AN EXAMINATION OF SOFTWARE ENGINEERING TECHNIQUES THROUGH A CASE STUDY OF FACEBOOK

by

Michael P. Dippery

A Proposal Submitted to the Honors Council

For Honors in Computer Science

October 12, 2007

Approved by:

____

Adviser: Xiannong Meng

____

Co-Adviser: Luiz Felipe Perrone

____

Department Chairperson: Xiannong Meng

1 Thesis Statement

Social networking websites, which allow individuals to build and strengthen social bonds between one another, are becoming increasingly common on the World Wide Web. One such social networking site is Facebook, a site that is particularly popular amongst college students across the United States [11]. Facebook recently opened up an application programming interface (API) that allows third-party developers to write web applications that run in conjunction with Facebook, as though they had been created by Facebook's own developers [5]. In doing so, they exposed some of the "innards" of Facebook to the rest of the world, which gives an excellent opportunity for others to study issues concerning the security and privacy of Facebook.

I plan to assess the security of Facebook's new API, primarily from the standpoint of users' privacy, and examine what (a) Facebook did *right* in its implementation that can be adapted by other developers, and (b) what Facebook did *wrong* that should be fixed by others who seek to develop a similar system.

2 Project Description

The project will consist of a security audit of Facebook's API from the perspective of a malicious user (viz., me). I plan to first create a developer's account with Facebook (such accounts are free to all users) and study the design of its API.

Essentially, Facebook's public API allows third-party developers to write applications that tie into Facebook's own software platform and reservoirs of data. These third-party applications are treated as "first-class citizens", meaning that, for the most part, they look and behave exactly the same as Facebook's own in-house applications. The only difference is that they are subject to stricter security controls than Facebook's own applications; this additional layer of security establishes the crux of my research proposal.

With my developer's account, I plan to write software to interact with Facebook's data—again, from the point of view of a malicious developer. Of course, I do not plan on doing anything *illegal*; by *malicious*, I mean a developer that is attempting to circumvent Facebook's security to steal or alter private data. All attempts will be made on "dummy" accounts set up by myself or my peers (since any person with a valid email address can set up a Facebook account).

There is no denying the fact that a great amount of personal data is stored—sometimes in a very open way—on social networking sites such as Facebook. In fact, recent research has sought to study this problem qualitatively and quantitatively [1, 11]. One study at the Massachusetts Institute of Technology (MIT) found that a staggering amount of personal information was stored on Facebook, even though most users were unfamiliar with the privacy settings available to them [1, 11]. On numerous occasions, this information has fallen into the hands of people who were not supposed to see the information [1]. Sometimes these incidents are merely embarassing, but sometimes they get the victim into trouble with his school or local legal authorities [1].

My primary interest is an assessment of how well Facebook protects private information. I plan to either (a) demonstrate flaws in the way Facebook protects users' data by gaining unauthorized access to personal information (in the dummy accounts I had previously set up); or (b) assess the proactive steps Facebook take in protecting private information. I will accomplish these goals by using Facebook's API. First, I will write an application that ties into Facebook's platform (using the API provided to third-party developers). This application will function just like other third-party Facebook applications, except that it will attempt to gain unauthorized access to a user's personal data. Initially, I plan on writing an application that acts like a *Trojan horse*, a piece of software that appears benign but really has a malicious intent. I am taking this approach because users must *add* applications to their Facebook user profiles/accounts before they can make use of them on Facebook; in other words, an application that acts like a computer virus or worm—i.e., one that spreads without user intervention—is likely not possible, but I could look into that possibility as well.

I hypothesize that Facebook is just a standard web application, and as such, suffers from the usual flaws that plague many web applications, e.g., buffer overflow errors and Structured Query Language (SQL) injection flaws. Both of these are “classic” problems in software. A buffer overflow occurs when a piece of software is allowed to access and write to an unauthorized chunk of memory; buffer overflows represent many of the vulnerabilities in software [3]. A SQL injection occurs when a malicious author runs specially-crafted commands wrapped in a SQL statement (SQL is a language used to communicate with many database software); these attacks can be quite damaging to web applications, too, and are relatively common and easy to exploit [2]. Much of the thrust of my research will be in these areas.

I also plan to use the third-party API access to ascertain how data transmissions occur between users and Facebook, i.e., is data secured, and are passwords and other sensitive data encrypted or otherwise obscured in any way? Secure data transmission between a client (the Facebook user) and the server (Facebook) is important because of the relatively insecure nature of computer networks. For example, on a typical college campus, a fair amount of data transfer takes place over a wireless connection [8]. Unfortunately, a wireless network is relatively insecure and prone to “sniffing” of data by malicious network users, which can expose passwords and other personal information [13]. Such exposure can be prevented in numerous ways (e.g., encrypting the transmission between Facebook and the user), *but only if such a mechanism is implemented by Facebook.* Thus, an audit of the security of Facebook’s web application framework would be incomplete without an assessment of the mechanisms by which Facebook protects the data stream between a client and the server.

Finally, Facebook’s framework is built on many freely available, open-source technologies.

Based on the information on its employment page, Facebook uses technologies like PHP, MySQL, and Python [6]. A thorough security examination cannot stop with Facebook's software alone. I plan to assess the very tools and software that are used to build the foundation of Facebook's software. Many of Facebook's underlying technologies are open-source, which means the source code—the computer code used to build the software—is freely available and can be downloaded by anyone with an Internet connection. Thus I will be able to examine the very software that makes up Facebook's foundation. Flaws found in such tools may easily have propagated to Facebook's framework.

3 Significance and Contribution

An assessment of best-practice techniques for building web applications would be my direct contribution in the field of web applications. Such a contribution would greatly aid in the development of better software engineering techniques for web applications.

Tangentially, a number of other contributions will possibly come to fruition. For example, part of my research will likely involve the creation of security-related tools, much like Jones and Soltren [11]. These, too, will aid in the furthering of techniques for the engineering of web applications software.

Research in this field is important because the use of web applications, both in general and specifically in social networking sites, is increasing steadily. Unfortunately, the growth of web applications in a social networking context is not necessarily parallel with an increase in techniques for engineering and securing such applications [9, 10, 12, 15]. The growth of the popularity of the World Wide Web was not entirely expected, and its commercial and personal uses quickly expanded beyond the original intent of the Internet, which brought new security problems to light [7]. This is a major problem for social networking sites like Facebook because such sites often deal with a fair amount of personal data [12].

It is important for software engineers to develop techniques to adequately ensure the security of personal data on such websites. The primary reason for my research is to assess the security of one such web applications model, and use the knowledge gained from this audit to advise other developers in positive techniques for enhancing the security of similar frameworks they may develop. Along the way, I will undoubtedly develop tools for assessing security that may be useful to other developers as well.

Furthermore, Facebook is immensely popular with college students, so popular that one study found that over 80% of matriculating freshmen joined Facebook *before* they entered college [11]. Therefore, this study is of great importance to most college students, who may be familiar with Facebook but are neither intimately familiar with Facebook's security options nor privacy issues [1, 11]. Likewise, many college faculty and staff members are equally uninformed about security and privacy issues related to social networking sites like Facebook. Therefore, there is an immediate connect between my study and the Bucknell population.

However, only a few studies specific to Facebook have been done, and none of them have involved an in-depth security audit of the platform. Jones and Solren [11], and Acquisti and Gross [1] have all done studies of Facebook privacy, but they have mainly been from the standpoint of user behavior, and both were based mostly on surveys. Lee [14] took a more technological approach to Facebook security by analyzing the security options available in Facebook's software, but his study also focused on user behavior in regards to these technological mechanisms; his study was also released too early to test the security of Facebook's new API. Strater and Richter also performed a study of Facebook privacy issues [16], but this, too, was more from a behavioral perspective than a technological perspective. Dwyer and Hiltz also performed a study on Facebook that, once again, focused on behavior and not technology [4]. While these studies are important, without a close look at Facebook's security architecture and its software components, they are incomplete. Thus my research will involve a look at social networking websites primarily from a technical, rather than psychological, perspective in order to gain knowledge on the best ways to implement social networking APIs.

References

[1] ACQUISTI, A., AND GROSS, R. Imagined Communities: Awareness, Information Sharing, and Privacy on the Facebook. In *6th Workshop on Privacy Enhancing Technologies* (2006), Privacy Enhancing Technologies.

[2] BUEHRER, G. T., WEIDE, B. W., AND SIVILOTTI, P. A. G. Using Parse Tree Validation to Prevent SQL Injection Attacks. In *Proceedings of the 5th International Workshop on Software Engineering and Middleware* (2005), SIGSOFT: ACM Special Interest Group on Software Engineering, ACM Press, pp. 106–113.

[3] COWAN, C., BEATTIE, S., JOHANSEN, J., AND WAGLE, P. PointGuard$^{\text{TM}}$: Protecting Pointers From Buffer Overflow Vulnerabilities. In *Proceedings of the 12th USENIX Security Symposium* (August 2003), USENIX, pp. 91–104.

[4] DWYER, C., AND HILTZ, S. R. Trust and privacy concern within social networking sites: A comparison of Facebook and MySpace. In *Proceedings of the 13th Americas Conference on Information Systems* (August 2007).

[5] <http://blog.facebook.com/blog.php?post=2437282130>. Accessed October 9, 2007.

[6] <http://www.facebook.com/jobs/>. Accessed October 9, 2007.

[7] GINIGE, A., AND MURUGESAN, S. Web Engineering: An Introduction. *IEEE MultiMedia* (January–March 2001), pp. 14–18.

[8] HENDERSON, T., KOTZ, D., AND ABYZOV, I. The Changing Usage of a Mature Campus-wide Wireless Network. In *Proceedings on the 10th Annual International Conference on Mobile Computing and Networking* (September–October 2004), SIGMOBILE: ACM Special Interest Group on Mobility of Systems, Users, Data, and Computing, ACM Press, pp. 187–201.

[9] HUANG, Y.-W., HUAN, S.-K., LIN, T.-P., AND TSAI, C.-H. Web Applications Security Assessment by Fault Injection and Behavior Monitoring. In *Proceedings of the 12th International Conference on World Wide Web* (May 2003), ACM, ACM Press, pp. 148–159.

[10] HUANG, Y.-W., YU, F., HANG, C., TSAI, C.-H., LEE, D. T., AND KUO, S.-Y. Securing Web Application Code by Static Analysis and Runtime Protection. In *Proceedings of the 13th International Conference on World Wide Web* (May 2004), ACM, ACM Press, pp. 40–52.

[11] JONES, H., AND SOLTREN, J. H. *Facebook: Threats to Privacy.* Thesis, December 2005.

[12] Joshi, J. B., Aref, W. G., Ghafoor, A., and Spafford, E. H. Security Models for Web-Based Applications. *Communications of the ACM 44*, 2 (February 2001), pp. 38–44.

[13] Karygiannis, T., and Owens, L. *Wireless Networking Security: 802.11, Bluetooth, and Handheld Devices: Recommendations of the National Institute of Standards and Technology.* NIST Special Publication 800-48, National Institute of Standards and Technology, November 2002.

[14] Lee, B. *Privacy and Awareness on Facebook.com.* Thesis, May 2007.

[15] Scott, D., and Sharp, R. Developing Secure Web Aapplications. *IEEE Internet Computing* (November–December 2002), pp. 38–45.

[16] Strater, K., and Richter, H. Examing Privacy and Disclosure in a Social Networking Community. In *Proceedings of the 3rd Symposium on Usable Privacy and Security* (2007), ACM, ACM Press, pp. 157–158.