This article outlines a four-point strategy for the development of secure Web-based applications within an agile development framework and introduces strategies to mitigate security risks that are commonly present in Web-based applications. The proposed strategy includes the representation of security requirements as test cases supported by the open source tool FIT, the deployment of a highly testable architecture allowing for security testing of the application at all levels, the outlining of an extensive security testing strategy supported by the open source unit-testing framework HTTPUnit, and the introduction of the novel technique of security refactoring that transforms insecure working code into a functionally-equivalent secure code. Today, many Web-based applications are not secure, and limited literature exists concerning the use of agile methods within this domain. It is the intention of this article to further discussions and research regarding the use of an agile methodology for the development of secure Web-based applications.
INTRODUCTION

E-commerce and Web-based applications have quickly become a staple for many businesses, and in some cases represents an entire business itself. Success or failure in the online marketplace can, in fact, determine a company’s fate. For example, in 2003, Dell Incorporated’s U.S. home and home office divisions generated $2.8 billion or nearly 50% of its revenues through its online storefront (Dell Inc., 2003). Web-based applications are typically “always on,” and although this allows customers to access products and services at all times, it also leaves the applications open to continuous access from malicious attackers. Security is therefore a major concern for Web-based applications, and a breach in security can lead to a significant loss in profit or the exposure of valuable information, such as trade secrets or confidential client information. Furthermore, security and privacy are listed as major concerns for customers utilizing e-commerce systems (Udo, 2001), and security is listed as one of the three most important quality criterion for Web-based application success (Offutt, 2002). According to a recent study, 75% of online security breaches occur at the application layer, not the transportation layer (Grossman, 2004a). Common exploits in modern Web-based applications are occurring due to security flaws within the Web-based application itself, regardless of protection from firewalls and Secure Socket Layer (SSL) communication channels. Malicious attackers appear to be concentrating on what they believe to be the weakest link—the application itself.

SQL injection, buffer overflows, cross-site scripting, file inclusion, URL injection, and remote code injection vulnerabilities have historically plagued Web-based applications developed by both the open-source and commercial communities. These vulnerabilities have been found in Web-based applications employed by organizations such as the FBI, CNN, Time Magazine, Ebay, Yahoo, Apple Computer, and Microsoft (Cgisecurity.com, 2002). These vulnerabilities are not only extremely common in many Web-based applications, but can also be extremely costly. For example, an SQL injection vulnerability in the PetCo.com Website resulted in 500,000 customers’ credit-card numbers and information being made vulnerable to anyone who could carefully construct a SQL query (Grossman, 2004b). Every month, 10 to 25 cross-site scripting security flaws are found within commercial Web-based applications (Cgisecurity.com, 2002). It is clear that the methodologies currently employed for the development of Web-based applications are not adequately meeting security needs.

The development of Web-based systems is often distinctly different from the development of traditional Information Technology (IT) systems as Web-based applications are typically much smaller than traditional IT applications, and consequently their production period is much shorter (Cusumano & Yoffie, 1999; Iansiti & MacCormack, 1997; Kirda, Jazayeri, Kerer, & Schranz, 2001; Ramesh, Pries-Heje, & Baskerville, 2002). Furthermore, these applications evolve at a much faster pace, and the development cycle is highly iterative in nature. Due to these factors, agile development methodologies are increasingly being used for the development of Web-based applications (Barnett & Narsu, 2005; Ramesh, Pries-Heje, & Baskerville, 2002). Until recently, agile development methodologies and traditional security engineering have been seen as orthogonal (Beznosov & Kruchten, 2004); however, recent discussions are beginning to harmonize the two seemingly different viewpoints (Beznosov, 2003; Beznosov & Kruchten, 2004). Discussions of agile security assurance to date have been at a very high level and although the works acknowledge the possibility of secure application development within an agile framework, they do not provide the necessary support infrastructure needed to develop secure applications for a specific domain. This paper will propose a strategy for the development of secure Web-based applications within an...
Related Content

Indexing Techniques for Web Access Logs
[www.igi-global.com/chapter/indexing-techniques-web-access-logs/31129?camid=4v1a](www.igi-global.com/chapter/indexing-techniques-web-access-logs/31129?camid=4v1a)

Surveys as a Complementary Method for Web Log Analysis
[www.igi-global.com/chapter/surveys-complementary-method-web-log/21995?camid=4v1a](www.igi-global.com/chapter/surveys-complementary-method-web-log/21995?camid=4v1a)

Third Party Multimedia Streaming Control with Guaranteed Quality of Service in Evolved Packet System

Model-Based System Design Using SysML: The Role of the Evaluation Diagram
[www.igi-global.com/chapter/model-based-system-design-using-sysml/137351?camid=4v1a](www.igi-global.com/chapter/model-based-system-design-using-sysml/137351?camid=4v1a)