A SURVEY REPORT ON THE ANALYSIS OF SESSION HIJACKING & MALICIOUS CODE ATTACK OF SOCIAL NETWORKING SITES

Amitesh Jaiswal [1], Swati Jaiswal [2], Yogesh Dangi [3]
[1]Capgemini, Mumbai, (India)
[2]Computer Engineering, SKNSITS, Lonavala, (India)
[3]TCS, Mumbai, (India)

ABSTRACT

Over the last few years millions of people worldwide are communicated through Online Social networking sites. A social network services is an online services, or sites that focuses on building and reflecting of social relations among peoples. The proliferation of social and collaborative media has been accompanied by an increased level of cyber attacks on social networking and collaboration sites.A social network service essentially consist information of each user called a profile, his/her social relationships, events and a variety of services. Recent years have seen exceptional growth in the usage of online social networks and there are about 300 online social networks such as Facebook, Twitter, and MySpace, having more than half a billion registered users worldwide. Users can often upload pictures of themselves to their profiles, post blog entries for others to read, share thoughts and ideas, search for other users and compile and share lists of contacts. The information they share are not only viewed by the trusted users but also by the third party users whose intention is to make harm to individual users. Another serious problem is session attack known as session hijacking. This attack enables the attacker to impersonate the victim and take over all his/her complete networking sessions. This paper provides the information about attacks and their solutions in the areas of security and privacy applicable to social networks.

Keywords*: *Cyber Attacks, Facebook, Privacy

I. INTRODUCTION

The social networking site becomes the most essential activity for the users. For them it plays a vital role for creating social relationships between numbers of users. The social networks, as process massive amounts of personal data on individual behavior, probably lead to the erosion of civil liberties through loss of privacy and personal freedom. Moreover, identity theft invites malicious acts such as phishing, spamming, and Sybil attacks. These negative developments threaten to undermine the potential and beneficial opportunities of Internet. Securing the future Internet becomes a concern of users' community. The real benefits of online social networking should not be lost while working for new security standards. It is like venturing into careful handling of a double-edged sword. The growth of social media and the increase in the number of users of social networking sites (SNSs) in the past few years are mind-boggling. Initially, social media has been used by ordinary people just for connecting with friends and for making new friends. A large population of people worldwide are now acclimated to social networking and the use of modern technology (e.g., smart phones, tablets) to communicate with friends and co-workers. Social media has recently started taking important role in business as well. Companies have started using social media websites such as Twitter and Facebook for doing marketing, market research and customer support. The proliferation of social media has, however, been accompanied by a similar level of growth in cyber attacks on social networking sites. In addition to phishing and spamming attacks, threats to SNSs include session hijacking attacks that enable the attackers to view private photos, broadcast messages, see personal web history, and do anything else that the owner of the hijacked account can do. The threat of weak security to a SNS could hurt its adoption and scare away future users from engaging in the site any more than they already do. Friendster is the first popular social networking site (SNS) that allows people to explicitly articulate their social network, present themselves through a profile such as interests and demographics, post public testimonials about one another, and browse a network of people. Friendster's tools support a powerful process of community formation around shared values and tastes. Social groups tend to converse collectively on a coherent presentation style and encourage other participants to follow the collective norms (e.g., regarding photos). Facebook, a social networking site that began with a focus on colleges and universities, has been studied and evaluated by several works [5–7]. These studies have collected profile information from Facebook through the use of a web crawler and through surveys of members. They show that Facebook members reveal a lot of information about them, and they are not aware of privacy options or controls who can actually view their profile [5]. MySpace, the largest social networking site in the world following Friendster, mainly focuses on music and popular culture. Figure 1 depicts the popularity of top social networking sites in India. In a comparative study of Facebook and MySpace [8], it is found that Facebook has shown more reputation than MySpace.

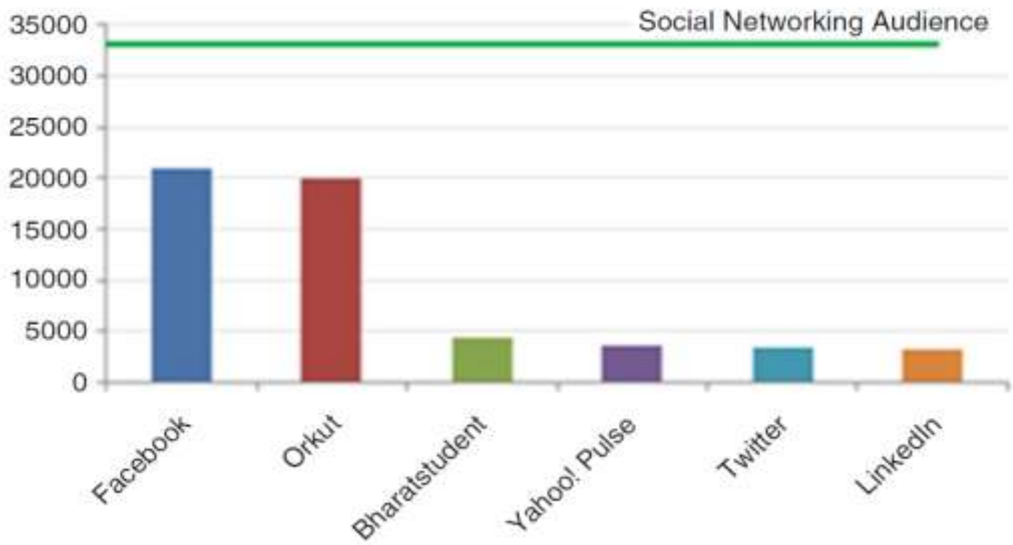

Fig-1: Graph for OSN

This paper provides information about session attack and harms caused by malicious codes.

II. RELATED WORK

Yogini A Kulkarni & Rajendra G Kaduskar [1], discussed about malicious codes. Malicious codes are codes which can change, delete, added from any software system to intentionally cause harm. Malicious codes are rapidly becoming a critical problem for business enterprise, government, colleges, industries and individuals. Once the malicious code entered into your system or network it will propagate itself which results in a disastrous way. As we all know that peoples are using facebook, twitter, LinkedIn and various other social sites very frequently. These sites attracted criminals behavior for performing illegal activities .offenders/criminals hacked facebook account and sent various illegal site link to other user that contain malicious codes. Even criminal can send tweets that contains links which reveals the user URL address. There are various other problems that may arise because of using social networking sites. In this paper, authors discussed an approach that contains the authentic web transaction issues when some user connects itself through web. For detecting malicious codes authors uses logos, flash applications, legitimate followers and various PDF's which are present in URL transaction and also provides security for the same. This paper deals with two level of security one for address translation and second for data formats. For detecting malicious codes authors presents two components one is technical component and other is an analytical component. Jeffrey Cashion & Mostafa Bassiouni [2], explained one serious type of attack known as session hijacking. This attack enables the attacker to impersonate the victim and take over all his/her complete networking sessions. Authors present a security mechanism by providing authentication protocol for overcoming the risk of session hijacking in social networking sites. The protocol is totally based on the recognition of that when users connect themselves with various platforms & other connection speeds. This approach will work for both mobile devices such as tablets and smart phones as well as this approach will also work on the pc's that are using high speed internet or Wi-Fi connection. The proposed protocol is named as Self-configuring Repeatable hash chains (SCRHC) protocol. The protocol provides different variations and it supports three different level of cashing, it allows user to forfeit storage space for increasing performance and it also reduces workload. This method provides valuable security to social networking sites in a flexible manner. Farad Ahmed & Muhammad Abulaish [3], provides a solution to spam profile and data. From the past few years, humans use online social networking sites for communication. This way of communication provides a wide range of application to access as well as it costs less. A wide range of information is being shared through popular sites. This aspects of information sharing between large numbers of users attracted spammers to exploit, network by spreading spam messages. This can be done by promoting personal blogs, phishing, advertisement, scams and so on. Authors used Markov clustering (MCL) based approach for detecting spam profiles. They use real data set of facebook profiles. Authors modeled social network using weighted graph in which profile is represented by nodes and interaction by edges. The weight of edges which connects user profiles calculated. This calculation is done on the basis of active friends, shared links and pages likes. The MCL algorithm classifies the behavior similarity to profile & determine the classes of outlier clusters. The huge amount of information available through the online social networking sites has attracted researchers to mine this information and study issues faced by the social network community. Considerable work has been done for collecting and mining the information for various problems such as community detection, information diffusion and spam filtering. In [5], the authors investigated the feasibility of using measurement calibrated graph models for sharing information among researchers without revealing private data. In [6], the authors presented a study of topological characteristics of Twitter OSN. The authors investigated the behavior of information diffusion over the Twitter network by analyzing re-tweets and found that information re-tweeted once reaches on average1000users. The authors in [7] presented a study of click stream data of social networks. Their analysis shows that the use of click-stream data provides rich information about social interactions, and that a majority of user activities on social networks consists of "browsing". Similarly in [8], the authors investigated social interactions of users on OSNs and proposed that a majority of interactions on OSN sites are latent in nature, whereas visible events occur less frequently. There has been some research for the detection and prevention of spam on OSNs. In [9], the authors proposed a real-time URL-spam detection scheme for Twitter. They logged browser activity as a URL loads in the browser and monitored a multitude of details including redirects, domains contacted while constructing a page, HTML content, pop-up windows, HTTP headers, and Java script and plug-in execution to detect spam links. Another substantial work on detection of spam on OSNs is presented in [4].

III. DATA COLLECTION

Facebook is the most popular online social network claiming 800million active users [16]. The popularity of Facebook can be associated to its platform features that make social interactions and information sharing more interactive. To develop a proof-of-concept model of the social network graph, we crawled publicly accessible Facebook data containing both normal and spam profiles. Some of the popular features considered by our data collection module are wall posts, fan pages and tags. A brief description of these features is presented in the following paragraphs.

- Wall Posts: A users' Facebook wall is a place where her friends (or everyone depending on the privacy settings) can interact by posting messages and useful links. Users can also like and comment on the wall posts. According to Facebook statistics (November 2011), in a single day about2billion wall posts are liked or commented.

- Pages: Facebook pages are designed for celebrities, business organizations, etc., that intend to share information to people outside their real social circle. Users can like certain pages to get latest updates about their interests. According to Facebook statistics (November 2011), a single user has indirect connection to larger groups of users via80(on average) community pages, groups and events.

- Tags: Facebook tagging feature allows users to tag friends and pages in posts (analogous to twitter mention). Once tagged in a post the content being shared becomes visible on the subjects' wall and hence affects information diffusion.

For Facebook profiles users' activity on his/her Facebook wall were logged. Only information available for public view was collected and users with restricted view of their profiles were not considered. We logged activities related to friendship requests, wall posts, fan page likes and links shared. We logged only the visible interactions of a profile.

IV. EFFECT ON WEB PAGE DUE TO MALICIOUS CODE

The popularity of social engineering is also demonstrated by the increase in 'scareware' programs. Such scams start with a pop-up message on a web site, which says the computer is infected and you should download a free anti-virus program to remove the malware from your system which has supposedly been found. But when you download and run the program, it tells you that you need the 'full' version in order to disinfect your computer and you have to pay for this. Of course, the cybercriminals potentially win twice with this scam: not only have they taken your money under false pretences, but they also now have your credit card details. The second Top Ten presents data generated by the web antivirus component, and reflects the online threat landscape. This ranking includes malicious programs detected on web pages and malware downloaded to victim machines from web pages as per table4. [1]

Table1: Various Attacks

Position	**Name**	**Number of attempted downloads**
1	Trojan-Downloader.JS.Gumblar.x	178965
2	Trojan.JS.Redirector.I	126277
3	Trojan-Clicker.JS.Iframe.ea	102226
4	Trojan.JS.Agent.aui	80654
5	Exploit.JS.CVE-2010-0806.i	148721
6	Trojan.HTML.Fraud.aj	68809
7	Packed.Win32.Krap.as	64329
8	Exploit.JS.CVE-2010-0806.b	50763
9	Trojan-Clicker.HTML.IFrame.fh	38266
10	Trojan-Downloader.JS.Twetti.a	46858

V. MALICIOUS CODE DETECTION

A technology used to detect malicious code has two components [1]– a technical component and an analytical component. The technical component is the sum of all functions and algorithms which provide the analytical component with data for analysis. The analytical component is analyzed. The technical component of a malware detection system collects data that will be used to analyze the situation. As any malicious program is both a file with specific content and the sum of the effects the malicious program has on the operating system, there are various methods used to collect data in order and to identify malicious code. These methods are given in the order of abstraction. The term abstraction means the point of view from which the program being run is viewed: as an original digital object (a collection of bytes), as a behavior (more abstract than the collection of bytes) or as the sum of effects on the operating system (more abstract than the behavior). Antivirus technology has, more or less, evolved along these lines: working with files, working with events via a file, working with a file via events, and working with the environment itself.

1. The first antivirus programs
2. Emulating program code
3. Virtualization
4. Monitoring system events
5. Search for system anomalies
6. Audit

For the analytical component, the sophistication of a decision making algorithm varies. They can be divided into three categories:

A. **Simple Comparison.** In this category, decision is issued based on the comparison of a single object with an available sample.

B. **Complex Comparison.** In this case a decision is issued based on the comparison of one or several objects with corresponding samples.

C. **Expert Systems.** In this category, a decision is issued after a sophisticated analysis of data. An expert system may include elements of artificial intelligence.

VI. SESSION HIJACKING

The protocol, called Rolling Code[11], utilizes the initial secure HTTPS authentication to exchange a shared secret between the server and the user browser. The shared secret consists of two components: a seed and value d, both of length 160 bits. For every transmission made from the client to the server, the client first updates the value of d by hashing it then generates a cookie code which is another hash operation applied on the XOR of seed and the updated value of d. The client sends the cookie code to the server which will perform similar steps on the shared secret stored at the server and compare the computed cookie code with the received cookie code.

Liu et al. [9] proposed a secure cookie protocol by making modifications to improve an earlier protocol proposed by K. Fu [10]. Their solution for ensuring integrity of each cookie involved embedding a username, expiration, data, and HMAC. This would inject a lot of repetitive data if there were a lot of cookies; each cookie would all have this information embedded in it. Also, it is not confidentially protecting the names of the cookies, but instead leaving them open for all to see what kinds of information is being shared. Their fundamental assumption is that their secure cookie protocol would run on top of SSL, which is an expensive protocol in terms of its computational overhead. Our protocol is based on the recognition that users of social media such as Facebook connect to their web sites using a variety of platforms. On one end, there are Wi-Fi connections with users connecting via mobile smart phones or tablets. On the other end, there are high-speed connections with users connecting via high-end PC's and workstations. We therefore employ two different authentication flavors: one for mobile devices using wireless connections and the other for high-end workstations using high-speed broadband connections. The core of the two flavors is the same, but they differ in how they exercise various aspects of the protocol. For devices of all types and connections, we modify the hash chains approach in order to overcome a known limitation regarding the need to estimate the number of transactions during the lifetime of a session. We call this modification the Self-Configuring Repeatable Hash Chains (SCRHC) Protocol. Below, we provide motivation for SCRHC then present its basic design. The hash chains approach has been used in the one-time cookies (OTC) authentication protocol [10] to prevent session hijacking. The use of a hash-chain requires the client and server to establish how many transactions they expect to do during the lifetime of the session. Such a value is expected to be estimated by the website administrator ahead of time using metrics based on usage statistics. If the number of transactions is overestimated, the authentication in the early steps will suffer from an unjustified large computational overhead. If the number of transactions is underestimated, there will be the undesirable synchronization overhead of establishing a new secret and a new number for the remaining transactions. Our protocol supports three different levels of caching, giving the user the ability to forfeit storage space for increased speed and reduced work-load. The simplest method is no caching. For each communication event, the SCRHC Step function must generate the code in its entirety from scratch. This is the most time consuming but also does not require any storage. There are situations where storage space might simply not be available or in limited supply, so we allow for this. The second level of caching is what we call selective dynamic caching. The third level of caching is full caching, where all of the hashes are computed ahead of time and stored for later reference. This requires the user to dedicate considerably more storage than either of the other two methods, but if this storage is available, it provides the fastest performance. Our technique requires much less space than OTC though, since our initial chain lengths are short, thus requiring fewer cached hashes.

Initialization: The initial value of the shared secret s and the base chain length Gk are selected and exchanged between the server and the client during the initial HTTPS authentication.

The cache is filled by the fillCache function, if required.

```
K:= Gk II k is initially set to Gk*1
R:=1 II r is the current chain number
Call fillCache()
```

Filling Cache:

This fills the cache, if it is being used.

fillCache (No Caching):

```
return    //don't do anything. No caching required.
```

fillCache (Selective Dynamic Caching):

```
miniCacheInterval : = √k   //square root of k
miniCacheK : = k – miniCacheInterval   //highest item
miniCacheIndex : = cache.size - 1    //point to last item
cache[O] = s;
For i : = 1 to miniCachelndex Do
cache[i] : = hashminiCachelnlerval( cache[i-l])
End-For
```

fillCache (Full Caching):

```
cache[O] : = s
//Perform the hash operation k times to obtain Hꞵs)
For i:=1 to k Do
cache[i] : = hash( cache[i-l])
End-For
```

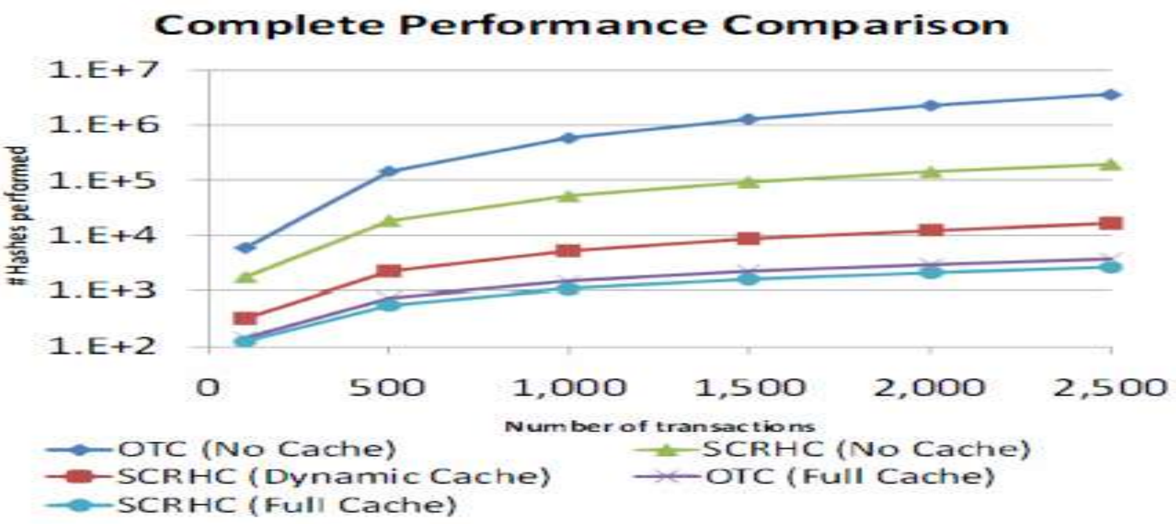

Fig-3: Performance Comparison

Performance comparisons were made between our protocol and OTC. Our tests modeled the limitation encountered at initialization regarding the lack of accurate knowledge of the session length (number of transactions during the lifetime of the session).

VII. CONCLUSION

In this paper, we presented two different issues caused by session hijacking and malicious code that proved to offer better. Our selective dynamic caching technique significantly reduced workloads while requiring trivial amounts of storage. We conclude that our algorithm could provide valuable security to Social Networking Sites in a flexible and adaptable manner.

REFERENCE

[1] **Security against Malicious Code in Web Based Applications,** Third International Conference on Emerging Trends in Engineering and Technology, **Mrs.Yogini A. Kulkarni Mr. Rajendra.G. Kaduskar,** 2010 IEEE,pg no. 286-291

[2] Protocol for Mitigating the Risk of Hijacking Social Networking Sites, Jeffrey Cashion and Mostafa Bassiouni, 2011 ICST,

[3] **Security, Privacy, and Trust in Social Networks, Komathy Karuppanan**

[4] http://en.wikipedia.org/wiki/Social network service

[5] Hack Attacks Revealed. A complete reference for UNIX, Windows & Linux with custom security Toolkit Second Edition Author: John Chririllo.

[6] Code Hacking a Developer Guide to Network Security Charles River Media Author: Richard Conway/Julian Cordingley First Edition.

[7] Hacking a Terror Network Syngress Author: Russ Rogess Mathew G. Devost first Edition

[8] Detecting Malicious JavaScript Code in Mozilla IEEE Oystein Hallaraker and Giovanni Vigna Available: http://www.ieee.org.

[9] A. X. Liu, J. M. Kovacs, C. Huang, and M.G. Gouda."A Secure Cookie Protocol." IEEE, 2005.

[10] K. Fu, E. Sit, K. Smith, and N. Feamster. "Dos and don'ts of client authentication on the web." Proceedings of the 10th USENIX Security Symposium, August 2001.

[11] J. Cashion and M. Bassiouni "Robust and Low-Cost Solution for Preventing Sidejacking Attacks in Wireless Networks using a Rolling Code" to appear in the Proceedings of the 7th ACM International Symposium on QoS and Security for Wireless and Mobile Networks (ACM Q2SWinet), Miami Beach, Florida, October 31-November 4,2011.