Chapter 14
Experiences with Threat Modeling on a Prototype Social Network

Anne V. D. M. Kayem
University of Cape Town, South Africa

Rotondwa Ratshidaho
University of Cape Town, South Africa

Molulaqhooa L. Maoyi
University of Cape Town, South Africa

Sanele Macanda
University of Cape Town, South Africa

ABSTRACT

Supported by the Web 3.0 platform that enables dynamic content sharing, social networking applications are a ubiquitous information exchange platform. Content sharing raises the question of privacy with concerns typically centered on vulnerabilities resulting in identity theft. Identifying privacy vulnerabilities is a challenging problem because mitigations are implemented at the end of the software development life cycle, sometimes resulting in severe vulnerabilities. The authors present a prototype experimental social networking platform (HACKMI2) as a case study for a comparative analysis of three popular industry threat-modeling approaches. They focus on identified vulnerabilities, risk impact, and mitigation strategies. The results indicate that software and/or asset-centric approaches provide only a high-level analysis of a system’s architecture and are not as effective as attacker-centric models in identifying high-risk security vulnerabilities in a system. Furthermore, attacker-centric models are effective in providing security administrators useful suggestions for addressing security vulnerabilities.

INTRODUCTION

Social networking applications have become a ubiquitous and popular Internet service enabling millions of users to share content dynamically (Westland, 2012), (Chen, 2013), (Story et al., 2012). This has been enabled in large part by the Web 3.0 technology that extends the Web 2.0 technology by enabling dynamic content distribution and/or sharing (Rowell, 2008), (Walters, 2009). As is the case with a lot of popular web applications storing or manipulating sensitive information, these applications are inherently vulnerable to privacy violations that can be exploited for identity theft and in certain cases compromises to professional integrity. Furthermore, scandals...
such as the one that occurred in July of 2011, where security consultant (Ron Bowles) collected personal data off FaceBook and published it on the public site, Pirate Bay (Pirate Bay, 2010), have further compounded this problem (Boshmaf et al., 2013). Corporate organizations have responded to the potential threats that social networks pose by limiting and sometimes even completely prohibiting access to social networks using corporate infrastructure (Rooksby & Sommerville, 2012), (Wang et al., 2011).

The problem of security and privacy violations in web applications hinges mainly on the fact that security and/or mitigations are typically implemented at the end of the software development cycle (Hackmi2, 2012). As a consequence, there is often no clear strategy or plan to prevent security violations during the software development cycle. In general, implemented mitigations are rushed jobs that can often result in more serious vulnerabilities. An added dimension to consider is that fixing vulnerabilities after an application has been deployed is more costly than if this were done during the design phase. The National Institute of Standards and Technology (NIST) in the United States of America estimates that code fixes performed after the release of a software product can result in 30 times the cost of performing these fixes during the design phase.

Threat modeling has received some attention with respect to integrating security and privacy modeling into the software design process. In general, threat modeling can broadly be described as an approach to security modeling whereby potential vulnerabilities or coding practices that are liable to result in security vulnerabilities are identified and categorized using security rating metrics. Threat modeling tools are typically packaged to additionally provide suggestions of implementable countermeasures. Approaches to threat modeling are categorized under two main themes namely, attack centric models and software/asset centric models. In attack centric models, as the name suggests, the focus is on the attacker’s goals and motivations for hacking into a system. The main motivation behind this model is that in order to design effective countermeasures for the identified security vulnerabilities, a security administrator needs to have a realistic perception of the potential weak points in the system that the attacker is most likely to exploit and why. The software/asset centric model by contrast aims at detecting the vulnerabilities that emerge in the design of the system. In this case, the idea is to have a sort of high-level parsing of the system’s components with the goal being to identify potentially exploitable vulnerabilities. This is a proactive approach to discovering the vulnerabilities on a system from the system designer’s perspective. The advantage is that vulnerabilities can then be addressed with suitable countermeasures to protect against potential attacks and the other associated losses (cost-wise).

Having a blueprint of potential threats is useful from the system designer’s viewpoint because potential attack scenarios as well as countermeasures can be simulated before the application is deployed. However, until recently, not much consideration has been given to comparing existing threat modeling approaches and that software tools that hinge on these models. Yet each one of the existing approaches has its pros and cons in the identification of security vulnerabilities in a system. In applying each one, it is important from an industry and academic perspective to be cognizant of the limitations of each with respect to the overall security modeling goal for the application and/or system.

In this chapter, we present a prototype experimental social networking platform (HACKMI2) as a case study basis for a comparative analysis of three popular industry threat modeling tools namely; the Microsoft Security Development Lifecycle (SDL), the SensePost Corporate Threat Modeling Tool (CTM), and the Microsoft Threat Analysis and Modeling (TAM) tool. The main reasons for selecting these tools are popularity of use in the industry and the fact that these tools