A Survey of Security Vulnerabilities in Social Networking Media – The Case of Facebook

Elizabeth Fokes
Department of Information Technology
Southern Polytechnic State University
1100 South Marietta Pkwy
Marietta GA 30060
01-678-915-4292
efokes@spsu.edu

Lei Li
Department of Information Technology
Southern Polytechnic State University
1100 South Marietta Pkwy
Marietta GA 30060
01-678-915-3915
lli3@spsu.edu

ABSTRACT

This paper conducted a survey study on the security vulnerabilities in one of most popular social networking site, Facebook. We divide the vulnerabilities into two main categories: platform-related and user-related. For each vulnerability, we present its origin, description and remedy if there is any. Our work not only increases users' awareness of those vulnerabilities, but also provides a comprehensive view to the researchers who are interested in improving security measures of social media services.

Categories and Subject Descriptors

A.1 [**General Literature**]: Introductory and Survey.

General Terms

Documentation, Security

Keywords

Security; Vulnerability; Facebook; Social Media.

1. INTRODUCTION

The statistics regarding use of social networking media are staggering. According to Statistic Brain in 2012, 58% of the population in US used any social network; 56% used Facebook; 14% used Linked in, 11% used Twitter; 9% used Google+. [18] Social media has also become a platform for a very large number of businesses. In 2011, 77% of Fortune Global 100 Companies use Twitter, and 61% have a Facebook page. [15]

Social media sites, in particular are the focus of vulnerabilities, because of their large user base and the instant attention that is brought about by the attack. Social networking sites have databases of user activities, their email addresses, potentially financial information, ISP addressing information, and some have authorized location tracking. These are all items that can be potentially utilized for great harm against users.

There are many different types of vulnerabilities that users may encounter. Vulnerabilities cause a disruption in the integrity, confidentiality and availability of services. Some are seen directly, and others are not seen at all by the user. Attacks can be launched against users and against the social networking media services individually. There are many motivations behind attacks: financial gain, social hacking, activism, or intentional harm against someone or a company that the attackers do not like. Attackers go after a user's account for financial gain, social hacking, activism, and intentional harm against someone or a company that the attackers do not like to name a few.

The rest of the paper is organized as follows: section two describes the research method used. Section three presents the security vulnerabilities specific to Facebook. Section four concludes the paper.

2. RESEARCH METHOD

We conducted extensive literature search on the vulnerabilities of Facebook. We collected information not only from academic sources but also Internet sources due to the topic our paper. We then created a categorization scheme to sort the sources into different categories and sub categories. For each source, we recorded the description of the vulnerability and potential remedy to it if there is any.

There are two major categories for vulnerabilities: platform-related and user-related. The platform-related vulnerability is divided into four sub categories: SMS verification, social authentication, applications, and general web-based vulnerabilities. For user-related category, there are three sub categories: fake profile, Sybil-type vulnerability, identify theft and access to user account.

The detail of each security vulnerability is described in the next section.

3. FACEBOOK SECURITY VULNERABILITIES

3.1 Technical Background of Facebook

Facebook has an extremely large user base. The company has had to come up with solutions to handle data at a rapid pace and to be able to keep up with the demands. As part of that initiative, in 2012, the company built Presto (a distributed SQL query engine that supports ANSI SQL) that would enables the system to work at a "petabyte scale." This system is implemented in Java, as

RIIT'14, October 15–18, 2014, Atlanta, Georgia, USA.
Copyright © 2014 ACM 978-1-4503-2711-4/14/10…$15.00.
http://dx.doi.org/10.1145/2656434.2656444

Facebook has used Java for the rest of the data infrastructure. Hive/HDFS and Scribe (log server) are on the backend. [19]

3.2 Vulnerabilities Relating Specifically to the Facebook Platform

As with other websites, Facebook has a user interface and the supporting infrastructure which handles requests, databases, and user services. Some weaknesses seen on the Facebook platform include issues with SMS Verification, Social Authentication, vulnerabilities that come through Applications, and Puppetnets. What is special about these types of attacks is that they are not affiliated with what a user does, but rather has to do with the programming that has been done by the service, and manipulates that code which is on the server(s). The issues that come from the vulnerabilities in this section cannot be solved by the user, but rather the company has to solve them. In some cases, companies like Facebook will offer large rewards for finding the coding vulnerabilities.

3.2.1 SMS Verification Weaknesses

SMS stands for "Short Message System." When a user signs onto a user has the option of having additional verification of their identity via the use of SMS verification. SMS verification is a great resource for a user who utilizes multiple computers to access Facebook and doesn't want to accidentally leave his or her account exposed for others to access, or for a user who feels that an additional layer of security is preferable for any other reason. Facebook offers SMS for secondary verification of account ownership, and for sending updates to profiles. This is a second layer to account verification in addition to the password on the account. The goal of this is to be able to make an account more secure because of the second layer of verification that the user has input. The process is simple for a user to follow. When someone chooses to sign on to Facebook and has the SMS option for secondary verification chosen on the account, a code is sent to the user's phone or to an email that the user assigns for verification purposes. When the user receives the code, he or she then inputs the code into a screen that comes up on Facebook that requests the code.

In June of 2013, Jack Whitten, a security researcher won an award from Facebook in the amount of $20,000 for finding a very serious flaw on Facebook. This flaw would allow a user's account to be taken over by a hacker through the exploitation of a weakness in SMS verification of user accounts. Facebook's system for SMS had a flaw in it wherein a hacker could modify the information that is input into the "profile_id" field to the identification of another user, thereby attaching the victim's account to the attacker. By doing this, the attacker could then request a password reset on the victim's account and the verification information would then be sent to the attacker's phone instead of the user's phone. The attacker could then reset the password without the user being aware of it, and the user is locked out of his or her account. "We enter this code into the form, choose a new password, and we're done. The account is ours[2]

The good news with this vulnerability is that this SMS bug was reported to Facebook on May 12, 2013. It was fixed on May 28, 2013. Facebook disabled the profile_id parameter from users in order to protect their users from this vulnerability.

3.2.2 Social Authentication

In response to Mark Zuckerberg's account having been hacked, Facebook added authorization features. Like SMS, social authentication was created as a type of two-factor authentication (where a user will have to provide two pieces of information for authentication). In Facebook, social authentication utilizes the use of a selection of photos belonging to friends in order to prove identity. "Facebook is the largest storage for photos with approximately 1 billion uploaded photos." [3]

The use of social authentication was put in place in Facebook to help an account holder retrieve his or her account when a password has been lost or stolen. Should Facebook determine that there is the possibility that an account has been stolen, the photos of friends will be posted for a user to say whether or not they know the people and who they are. Facebook questions suspicious logins if a user logs in from a different location than normal, or if the user is logging in for the first time to the account from a device that is not recognized as being affiliated with the user's account. The social authentication layer that Facebook put in place utilizes photos that the user must identify in order to access the account. It is important to note that until recently, one could not keep others from accessing the photos on their account. It was easy to copy the photos without a user's permission.

When the social authentication protocol responds, the first item that is presented is a CAPTCHA that has to be solved before the challenges are presented. While this is not something that computer can easily solve, a person can. It does, however, serve the purpose of slowing down a potential attacker. Next come the social authentication challenges. Specifically, there are seven challenges presented, which must be completed within 5 minutes. Each "challenge is comprised of 3 photos of an online friend; the names of 6 people from the user's social circle are listed and he has to select the one depicted. The user is allowed to fail in 2 challenges, or skip them, but must correctly identify the people of at last 5 to pass." [4] The average Facebook user has 190 friends, with the allowable upper limit of 5000.

When Polakis and friends looked at Facebook initially, they thought that the vulnerability would only attack those users who had their photos and friends lists publicly visible. However, they discovered that by 47% of users did not secure their photos and friends lists. When someone "friended" one of the users, there was a 90% success rate in matching friends with photos. [4] This brought the overall percentages of Facebook users who were vulnerable to 84%. It was also found that 71% of Facebook users have at least one photo album accessible publicly.[4] When tested, the research group was able to access a user's account within a minute with the use of a software program designed for facial recognition.

The reason that the testers were able to access user accounts was partially due to the "tagging" option related to photos on Facebook. "Tagging" refers to the option to click on an image and say who the person or people are, individually on a photo. If that person is a Facebook user, it will attach to the user's account. One does have the option to be notified when someone "tags" a user, and there is the ability to refuse the "tag." However, many users do not have the protocol set up to be notified and refuse the tagging. Also, when a user has open "friending" allowed on the account, an attacker can easily access the information that the user cares to share, whether it be photo albums or other pertinent information. Users who do not allow themselves to be tagged, and do not have public profiles or photo albums are the ones who are not as easily accessed, however other users can compromise this should they tag or share information about the user.

In testing the vulnerability, the researchers found that they were able to issue a large number of friend requests to users, and many would passively accept the requests. They collected photos and the associated URLs from the targets via screen scraping methods, and they stored the metadata, comprised of URL of the individual users, UID of the owner, tags, and coordinates. [4] They then scanned the downloaded photos from photo albums and used a face detection classifier. They were able to label the photos with the UIDs of the user associated with them. Next, the researchers assigned the user names with the photos. Through the short range of user names that are presented in the challenge, the attackers were able to lower the scope of matching photo with user name.

There are ways to make sure that an account is secure against such attempts of social authentication attacks. Users can have a message sent to a trusted device to alert them that their account is experiencing an attempt for access. That feature is set up in the security settings on Facebook. The user would simply receive the notification and verify or deny that they are trying to access the account. The message has a security token sent and they can input the information into the account. Of course, if an attacker has access to that device, he or she can get around the extra layer of protection. If the challenge is failed in an attempt, the account moves to a security page that notifies the user (or attacker) that the authentication has failed. If the attacker is successful, there will be no notification sent to the user. Another possible solution is for the account to automatically notify a user whenever the account is accessed from any location. This is typically sent to the user's email account. As long as the attacker does not have access to the email, the user can respond. If the email is accessible, the attacker will still have access until such time as he or she is able to change the password on the account.

Specifically in relation to photos, a user can use one of the following solutions: lock a photo when it is being uploaded to their account, stop the use of print screen and use of snipping tool, remove the right click option on the locked photo, and remove the feature of photo share on a locked photo. [5]

3.2.3 Vulnerabilities from Applications

Facebook offers applications (apps) to enhance the user experience. These applications come from third party vendors who host them on remote sites, which are accessed via the Facebook platform. Applications offer music, games, horoscopes, and puzzles. "Facebook provides developers an API that facilitates app integration into the Facebook user-experience. There are 500K apps available on Facebook, and on average, 20M apps are installed every day. Furthermore, many apps have acquired and maintain a large user-base. For instance, FarmVille and CityVille apps have 26.5M and 42.8M users to date." [6] Many apps ask for far more information from user accounts than they truly need. This information can be leaked to third parties. It only costs $25 to put an app on Facebook. [6] Before an app is added to a user account, he or she needs to authorize it (Facebook uses OAuth 2.0 for authentication and authorization of third-party applications). By default, basic information is provided once the user authorizes an application access to his or her account. [7] Some applications have extended permissions, and will post on user accounts, access posting information, gain the user's birthday information, email address, and access user's messages to other users. They can even access information from the user's friends' accounts.[7] In 2012, the App Center was introduced to Facebook. Applications found here are considered to be of higher quality than others, and only contains a few thousand applications in the list.

Hackers are particularly interested in using apps, because it can be financially rewarding. Through malicious apps, personal information about users can be obtained, apps can suggest use of other apps (reproduction), and can spread to a large number of users via spam generated from friends lists. There is really no way to know if an app is malicious or not from the user end, and therefore, they can easily spread from one user to another. Instead of concentrating on malicious apps, Facebook has concentrated on spam and malicious posts.

3.2.4 Puppetnets

"Puppetnets exploit the design principles of World Wide Web. Web pages can contain links to pages hosted at different domains, other than the one they are hosted at. A malicious user can craft special pages that contain thousands of links pointing at a victim site. When an unsuspecting user visits that page, her browser starts downloading elements from the victim site and thus consuming its bandwidth." [8] Basically, "Puppetnets rely on web sites to coerce web browsers to (unknowingly) participate in malicious activities. Such activities include distributed denial of service, worm propagation, and reconnaissance probing, and can be engineered to be carried out in stealth, without any observable impact on an otherwise innocent-looking website." [9] Users come into contact with this vulnerability, because Facebook hosts applications for games and other "fun" activities for users. These applications are very popular on Facebook, and they insist on inviting friends on playing the games, either by automatically posting on user accounts or causing "better play" through invitation of friends. When a designer of an application wants it to be hosted by Facebook, he or she registers with Facebook and then submits it through Facebook's developer application. The developer application requires information on the Canvas page (the main page of the application) URL and the Canvas callback URL (the address of the web server where it is actually being hosted on). One can write a malicious application and have hidden documents that the victim then hosts and unknowingly shares (for instance, self-executing files). [8]

"An adversary can take full advantage of popular social utilities, to emit a high amount of traffic towards a victim host. However, apart from launching a DDOS attack on third parties, there are other possible misuses in the fashion of Puppetnets" [8] They can be host scanning, where an application can identify open ports through HTTP requests, malware propagation ("Every user that interacts with the application will propagate the attack" [8]) attacking cookie-based mechanisms, embedded self-signed Java applets, personal information leakage, URL scanner cloaking, and collection of sensitive information. A collection of web browsers can be transformed into a distributed system that an attacker can control, hence the term "puppetnet." "Puppetnets expose a deeper problem in the design of the web. The problem is that the security model is focused most exclusively on protecting browsers and their host environment from malicious web servers, as well as servers from malicious browsers. As a result, the model ignores the potential of attacks against third parties." [9] Puppetnets exploit the architecture of a site that allows dynamic content and then amplifies vulnerabilities, which can cause significant damage to a website. It is tough to eliminate because site functionality will be eliminated in doing so.

In order to protect against puppetnets, disabling Javascript would cause a reduction of this threat, but it would not eliminate it. Of course, on Facebook, if one disables Javascript, one cannot access the applications that are the potential source of the issue. In actuality, puppetnets can still work with Javascript disabled, but it the effectiveness of the attacks would be less. On the server side,

"one way for doing this is for servers to use the "Referer" tag of HTTP requests to determine whether a particular request is legitimate or compliant… the server could consult the appropriate access policy and decide whether to honor a request. This approach would protect servers against wasting their egress bandwidth, but does not allow the server to exercise any control over incoming traffic." [9]

Another way to utilize the "Referer" tag is to shut down the controlling website by tracing the source of the attack. This process is time consuming, as it is done by people and not machines. Once the controlling website no longer has access, it can still take up to an hour for all of the puppet browsers to become pointed elsewhere. [9]

3.3 Vulnerabilities Relating Directly to the Facebook User

These vulnerabilities are affiliated with the users, because they do not attack the servers, and they do not involve third-party applications. Instead, these vulnerabilities involve the actions of other users, which can be thwarted by the users themselves. Users, for the most part are aware when these types of attacks happen, unlike those affiliated with the service and with applications. On an emotional and psychological level, the user can be gravely impacted, because of the social aspect of social media. Like chat rooms of the past, where users would become emotionally attached, social networking sites have the same emotional draw. Part of this is due to the fact that real life friends and family interact directly with the user through this platform. In this section, fake profiles, identity cloning, cyberbullying, and injection attacks will be considered.

3.3.1 Fake Profiles

Fake profiles have been used by sexual offenders, people meaning to defame or harm other users, and others who wish to launch attacks. [10] "The personal risk associated with these types of attacks includes kidnappings, child molestation, sexual abuse, defamation and other forms of harassment and indecency." [10] Because of these types of activities, fake profiles are a risk that can be very serious. As mentioned previously, users can put whatever information they want when an account is set up. There is no authentication other than checking to see if the email address affiliated with the account is real. The other aspects of the user profile are not checked.

Users are asked if they would like to add a friend to their friends list. While many users do check to see if they truly know a person who would like to connect, others do not properly scrutinize the invitation and will accept anyone who wants to be a friend. This opens the user up to the possibility that the friend requesting acceptance could be malicious. This is a trust relationship between the user and the person asking to be added as a friend. "Trust can be defined as the willingness of an individual to be vulnerable to the actions of another individual, based on the expectation that the other will perform a particular action. This acceptance of vulnerability and risk is irrespective of the ability to monitor or control the behavior exhibited by the other party involved. Another view defines trust as a mental phenomenon that occurs within social contexts and applies to both online and offline environments." [10] Based on the profile presented to the user, trust will determined by what information is presented regarding the person wishing to be added to the friends list.

Users feel a confidence in the system, because of the following elements: Users also trust that their posted information is honest. Users have a strong perception that their information is safe and that they will encounter honesty from other users. They not only trust other users, but they trust the system[10] There are controls available to user regarding what is seen by other users, and this can limit who sees what is posted by the user. This adds to user confidence. When a person is trusted, less controls put into place by the user. There are also controls available if another user needs to be removed from a friends list, or if a user is upset by something that another user posts.

When an attacker decides to harm another user with a fake profile, there are many things that they can do. Harm can come to a user via psychological harm. There have been reports of users who have been bullied by other users who intentionally joined the service under a fake profile for this sole purpose. Due to the psychological harm of bullying, there can be physical and emotional implications. One can identify a problem user to the service for investigation. While the offending user profile can be blocked by the service, there is nothing in place to keep the malicious user from creating a new profile and resume the malicious activities against the victim. This makes the blocking of users ineffective.

3.3.2 Sybil

This type of attack is most often found in peer-to-peer networks, which Facebook, as a social networking platform, can be described as a model of. In this type of attack, the reputation system is forced to make decisions that benefit an adversary by being provided false or biased information under a number of identities. [11] In a Sybil attack, "a single faulty entity can present multiple identities, it can control a substantial fraction of the system, thereby undermining this redundancy. [12] "Our third example comes from a Facebook voting application. If an adversary maliciously creates many identities, she can easily change the overall popularity of an option by providing plenty of false praise, or bad-mouthing of the option through Sybil ids." [11] Sybil attacks can look a great deal like identity cloning. However, in a Sybil attack, the attacker is not stealing the identity of another user; he or she is making multiple profiles instead. Each identity that a Sybil attack creates has a direct node attached to it. By having the multiple profiles, one can "influence the choices made by victims' friends using the trust built in friendships. [13] An attacker can use the identities to launch malicious messages and spam other users.

One way to mitigate the issue of a Sybil attack on distributed hash tables is use a product called X-Vine, which is "resilient to denial of service attacks, and in fact the first Sybil defense that requires only a logarithmic amount of state per node, making it suitable for large-scale and dynamic settings. X-vine also helps protect the privacy of users social network contacts and keeps their IP addresses hidden from those outside of their social circle, providing a basis for pseudonymous communication." [14]

3.3.3 Identity Theft

As mentioned when discussing fake profiles, users merely need a working email address to create an account on Facebook. Web-based email accounts are accepted as proper identification of a user on the Facebook platform. Some users allow their web-based email accounts to expire, due to inactivity, while they are user of Facebook. "some users may decide to delete their own email accounts, without realizing the security threats that this action entails. Such threats arise from the fact that the same web-based email services allow any other willing user to *reactivate* and use the *same* email address which had previously expired, when they sign up." [16]

In relation to identity theft, it has been found that one can take a photo that is unlocked of a user, and the user name to create a new account (with an email address) and post it into the details of the account. By doing so, the account will show the name of a user and the associated photo. An attacker can then accumulate friends associated with the user name, because they are none the wiser. This effectively is another route for identity theft. Users need to be careful when adding someone by verifying with their friend before adding an additional account with their name and photo attached to it.

The researchers found that they were able to access a user's account if they were aware that the user was using a web-based email that had expired. This was relatively simple. "Once we have acquired control of a previously expired email address, which had once been used to open up a Facebook account, we can visit Facebook on the web and claim to the user in question and have forgotten our password. Facebook then promptly sends an email to our reactivated Hotmail email address, which contains a code that allows us to reset the password for the Facebook account in question." [16] Once the researchers have access to the account, they can look at the friends list and see the email addresses affiliated with those who share personal information. They can then test the email addresses to determine which ones have expired web-based email and take over the newly exposed accounts. In testing, the researchers found that out of 760 friends, 4 were susceptible to the exploit. From there, they were able to get to 15 accounts, and subsequently they attempted their attack on 2000 friends to find that 23 were vulnerable. [16]The fault of this issue is not Facebook's alone, but also Hotmail's fault, because Hotmail did not delete inactive accounts. While the researchers chose Hotmail as the attack vector, there are other web-based email services that may have the same level of laxity on account deletion. Researchers noted that, "techniques such as IP spoofing, using a proxy server, or using a public workstation would significantly reduce the risk of tracing the attack back to its origin."[16]

There are limitations to this kind of attack. First of all, the attackers were unable to target a specific user. An attack has to be initiated from the friends list, which the user has imported from his or her Hotmail address. "Hotmail and Windows Live user are currently susceptible to this kind of attack." [16]

Recommended ways for users to protect themselves from the vulnerability of identity theft are simple. Users could use an email address that is not from a web-based service, and can make sure that the email is active. The user can also add SMS authentication to their account as an additional layer of security. While SMS is not a solution on its own, it can be used in addition to other security choices.

3.3.4 Accessing User Accounts Even When Blocked

In 2008, penetration tester Byron Ng discovered a way to clone an account on Facebook the use of a user's ID number. The vulnerability works even on accounts that have the attacker blocked, as long as there has been some level of correspondence between the victim and the attacker. Every user account on Facebook has a number and it can be found just after the "profile.php?id=" part of the URL for a user's account. In testing, he got the number and then clicked a link that would send a message to the victim. Obtaining the number is pretty simple. One merely needs to find a tagged photo. The identification number in the tag is the album's owner's number. Upon obtaining the ID number, it is easy to access the user's account. An attacker would then do a search on the ID number. Even if an error message pops up, it will autocorrect to the pid number that the user was most recently tagged in. "From there, you'd take the given URL and delete the entire &id portion, leaving just &subj=####### as the end of the URL. Hit enter, and *voila*! Instant access to the last photograph the target was tagged in, and access to the entire album of pictures from which that one image resides, whether you're the friend of the individual who created it or not." [17]

With a little Firefox extension called Firebug, a user can open up web pages to "tweak" them. For instance, some applications have the option to "send gifts" to other users. The attacker simply needs to go to the gift sending page and enter the name of your friend in the **to:** field. "Right click on the Send Gift button and click Inspect Element. Then click on the Dom tab at the top of Firebug's little window. Scroll down – you're looking for the To field. When you find it, you'll see a number. Guess what? It is the Facebook ID number of the person you entered in the To: field! Click on the number and Firebug will open up a large list of other options. Scroll down until you've found the "Value" field – it should be right below the "Type: Hidden" option. Double click on the ID number and enter the target's Facebook ID in quotes. Hit enter, then turn your attention to the Free Gifts sending page and hit Send. Blam. One anonymous gift to someone who isn't your friend/has blocked you/ whatever." [17]

4. DISCUSSIONS

This paper surveyed security vulnerabilities in one of most popular social networking sites, Facebook. We hope our research not only increase users' awareness of those vulnerabilities, but also provide a comprehensive view to the researchers who are interested in improving security measures of social media services.

5. REFERENCES

[1] Bowman, M., Debray, S. K., and Peterson, L. L. 1993. Reasoning about naming systems. *ACM Trans. Program. Lang. Syst.* 15, 5 (Nov. 1993), 795-825. DOI= http://doi.acm.org/10.1145/161468.16147.

[2] Kirk, J. (2013, July 27). *Researcher nets @20K for finding serious Facebook flaw*. Retrieved September 25, 2013, from CSO: http://www.cso.com.au/article/466029/researcher_nets_20k_finding_serious_facebook_flaw/

[3] Albesher, A., & Alhussain, T. (2013). Privacy and Security Issues in Social Networks: An Evaluation of Facebook. *ISDOC '13 Proceedings of the 2013 International*

Conference on Information Systems and Design of Communication (pp. 7-10). New York: ACM.

[4] Polakis, I., Lancini, M., Kontaxis, G., Maggi, F., Ioannidis, S., Keromytis, A. D., et al. (2012). All Your Face Are Belong to Us: Breaking Faceook's Social Authentication. *Annual Computer Security Applications Conference* (p. 399). Orlando: ACSAC.

[5] Sharma, R., Jain, A., & Rastogi, R. (2013). A new face to photo security of Facebook. *2013 Sixth International Conference on Contemporary Computing (IC3)* (pp. 415-420). Noida: IEEE.

[6] Rahman, M. S., Huang, T.-K., Madhyastha, H. V., & Faloutsos, M. (2012). FRAppE: Detecting Malicious Facebook Applications. *CoNEXT 2012 Proceedings of the 8th International Conference on Emerging Network Experiments and Technologies* (pp. 313-324). New York: ACM.

[7] Huber, M., Mulazzani, M., Schrittwieser, S., & Weippi, E. (2013). AppInspect: Large-scale Evaluation of Social Networking Apps. *ACM COSN Proceedings of the First ACM Conference on Online Social Networks* (pp. 143-154). Boston: ACM.

[8] Jagnere, P. (2012). Vulnerabilities in Social Networking Sites. *2nd IEEE International Conference on Parallel Distributed and Grid Computing (PDGC)* (pp. 463-468). Solan: IEEE.

[9] Lam, V. T., Antonatos, S., Akritidis, P., & Anagnostakis, K. G. (2006). Puppetnets: Misusing Web Browsers as a Distributed Attack Infrastructure. *ACM Conference on Computer and Communications Security. 12.* New York: ACM.

[10] Galpin, R., & Flowerday, S. V. (2011). Online Social Networks: Enhancing User Trust Through Effective Controls and Identity Management. *Information Security South Africa (ISSA)*, 1-8.

[11] Chang, W., & Wu, J. (n.d). *A Survey of Sybil Attacks in Networks.* Temple University, Computer and Information Sciences. Philadelphia: Temple University.

[12] Douceur, J. R., & Donath, J. S. (2002). The Sybil Attack. *International Workshop on Peer-to-Peer Systems* (pp. 251-260). Cambridge, MA: IPTPS.

[13] Jin, L., Long, X., Takabi, H., & Joshi, J. B. (n.d). *Sybil Attacks VS Identity Clone Attacks in Online Social Networks.* Pittsburgh: University of Pittsburgh.

[14] Mittal, P., Caesar, M., & Borisov, N. (2010). Facebook under attack on all fronts. *Network Security, 5*, 1-2.

[15] McNaughton, M. (2012). *77% of Fortune Global 100 Companies Use Twitter.* Retrieved November 20, 2013, from The Realtime Report: http://therealtimereport.com/2011/03/18/77-of-fortune-global-100-companies-use-twitter/

[16] Parwani, T., Kholoussi, R., & Karras, P. (2013). How To Hack Into Facebook Without Being A Hacker. *WWW '13 Proceedings of the 22nd International Conference on World Wide Web Companion*, 751-754.

[17] Murphy, D. (2008, March 27). The Tip of the Facebook Exploit Iceberg. *MaximumPC.*

[18] *Social Networking Statistics*. (2013, August 12). Retrieved November 5, 2013, from Statistic Brain: http://www.statisticbrain.com/social-networking-statistics/

[19] Traverso, M. (2013, November 6). *Presto: Interating with petabytes of data at Facebook.* Retrieved February 4, 2014, from Facebook Engineering: https://www.facebook.com/notes/facebook-engineering/presto-interacting-with-petabytes-of-data-at-facebook/10151786197628920