Chapter 14

Case Study: Secure Web Development

Daljit Kaur
Lyallpur Khalsa College, India

Parminder Kaur
Guru Nanak Dev University, India

ABSTRACT

This chapter is an effort to develop secure web applications based on known vulnerabilities. It has been seen that in the rapid race of developing web applications in minimum time and budget, security is given the least importance, and a consequence of which is that web applications are developed and hosted with a number of vulnerabilities in them. In this race, one thing is constant that attackers take advantage of weaknesses existing in technology for financial gain and theft of intellectual property. In this proposed method of secure web development, most common vulnerabilities and their occurrence in development process is discussed. Mapping vulnerabilities to the actions needed to take during development process may help developers to understand vulnerability and avoid vulnerabilities in application.

INTRODUCTION

Since the development of internet, web applications have become very popular, and, nowadays, they are used in every environment, such as medical, financial, military systems. But in the race to develop these online services, web applications have been developed and deployed with minimal attention given to security risks, resulting in surprising number of corporate sites that are vulnerable to hackers (Halkidis, 2008). With the rise in web applications for security critical environment, number of attacks against these applications has grown as well. Organizations face more security risks imposed upon them by hackers on achieving fame, glory, profit or information. Malicious
hackers are finding out constructively new ways to exploit web applications. According to study conducted by Imperva’s Application Defense Center (ADC), web sites experience an average of 27 attacks per hour or about every two minutes (Be’ery et al, 2011). When sites come under automated attack, the target can experience up to 25,000 attacks per hour or 7 per second (Be’ery et al, 2011). The number of daily web based attacks observed was 93% higher in 2010 as compare to 2009, according to report by Symantec on attack Kits and malicious websites (SYM).

Web based attacks are rising consistently and there are number of reasons that make applications so vulnerable. The rise in attack volume in web applications is due to ready availability of attack toolkits, many of which exploit known vulnerabilities (Malek, 2004). Hacking tools and guides are available on-line, almost anyone can launch attack using these resources. Web applications have become interesting target for many for achieving fame, profit or information. Hackers continue to focus on web applications because these are easy points of entry and valuable data is exchanged in business processes run by these web applications. Developers are mandated to deliver functionality on time and on budget but not to develop secure applications. Developers are not generally educated in secure code practices. According to CERT/CC, more than 90% of vulnerabilities leak out during development. They are result of ignoring known vulnerabilities found in other systems (Nany and Gary, 2005). Keeping this fact in mind, this chapter is an attempt that will help to develop partially secure web applications avoiding vulnerabilities.

The chapter describes related work, the development life cycle, most common vulnerabilities in web applications and then explains the actions proposed to avoid vulnerabilities. In the end, mapping of vulnerability to action is done to understand and avoid the vulnerabilities and future scope is discussed.

**RELATED WORK**

There are existing processes, standards, life cycle models, frameworks, and methodologies that support or could support secure software development. Security Development Lifecycle (SDL), which focuses on producing secure software, is a software assurance methodology and tool that aims at assisting software developers, designers, builders, and educators in improving the security of software production. SDL prescribes activities to embed security into applications and supplies the foundation for a broad software security assurance that extends across an IT enterprise (FOR), (Hajar and Salman, 2011). The SDL introduces security and privacy throughout all phases of the development process. SDL includes five phases: Training, policy, and organizational capabilities, Requirements and design, Implementation, Verification, and Release and response. It includes mandatory security activities executed as part of a software development process.

The Requirements phase of the SDL includes consideration of security and privacy at a foundational level. The best opportunity to build trusted software is during the initial planning stages, when development teams can identify key objects and integrate security and privacy (MSD). The Design phase includes building the plan for implementation, release, and functional and design specifications, and performing risk analysis to identify threats and vulnerabilities. Functional specifications may describe security and privacy features directly exposed to users, such as requiring user authentication to access specific data or user consent before use of a high-risk privacy feature. Design specifications describe how to implement these features and how to implement all functionality as secure features (MSD). An extra security activity includes a final code review of new as well as legacy code during the Verification phase. Finally, during the release phase, a Final Security Review is conducted by the Central Microsoft Security team, a team of security experts who are also available to the product development team.