Vulnerability Analysis of Web-Based Applications

Marco Cova, Viktoria Felmetsger and Giovanni Vigna

Reliable Software Group, Department of Computer Science, University of California, Santa Barbara
[marco, rusvika, vigna]@cs.ucsb.edu

Abstract. In the last few years, the popularity of web-based applications has grown tremendously. A number of factors have led an increasing number of organizations and individuals to rely on web-based applications to provide access to a variety of services. Today, web-based applications are routinely used in security-critical environments, such as medical, financial, and military systems.

Web-based systems are a composition of infrastructure components, such as web servers and databases, and of application-specific code, such as HTML-embedded scripts and server-side CGI programs. While the infrastructure components are usually developed by experienced programmers with solid security skills, the application-specific code is often developed under strict time constraints by programmers with little security training. As a result, vulnerable web-based applications are deployed and made available to the whole Internet, creating easily exploitable entry points for the compromise of entire networks.

To ameliorate these security problems, it is necessary to develop tools and techniques to improve the security of web-based applications. The most effective approach would be to provide secure mechanisms that can be used by well-trained developers. Unfortunately, this is not always possible, and a second line of defense is represented by auditing the application code for possible security problems. This activity, often referred to as web vulnerability analysis, allows one to identify security problems in web-based applications at early stages of development and deployment.

Recently, a number of methodologies and tools have been proposed to support the assessment of the security of web-based applications. In this chapter, we survey the current approaches to web vulnerability analysis and we propose a classification along two characterizing axes: detection model and analysis technique. We also present the most common attacks against web-based applications and discuss the effectiveness of certain analysis techniques in identifying specific classes of flaws.

13.1 Introduction

The World Wide Web started in the mid-1990s as a system to support hyper-textual access to static information and has since then evolved into a full-fledged platform for the development of distributed applications. This
has been made possible by the introduction of a number of mechanisms that can be used to trigger the execution of code on both the client and the server side. These mechanisms are the basis to implement web-based applications.

As the use of web applications for critical services has increased, the number and sophistication of attacks against web application has grown as well. A series of characteristics of web-based applications make them a valuable target for an attacker. First, web applications are often designed to be widely accessible. Indeed, by design, they are almost always reachable through firewalls and a significant part of their functionality is available to anonymous users. Because of this, they are considered the most effective entry point for the compromise of computer networks. Second, web-based applications often interface with back-end components, such as mainframes and product databases, that might contain sensitive data, such as credit card information. Therefore, they become an attractive target for attackers who aim at gaining a financial profit. Third, the technology used to implement, test, and interact with web-based applications is inexpensive, well known, and widely available. Therefore, attackers can easily develop tools that expose and automatically exploit vulnerabilities.

Other factors contribute to make web applications a preferred target for attackers. For example, some of the most popular languages used to develop web-based applications are currently easy enough to allow novices to start writing their own applications, but, at the same time, they do not provide a comprehensive, easy-to-use set of mechanisms that support the development of secure applications. This problem is particularly difficult to solve. In fact, while the infrastructure components, such as web servers and browsers, are usually developed by experienced programmers with solid security skills and reviewed by a large developer team, the application-specific code is often developed under strict time constraints by few programmers with little security training. As a consequence, vulnerable code is made available on the web.

This trend is confirmed by various statistics. In the first semester of 2005, Symantec cataloged 1,100 new vulnerabilities, which represent well over half of all new vulnerabilities, as affecting web-based applications. This is a 59% increase over the previous semester, and a 109% increase over the same period of the previous year [33].

An analysis of the reported vulnerabilities shows various types of problems. Web applications can be affected by flaws that are not web specific and that have been commonly found also in traditional applications. Examples of such problems include broken authentication and authorization management, where account credentials and session tokens are not properly protected; improper handling of errors or exceptional conditions, which leads to the leaking of confidential information or to unexpected system behavior.

In addition to these well-known security problems, web-based applications are affected by a number of vulnerabilities that are specific of the web environment. Some vulnerabilities are due to architectural choices, such as the use of relational databases as back-ends for long-term storage, which lead