ISSN : 2231-0711

Priyam Kaur Sandhu et al | IJCSET(www.ijcset.net) | July 2015 | Vol 5, Issue 7,283-287

Google Safe Browsing – Web Security

Priyam Kaur Sandhu

priyamsandhu@gmail.com
Chandigarh, India

Sanjam Singla

sanjamsingla@gmail.com
Chandigarh, India

***Abstract*— Google safe browsing is the new shimmer in the market to safeguard the web browsers. Now a days, the web browsers such as Internet Explorer, Mozilla Firefox and Google Chrome are installed on almost all the computers and are used so frequently that their configuration should be done securely. The plug-in extensions are the lucrative vectors for the malwares. The lack in the security mechanism leads to intrusion of malwares in the systems. So it becomes imperative to configure the web browser for safer internet security. A List of URLs is provided by Google that consists of phishing and malware data. This list is used by various web browsers to check the pages against potential threats. The introduction of the Safe Browsing is a life saviour step towards securing the internet surf by the users. This paper gives an overview about how Google provides web security by providing various checklists and focuses on, how a particular site can be analysed in order to mitigate the threat.**

***Keywords*— Google safe browsing; Malwares; Phishing; API; Security**

I. INTRODUCTION

The most frequently used web based application, i.e. web browser is utilized in almost all the fields. It offers an interface to the users to perform wide range of activities like, sending/ receiving emails, net surfing, finance management, social engineering, online shopping and professional business. Therefore, a large number of applications require a safe means to operate the data. All the important and sensitive data of the users like usernames, passwords are open and not safe even after implementing encryption mechanisms on the movable data traffic, which leads to breach of authenticity, confidentiality and integrity. The vulnerability of the web browsers are the targeted spots by the hackers for software attacks. The use of malicious web sites exploits the web browsers [1] [2]. There are a number of factors those lead to the intrusion, like, the users have a habit to visit the links without prior knowledge about the site, a large number of web browsers do not focus on security and tend to enhance the functionality, various malicious web page addresses direct the user to the infected sites, etc. As a consequence, the attackers succeed in compromising the computer systems by exploiting the vulnerabilities in the web browsers [9]. The attacker mainly tries to exploit the client-side systems through various vulnerabilities to gain access to the sensitive information. These vulnerabilities are used in a way to take system control, destroy files, modify data, steal data and to attack other systems. The associated risks with less secure environment are: Cross zone vulnerabilities, Cross site scripting and Detection evasion. So, these conditions arise the need to secure the system by configuring the web browsers and disabling the vulnerability causing features. Google Safe Browsing is one of the techniques to deal with the malicious websites. It prevents the harmful sites from opening or stealing personal data. It is an extremely handy tool that keeps a check on the malicious content, links and websites those are being circulated in the vast web of internet. In this process, the list with the local server is compared with the inquired URL; if it is a malicious content then the site is restricted from opening [6].

II. WEB BROWSER ATTACKS

This section gives a brief overview about different web browser attacks that can harm the system.

A. Phishing

Phishing is a process [7] in which the attacker tries to acquire the personal data such as ids, passwords, usernames, details of credit card, etc., by representing as a legitimate entity. The communications pretending to be coming from banks, online payment sites, auction sites and social websites are often used to tempt the people. Phishing emails are one of the widely used tools by the hackers that contain the links to the infected websites. Phishing can be conducted by email spoofing or instant messaging. The websites direct the users to fill in the details in this fake website whose look is similar to the original website. Phishing can be viewed as social engineering techniques that are used to trick the users and exploit the web security technologies. The phishing attack is directly on the money, this trend can be seen from the continuous targeted attacks on the eBay and PayPal websites. As the time is passing, these attacks are getting more refined, studied and creative. The phishing webpage is available for a time period of less than an hour to dodge its presence and exposure on Internet. The various phishing techniques are listed below:

1) *Phishing:* It is a technique in which the data such as passwords, credit card details and IDs are extracted from the users by impersonating a legitimate entity.

2) *Whaling:* The attacks targeted on senior executives and other high profile in businesses is called whaling.

3) *Link Manipulation:* In this technique, the URLs are manipulated by use of sub domains and misspelling the words to trick the target. The technical designs are brought into use to create the links in an email and in infected sites.

4) *Filter Evasion:* In this particular technique, the text is replaced by malicious image in order to conceal their presence from anti-phishing filters.

5) *Clone Phishing:* In this attack, the originally delivered email is brought into use; the information present in the email acquired by the hacker is used to make a fake copy of the original email. The spoofed email now contains the malicious data and is sent to the destination email address in a way that it appears to have come from the original sender.

6) *Search Engine Phishing:* In this type of phishing, the hackers make the website appear more attractive with the help of sounds, videos and images and index them legitimately with a search engine. The user gets in contact with the phished content in normal searching course and gets trapped and ends up in giving the personal information.

7) *Content Injection Phishing:* In this type of phishing, the hackers replace a small portion of legitimate content with fake content which is designed to extract the confidential information.

8) *Spear Phishing:* In this type of phishing technique, the target is spotted on a specific person or a company. The hackers get knowledge about the targets and use their information to exploit them.

B. Malware

Malware is software [8] that interrupts the operations of the computer, retrieves the personal data and gains control over the user's system. Malware can be of various forms such as a code, active content, scripts, etc. Malware contains computer worms, root kits, dialers, viruses, Trojan horses, key loggers, spyware, etc. The most popular threats are worms and Trojans instead of viruses. Malware websites may harm the visitors in the following ways: Compromised websites which are legitimate are used to attract the user and deliver distribute malwares on the user system. After the legitimate website gets compromised, some content is added from an attack site which contains the links that redirects to an infected site. These infected sites initiate the drive by download. The drive by download leads to execution of a malicious program on the computer without user's consent. It has catastrophic effects as the spyware gathers data such as bank credentials and uses the system to send spam. The main aim of these attack sites is to dissimilate malware and to avoid detection through various techniques, like changing their hosting locations very frequently or by generating new domain names automatically [10]. This section presents different classes of malwares which have been encountered over the period of time. There are various methods of categorizing malwares such as:

1) *Polymorphic Malwares:* Changes itself each time it runs, but the function of the code does not change.

2) *Metamorphic Malwares:* Rewritten when executed so that each succeeding version of the code is different from the preceding one.

Various other well known malwares are:

3) *Worm:* These are the malwares which does not require any host to propagate from one system to another. These malwares can run independently and can infect many systems without requiring any host. The first publicly known worm is Morris worm.

4) *Virus:* In contrast to worms, these are malwares which cannot run independently and these require a host to propagate from one system to another. These malwares attach themselves to the files and infect the system.

5) *Trojan Horse:* This type of malware appears to be legitimate but at the backend it performs malicious actions to gain control over the infected system.

6) *Spyware:* As the name suggests, it spies the infected system and gets control over the infected system, moreover steals the sensitive information from the user system.

7) *Bot:* It is a malware which allows the malware author to gain control over the infected system sitting at a remote region. If more than one infected system is controlled remotely by the author then the set of those systems is known as Botnets.

8) *Rootkit:* These are the type of attacks which hide their presence from the users and gain control over the information and system without user's consent.

III. GOOGLE SAFE BROWSING

Google safe browsing [5] is basically a service that is being provided by Google to enhance the security of web browsing. It has emerged as one of the most robust security implementation techniques to protect against the cyber attacks like phishing, malware and unwanted downloads. Google safe browsing is a kind of storage house of all the information regarding malicious activities. It consists of a record of all the URLs of malware and phishing sites. The Internet Service Providers are also provided with the information by Google regarding threats hosted on their networks by sending e-mail alerts to Autonomous System operators. Google Safe Browsing inspects a large number of URLs on daily basis in search of infected websites. The unsafe websites comprises of large number of fake sites and the legitimate websites those have been compromised by the intruder. In this process, when an unsafe website is identified, warnings are issued to the user in the browser. The infected site could be a phishing site or a malware site. In the phishing sites, the user is tricked into providing his personal details as the site pretends to be legitimate. In such cases, the high security details like bank passwords, personal id numbers get leaked out that leads to catastrophic results. In case of malware sites, the code is used to inject infected files in the user's computer from where the hackers steal the information present in the computer. Safe Browsing takes different actions in different scenarios to safeguard the web searching. If suppose the hyperlink present on the current page of the user' browser is the one that would take the user to an unsafe site, then safe browsing issues a warning to the user before clicking it.

Safe browsing performs the task of matching and checking the site's name with the one's present in the list of Google's malicious suspected websites. It also restrains the users to post the links from current site to the suspected malicious websites. The early versions which were released for Google safe browsing were version 1 and version 2. The brief description about the two versions is given in the Table 1.

As it is seen from Table 1, the two versions suffered a number of drawbacks. So, a new version that is version 3 was introduced to overcome the ongoing problems. A brief comparison of the three versions is given in Table 2.

From Table 2, it is observed that over the period of time Google has improved the web browser security as we can see a noticeable difference between the different versions.

In the current scenario, Safe Browsing API v3 is the latest version available for safe browsing, the Safe Browsing API v2 has been deprecated and v1 has been discontinued due to lack of security. However, there is another type of browsing API known as Safe Browsing Lookup API which is simple to implement as the API user sends a HTTP GET or POST request with the URLs and the server responds the share of the URLs. However, there are some drawbacks like privacy and response time. As the URLs are not hashed, so the server knows about which URLs API users are looking for. Second is the high response time as every request is processed by Safe Browsing server which increases the response time [2] [3].

TABLE I
COMPARISON OF VERSION 1 AND VERSION 2

Features	**Version 1**	**Version 2**
Internet connection	Works with unreliable internet connection	Works with reliable internet connection
Latency	It is low as large number of URLs is scanned at the same time.	It is comparatively more.
Availability	It should be available all the time	Availability is not the concern
Lookups	Performs lookups offline	Performs lookups online
Match	Same latency in all the cases, match or no match	Latency differs
Up to date list	It could get a bit older in content	It gives the most up to date list.
Downloading	More data downloaded	Less data downloaded
Load on Google server	Maximizes the load	Minimizes the load

TABLE II

COMPARISON BETWEEN VERSION 1.0, 2.0 AND 3.0

Version 1	**Version 2**	**Version 3**
-The hashing algorithm used in this version is MD5. -The efficiency is less and it is not scalable. -The entire phishing list entries are to be downloaded at once because the partial list updates are not supported till the time the user fully downloads the recent version of list. -The phishing data is given to the client in an order from old to new. For the phishing sites it is inefficient as they have a short lifetime. -As regular updates are required, so the bandwidth consumption is escalated. -It is time consuming as the users scarcely find a match with the present pattern. -It is slow and has high latency.	-The hashing algorithm used in this version is SHA 256. -In the place of a single versioned list, a series of "chunks" is used. -A list of URLs is received when the list of chunks is communicated while updating. -The Chunks present are 32-bit truncated hashes -As soon as a match is discovered, the 32 bit chunk is communicated to Google and in return a list of 256 bit hash is acquired. -As compared to version 1, it is faster. -It has improved speed and reduced Latency.	-To improve the efficiency, protocol buffers are used by encoding the chunk data. -Host keys are not used. -Optional metadata was included in HTTP Response for Full-Length Hashes . -To differentiate between the kinds of sites and to allow more warnings, metadata functionality was used by Google-malware-shavar list. -For the full hash, modifications are carried out in the cashing semantics. -An expiration time is included in the HTTP Response for Full-Length Hashes in response. -When an update request is sent, each time the clients are required to wash off cashed full length hashes. -Message Authentication Code (MAC) support is eliminated. -The API key format is modified. The Google Developers Console manages the API keys.

IV. WORKING

This section briefly explains the systematic working of Google safe browsing. The Figure 1 below describes the working:

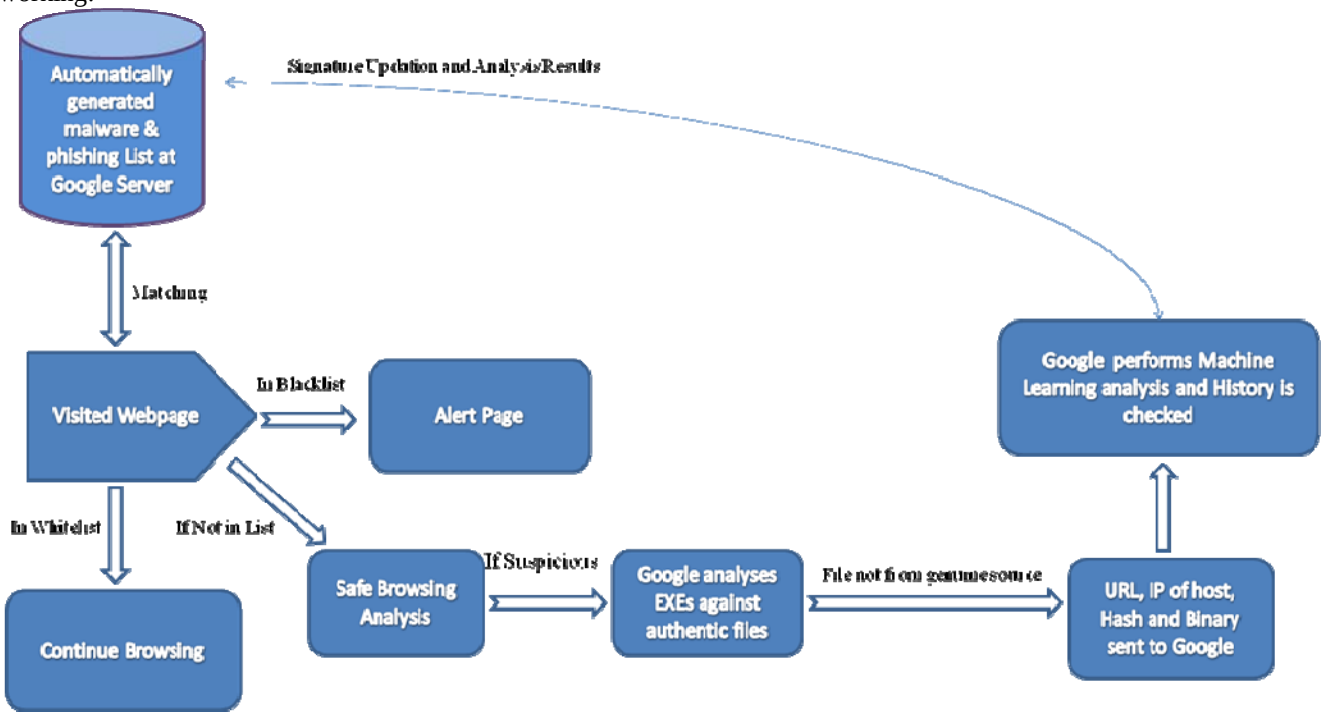

Figure 1. Working Model

The Safe Browsing feature is implemented to protect the system and the client's privacy and conserving transmission bandwidth by sending small chunks of information to and from the client's system. The Safe Browsing downloads a list of known malware and phishing websites which are regularly updated. These are produced by an automated analysis of the entire web of internet. Google presents two main lists of infected URLs: a malware list and a phishing list. Hashed values are present in these URL lists. Through the Google safe browsing API, the lists can be accessed. The Google safe browsing maintains a local database that contains the white and the black list. The complete hashes are stored locally in them. After every 30 minutes it is updated. The process adopted is as follows: Firstly, pre-filtering mechanism is carried out in which URL hostname is used initially then the full path is picked. This pre-filtered data is stored locally that is synchronized with Google after fixed interval of time. Then the hash is calculated from the URL. As soon as the URL key and the host key are spotted, the pre-filtering mechanism is carried out against the white list that is stored locally. At the moment when one of the host keys and the URL keys are discovered in the white list, it indicates that it is not a malicious URL. If the URL under scrutiny is not found in the local white list, then its presence is checked in the local black list. If it is found in the black list, then full hash of the URL is looked upon. The local database is not used for this purpose, the request is sent to sent to Google. The file undergoes machine learning analysis and the reputation and history records are checked. the Google Safe Browsing server. This is the procedure followed to detect the infected URL [5]. All the links and resources those are visited by user present on the page are checked against these lists to detect the malicious content. The checking of the URLs does not reveal the websites that are being visited. If the browser detects that the visited page is in the list, then it immediately warns the user by showing an alert page that restrains from surfing it and directs to take a safety precaution. This technique is followed if the browser is already aware that the website is infected. To improve protection and security, two mechanisms are employed those are capable of detecting phishing attacks and harmful downloads the system encounters for the rest of the time. The phishing attacks are in the active form for short duration of time only, so it is required to detect new attacks as they occur. The behaviour and features of the page that is being visited is noticed by the safe browser. This procedure is followed at the user's computer and no information is given to the Google. If the page appears to be suspicious, then the corresponding URL is send to the Google for further analysis. It is very difficult to detect malicious downloads as the URL are very swiftly changed. Sometimes these are repacked to bypass the antivirus. To counter this behaviour, the executable downloads are checked against a list of authentic files [4]. If the file is not from a genuine source, then the URL and IP of the host, such as the hash of the file and binary size will be The results are then sent to the browser and the user is warned for the actions.

V. CONCLUSIONS

In the present era, Google Safe Browsing has emerged as the well known safeguard for the web browsers to provide high level security to the client. The attackers use various means to send the malicious content to the client's device to compromise the security system and intercept user's sensitive information. The Google Safe Browsing has the propensity to list down all the links of malicious websites and identify it before opening in order to discover and obstruct the attacks those are concealed under legitimate appearing websites. On comparing the three versions, currently Safe Browsing version v3 is adopted by the organizations as it meets the security policies required by the users in present world, whereas on the other hand Safe Browsing API v2 has been deprecated and v1 has been discontinued due to lack of security. Having a look on the future shine of the Google Safe Browsing, it needs to be more sophisticated as every day new attacks are born that adversely affects the clients. Still a large number of malicious websites are being passed by this barrier unnoticed due to high skills of the hackers. More techniques need to be implied on how to recognize the malicious URLs and restrict them from stealing data.

REFERENCES

[1] http://dev.chromium.org/developers/
[2] https://developers.google.com/safe-browsing/
[3] https://developers.google.com/safe-browsing/developers_guide_v3
[4] http://research.zscaler.com/
[5] Julien Sobrier, [Whitepaper]: Google Safe Browsing v2 API: implementation notes Technical information about using Google Safe Browsing v2, Zscaler, 17 january, 2011, Version 1.1
[6] Joshua Drake, Paul Mehta, Charlie Miller, Shawn Moyer, Ryan Smith and Chris Valasek, "Browser Security Comparison – A Quantitative Approach", 12 June, 2011, Version 0.0
[7] Ter Louw Mike Jin Soon Lim and V. N. Venkatakrishnan, August 2008, "Enhancing web browser security against malware extensions", *Journal in Computer Virology*, Springer, Chicago, Illinois, USA, Vol. 4, pp.179-195.
[8] Ter Louw Mike, Jin Soon Lim, and V. N. Venkatakrishnan, "Extensible web browser security, *Detection of Intrusions and Malware, and Vulnerability Assessment*", Springer Berlin Heidelberg, July 12-13, 2007. pp. 1-19.
[9] Holzammer, Andreas, "Security Issues about Web Browser Add-ons." Seminar Internet Sicherheit, Technisc-he Universit at Berlin, 2008.
[10] Van Dongen, Wouter S. "Browser security." (2009).