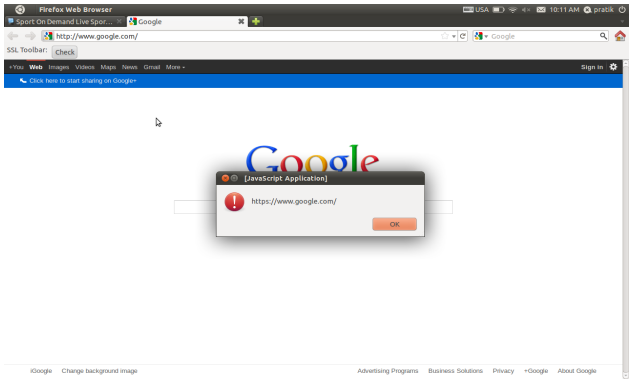

Figure 6: Redirection to secure page

These strategies lead to many indirect implications like eavesdropping, identity theft and masquerading. So the add-on extracts the URL of the system and automatically redirects the user to secure version of their websites. Figure 7 shows an instance where a user is redirected to SSL encrypted version of website i.e. Gmail precisely.

5 Implementation

We have implemented Suraksha as an add-on for firefox. It comprises of approximately 200 lines of code and written in javascript with the help of freely available SQLite Database library. To store the details of the certificate SQLite, a lightweight add-on for firefox was utilized.

The initial version of the add-on includes a toolbar with a check button to monitor the status of SSL Certificates and connection. The add-on has a simple user interface which displays alert messages to prompt the user

about any discrepancies encountered. The database first-db mainly consists of two tables: (1) Cert-tracker, which keep tracks of certificates of various websites, their associated sha1 and md5 fingerprint, name of the issuer i.e. Certificate authority and; (2) Website-tracker which stores the pass code associated with each website. These databases are used to monitor and maintain the data associated with the add-on.

The certificate details were extracted using ”gBrowser” element which has a property ”securityUI” that implements ”nsISSLStatusProvider” which allows us to get ”nsISSLStatus”. This property contains ”nsIX509Cert” which has all the necessary certificate information. We first establish a connection to the database system and create a customized database in the profile folder maintaining data across multiple sessions.

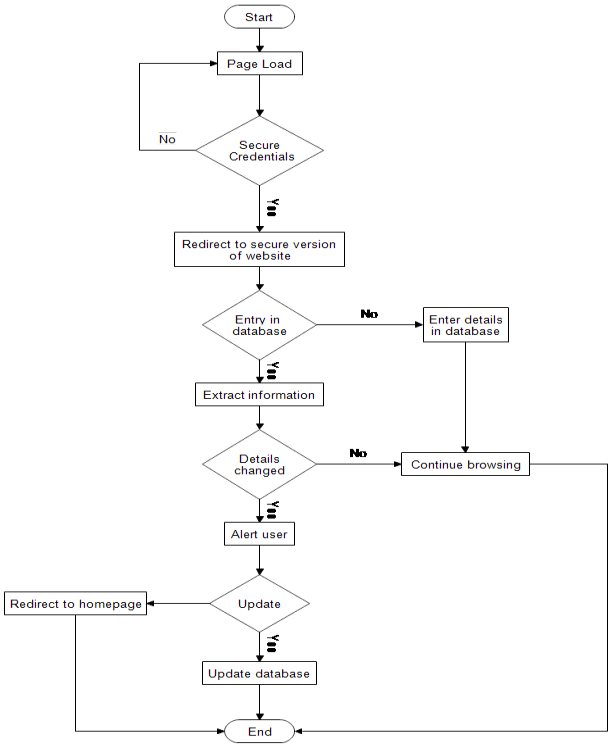

Figure 7: Flow chart for Certificate detection and Unsercure connection module

Each time the user visits a website, its certificate is monitored and any changes in hash, issuer name, name or country of certificate authority are alerted to the user. The user depending on his usage is given an authority to either add an exception or reject it, thereby redirecting him to the home page. The user is also notified in case his/her credentials are about to be transmitted in a un-secure manner and also provided redirection to secure version of websites. Figure 8 shows the flow chart about the working of the change in certificate detection and unsercure connection.

For implementing the phishing detection functionality, we created a table, using SQLite Manager, which stores websites URL and the passcode as associated by the user. Initially, the database is initialized with some of the most used banks and emails services according to the ALEXA ranking. To detect the current URL of the page the user is browsing, we used window.content.location.href property and stored it in the database. Whenever, a user visits a website whose passcode is not there in the database, the add-on will ask user to associate a small passcode with the website. Hence, whenever a user revisits the same website, he will see his passcode along with the website, which will help him in recognizing the authenticity of the website. Figure below shows the flow chart about the working of the phishing detection module.

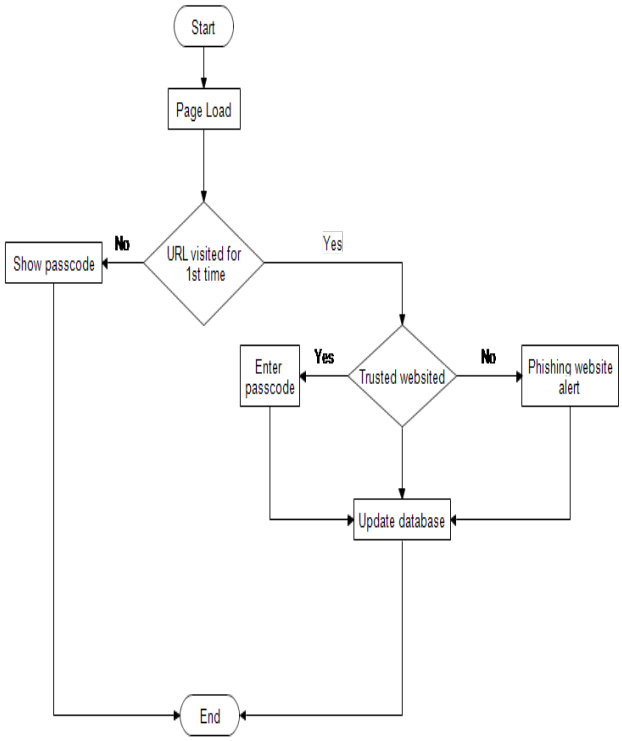

Figure 8: Flow chart for phishing detection module

As the passcode is associated given by the user, it acts as a way of uniquely identifying the website. Whenever a user visits a website whose information is not present in the database, the addon will ask the user about whether he trust the websites authenticity. If so, he will be asked to enter passcode. Thus, that website will be enlisted in the trusted knowledge-base database and upon subsequent visit to that website, the user will see a passcode associated with it. Seeing passcode associated with a website, gives user a confidence that the website he or she is visiting is indeed genuine and had also been used previously.

6 Evaluation

In order to evaluate our firefox addon, we conducted several experiments. We tested our phishing detection module in terms of false positives and true positives and their effectiveness in preventing phishing attacks. We evaluated the working of our encryption detection module by visiting few websites that uses unsecured connection. Next, we evaluated detection of abrupt changes in websites certificate by creating a custom made certificate and detecting changes in it.

6.1 The addon helps in detecting phishing

In order to prove our hypothesis, we performed two experiments to check whether our addon detects phishing websites. As a default, all the URLs present in the database are considered safe and the ones that are not in the database to be either phishing websites or false positives.

6.1.1 Experiment 1: Detecting Phishing websites with legit websites URL in database

In this experiment, we evaluated the effectiveness of the phishing detection module by testing our addon on various websites. In order to test our addon, we visited a few websites, with their URLs initialized in database, to check whether it alerts user about the phishing attack. As for the test bed, the database we used contains login URLs of top 10 legit and highly used email and bank websites. Upon entering the passcode, if website was visited for the first time, we were able to display passcode in 95 % websites successfully. The only website that didnt display passcode was because it used dynamic login page instead of static one. The dynamic login page created a URL based on the session and hence the addon could not find that URL in the database. This gave a false positive result. However, most of the websites uses static webpage for login, so our addon should successfully detect and alert about authenticity of the website.

6.1.2 Experiment 2: Detecting true positives

In order to detect true positives, we visited a phishing website that had its legit login URL in the database with its passcode associated. Even though both the websites looked exactly the same, the add-on helped in detecting that the website is a phishing website as its URL was not found in the database. As we had already initialized passcode for the said website, upon visiting the website and not displaying passcode raises suspicion and thereby it concludes that the website is indeed a phishing website.

6.2 The addon places less burden on user, as the database content increases

As initially, the database is loaded with a few websites, the user has to interact and enter passcodes. He has to filter websites depending on his own knowledge to classify websites as either phishing or genuine. When a user visits a website whose URL is not present in the database, the add-on alerts user to proceed with caution as it may or may not be a phishing website. However, as user visits different websites and filters those websites regarding phishing and legitimate, he is implicitly making the knowledge-base more powerful, with more entries of phishing and legitimate website.

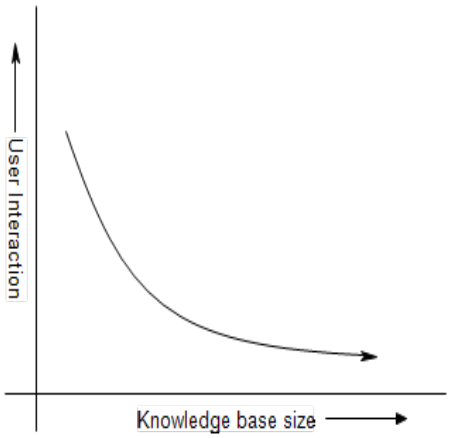

Figure 9: User interaction

As the knowledge base increases, user involvement in the learning process decreases and thus enforcing fewer burdens on the user side, providing smoother browsing environment. Thus, initially the user interaction is more and as user visits different websites, increasing knowledge base, thereby the addon will automatically alert user of the phishing website.

6.3 The addon is effective and easy to use

Whenver a user loads a new Page or starts a new Tab, our addon automatically does all the work to detect phishing attack without requiring user interaction. We have implemented our addon to detect phishing attack on the basis of information stored in the passcode associated with the websites that the user visits. Displaying a passcode of the website when a user visits it, enables user to identify the website as he had used it previously. The identity (passcode) gives confidence to the user that the website he visited is indeed genuine.

6.4 The add-on will detect changes in the certificates

OpenSSL was used for creation of self signed certificates, which is an open source implementation of SSL and TLS protocols. We performed the following three experiments to evaluate the effectiveness and efficiency of our prototype. In the first experiment, self signed certificate was temporarily imported to the certificate repository and any changes made to it were evaluated. Second experiment which is based on redirecting users to secure websites for providing credentials used facebook.com as a test bed. The performance evaluation was based on the following parameters: (1) User interactivity (minimum interaction on the part of the user) and; (2) False positives (reducing the amount of false alarms)

For testing purpose we used Self signed certificates associated with localhost which were successfully imported to firefox repository, then some abrupt changes were made to the certificate details i.e. changes to the hash, country of the CA or the name of the CA. Such abrupt changes to the certificate was detected by the add-on thereby alerting the user of the possibility of an attack and asking the user to proceed with caution. This functionality however requires the user to interact with the system using a simple button click. The add-on however suffers from the drawback of false positives i.e. whenever a website intentionally changes its certificates, it will display un-necessary warnings which may scare the users. There are cases of expiration of old certificates, certificates are revoked due to loss of private key of servers giving rise to a lot of false positives.

6.5 The add-on will alert user of unsecure connection on websites which require entering of secure credentials for his authentication

Many servers like facebook, google tend to use un-secure connections for communication purpose in order to provide faster and efficient browsing, thereby putting user credentials at the risk of being eavesdropped. The add-on will redirect the users automatically to the secure version of websites thereby ensuring that user credentials are transmitted in a secure manner. The add-on however will require user interaction and also some cases of false positives where no such websites exists that utilize SSL Certificates for communication purpose.

7 Limitations

There are are several limitations of Suraksha. The very first limitation is that this addon is only applicable to banks and email websites. Because of time constraint, we have implemented Suraksha to address only these two domains as they are highly attacked than any other domains. Suraksha works more like a whitelist filtering addon where initially the URLs contained in the database are considered safe and all others URL as either phishing websites or false positives. The user has to manually decide the URLs which are not in the database to be either phishing website or genuine. Another limitation of Suraksha is its ability to detect phishing website when visiting websites that have dynamic login page. This dynamic login page had URL which is a combination of site domain and session ID associated with that connection. As the URL will be different at different times, the addon will give false positive and detect the website as a untrusted and alerts the user to proceed with caution.

There is one more limitation that can be applied to Suraksha is the limitation on the amount of content initialized in the database (user knowledge base). As new sites keep emerging up every now and then, it becomes difficult to track of all such banks and emails websites which are genuine.

Suraksha protects user from phishing attacks user in part database and in part user knowledge and instinct. Hence, if some websites URL is not present in the database, the addon will rely on the users ability to categorize this URL as either phishing or safe website. If the user visits a genuine website and catergorized it as phishing, every time user visits the website, the addon will alert user that the website is phishing. Moreover, if the user visits a website that is indeed phishing and user categorizes that as genuine by assigning passcode then our addon will mark this phishing website as safe.

Also, this addon works on the ability of the user to recall the passcode that he or she had associated with the login URL. If the user forgets about being visited a website and unknowingly adds the phishing URL to the database, then the add-on will not help to protect against the phishing attack. However, people hardly use more than a couple of different banks and email services, hence remembering the site and passcode for which user had visited previously would not be a problem.

The biggest limitation faced when designing a detection system is to limit the number of false positives, since there are many instance of websites changing certificates either due revocation or expiration. In such cases the addon does fail and such false alarms sometimes tend to desensitize the users. Another minor limitation is that user has to interact with the system to gain its benefits, which will be like placing too much burden on an average user who is totally clueless about what an SSL or CA is?

8 Future Work

In future, we plan to extend the phishing website detection capability to all banks and email services. We also plan to add detection of fake social networking websites. In order to maintain a more comprehensive list of trusted website, we plan to implement a server that collects details from all users using the addon and analyze the websites. This analyzed database containing trusted URLs will then be added to users knowledge base, thereby decreasing false positive.

Also, design of the addon is very important as the user might become a victim if he or she doesnt understand what the addon is trying to communicate. Hence in the future, we plan to work on improving User Interface to make it visually more appealing. We also plan to add audio cues to alert user in a soothing way instead of using annoying pop-ups.

Currently our addon displays strings as passcode. In future, we plan to implement pictures and audio cues instead of plain text passcodes. Using pictures and audio cues will indeed lessen the burden on the user side and will make the addon more easy to use. It will also be more convenient to elderly and disabled people and the requirement to memorize the passcode would not be required. Also it is easy to identify from pictures and audios in compared to plain text and this feature would enhance user experience. Sometimes, it becomes annoying for the user to see pop-ups every now and then. Hence, in future we plan port our addon to using AJAX and flash to make user interaction minimal.

The add-on to some extent requires user interaction which may deter him from using our system. The entire process should be carried out transparently with minimal interaction from the user side. So the design can further be enhanced by reducing the extent of user interaction and number of false positives. We also plan to add a small feature that will allow users to submit potentially fake or suspicious certificates to a central server which will periodically flash updates to the various browsers. The add-on Suraksha is not a full fledged solution to MITM attacks caused due to compelled certificate creation, enhancing the circle of trust can further help in achieving much more effective solution. The add-on can at the same be ported to smartphones using Mozilla firefox for browsing purpose since the future lies in the portability provided by them.

9 Conclusion

In this paper, we presented a firefox addon called suraksha to protect users from phishing websites, man in the middle attack and un-encrypted transfer of credentials. As web security is very important with the facilities to use crefit cards online, it is important to protect the users data. We also discussed the design and implementation of Suraksha by evaluating in terms of detecting phishing websites, unencrypted connections and abrupt changes in the websites security certificate. According to our result, we are able to detect phishing websites with 95 % accuracy if the legit websites URL is in the trust knowledge base (database). Our addon also helps in detecting abrupt changes in the hash of Certificate, CA, Country of issuing CA. Moreover, our addon also detects unencrypted connections to website and automatically redirects the user to websites login URL which uses encryption before sending username and password. Using passcode. Thus, our addon will definitely help users in detecting and preventing frauds online by alerting users about authenticity of the website.

References

[1] Monkeysphere. `web.monkeysphere.info`, 2010.

[2] M. Anka. Ssl blacklist 4.0. `www.codefromthe70s.org/sslblacklist.aspx`, January 2010.

[3] P. S. Bance. Ssl: Whom do you trust? April 2005.

[4] F. Callegati, W. Cerroni, and M. Ramilli. Man-in-the-middle attack to the https protocol. *Security Privacy, IEEE*, 7(1):78 –81, Jan.Feb. 2009.

[5] Y. Cao, W. Han, and Y. Le. Anti-phishing based on automated individual white-list. In *Proceedings of the 4th ACM workshop on Digital identity management*, DIM '08, pages 51–60. ACM, 2008.

[6] M. Chandrasekaran, R. Chinchani, and S. Upadhyaya. Phoney: Mimicking user response to detect phishing attacks. In *Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks*, WOWMOM '06, pages 668–672. IEEE Computer Society, 2006.

[7] J. Chen and C. Guo. Online detection and prevention of phishing attacks. In *Communications and Networking in China, 2006. ChinaCom '06. First International Conference on*, pages 1 –7, oct. 2006.

[8] N. Chou, R. Ledesma, Y. Teraguchi, and J. C. Mitchell. Client-side defense against web-based identity theft. 2004.

[9] T. Close. Petname tool. `www.waterken.com/user/PetnameTool`, 2005.

[10] B. Crispo and T. M. A. Lomas. A certification scheme for electronic commerce. In *Proceedings of the International Workshop on Security Protocols*, pages 19–32. Springer-Verlag, 1997.

[11] R. Dhamija and J. D. Tygar. The battle against phishing: Dynamic security skins. In *Proceedings of the 2005 symposium on Usable privacy and security*, SOUPS '05, pages 77–88. ACM, 2005.

[12] R. Dhamija, J. D. Tygar, and M. Hearst. Why phishing works. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, CHI '06, pages 581–590. ACM, 2006.

[13] M. Dunlop, S. Groat, and D. Shelly. Goldphish: Using images for content-based phishing analysis. In *Internet Monitoring and Protection (ICIMP), 2010 Fifth International Conference on*, pages 123 –128, may 2010.

[14] Ebay. ebay toolbar and account guard. `http://pages.ebay.com/help/account/securing-account.html`.

[15] K. Engert. Conspiracy — a mozilla firefox extension. `www.kuix.de/conspiracy`, March 2010.

[16] E. Gerck. Overview of certification systems: X.509, ca, pgp and skip. `1999.www.blackhat.com/presentations/bh-dc-09/Marlinspike/BlackHat-DC-09-Marlinspike-Defeating-SSL.pdf`, 1999.

[17] D. K. Gillmor. Technical architecture shapes social structure: an example from the real world. `lair.fifthhorseman.net/~dkg/tls-centralization`, February 2007.

[18] M. Hara, A. Yamada, and Y. Miyake. Visual similarity-based phishing detection without victim site information. In *Computational Intelligence in Cyber Security, 2009. CICS '09. IEEE Symposium on*, pages 30 –36, 30 2009-april 2 2009.

[19] C. Herley. So long, and no thanks for the externalities: the rational rejection of security advice by users. In *Proceedings of the 2009 workshop on New security paradigms workshop*, NSPW '09, pages 133–144. ACM, 2009.

[20] A. Herzberg and A. Jbara. Security and identification indicators for browsers against spoofing and phishing attacks. *ACM Trans. Internet Technol.*, 8:16:1–16:36, October 2008.

[21] C. Inc. Cloudmark desktop one. `http://www.cloudmarkdesktop.com/en/home`, November 2006.

[22] C. Jackson and A. Barth. Forcehttps: protecting high-security web sites from network attacks. In *Proceeding of the 17th international conference on World Wide Web*, WWW '08, pages 525–534, New York, NY, USA, 2008. ACM.

[23] C. J. Je Hodges and A. Barth. Strict transport security. `lists.w3.org/Archives/Public/www-archive/2009Dec/att-0048/draft-hodges-strict-transport-sec-06.plain.html`, December 2009.

[24] Y. Joshi, S. Saklikar, D. Das, and S. Saha. Phishguard: A browser plug-in for protection from phishing. In *Internet Multimedia Services Architecture and Applications, 2008. IMSAA 2008. 2nd International Conference on*, pages 1 –6, dec. 2008.

[25] Y.-G. Kim, S. Cho, J.-S. Lee, M.-S. Lee, I. H. Kim, and S. H. Kim. Method for evaluating the security risk of a website against phishing attacks. In *Proceedings of the IEEE ISI 2008 PAISI, PACCF, and SOCO international workshops on Intelligence and Security Informatics*, PAISI, PACCF and SOCO '08, pages 21–31. Springer-Verlag, 2008.

[26] P. Kumaraguru, J. Cranshaw, A. Acquisti, L. Cranor, J. Hong, M. A. Blair, and T. Pham. School of phish: a real-world evaluation of anti-phishing training. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, SOUPS '09, pages 3:1–3:12. ACM, 2009.

[27] P. S. LLC. Image based web-site authentication, July.

[28] N. Ltd. Netcraft anti-phishing toolbar. `http://toolbar.netcraft.com`, November 2006.

[29] G. Megaw and S. Flowerday. Phishing within e-commerce: A trust and confidence game. In *Information Security for South Africa (ISSA), 2010*, pages 1 –8, aug. 2010.

[30] N. Nikiforakis, Y. Younan, and W. Joosen. Hproxy: client-side detection of ssl stripping attacks. In *Proceedings of the 7th international conference on Detection of intrusions and malware, and vulnerability assessment*, DIMVA'10, pages 200–218, Berlin, Heidelberg, 2010. Springer-Verlag.

[31] T. Ronda, S. Saroiu, and A. Wolman. Itrustpage: a user-assisted anti-phishing tool. In *Proceedings*

of the 3rd ACM SIGOPS/EuroSys European Conference on Computer Systems 2008, Eurosys '08, pages 261–272. ACM, 2008.

[32] A. P. E. Rosiello, E. Kirda, C. Kruegel, and F. Ferrandi. A layout-similarity-based approach for detecting phishing pages. In *Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007. Third International Conference on*, pages 454 –463, sept. 2007.

[33] J. Saltzer and M. Schroeder. The protection of information in computer systems. *Proceedings of the IEEE*, 63:1278 – 1308, sept. 1975.

[34] S. E. Schechter, R. Dhamija, A. Ozment, and I. Fischer. The emperor's new security indicators. In *Proceedings of the 2007 IEEE Symposium on Security and Privacy*, SP '07, pages 51–65. IEEE Computer Society, 2007.

[35] C. Soghoian and S. Stamm. Certified lies: Detecting and defeating government interception attacks against ssl. April 2010.

[36] M. Stevens, A. Sotirov, J. Appelbaum, A. Lenstra, D. Molnar, D. A. Osvik, and B. Weger. Short chosen-prefix collisions for md5 and the creation of a rogue ca certificate. In *Proceedings of the 29th Annual International Cryptology Conference on Advances in Cryptology*, pages 55–69. Springer-Verlag, 2009.

[37] J. Sunshine, S. Egelman, H. Almuhimedi, N. Atri, and L. F. Cranor. Crying wolf: an empirical study of ssl warning effectiveness. In *Proceedings of the 18th conference on USENIX security symposium*, SSYM'09, pages 399–416. USENIX Association, 2009.

[38] M. Wu, R. C. Miller, and S. L. Garfinkel. Do security toolbars actually prevent phishing attacks? In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, CHI '06, pages 601–610. ACM, 2006.

[39] M. Wu, R. C. Miller, and G. Little. Web wallet: preventing phishing attacks by revealing user intentions. In *Proceedings of the second symposium on Usable privacy and security*, SOUPS '06, pages 102–113. ACM, 2006.

[40] W. Yu, S. Nargundkar, and N. Tiruthani. Phishcatch - a phishing detection tool. In *Computer Software and Applications Conference, 2009. COMPSAC '09. 33rd Annual IEEE International*, volume 2, pages 451 –456, july 2009.

[41] Y. Zhang, J. I. Hong, and L. F. Cranor. Cantina: a content-based approach to detecting phishing web sites. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 639–648. ACM, 2007.