Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Securing web applications from injection and logic vulnerabilities: Approaches and challenges

G. Deepa[*], P. Santhi Thilagam

Department of Computer Science and Engineering, National Institute of Technology Karnataka, Surathkal, India

ARTICLE INFO

Article history:
Received 6 June 2015
Revised 17 February 2016
Accepted 18 February 2016
Available online 27 February 2016

Keywords:
SQL injection
Cross-site scripting
Business logic vulnerabilities
Application logic vulnerabilities
Web application security
Injection flaws

ABSTRACT

Context: Web applications are trusted by billions of users for performing day-to-day activities. Accessibility, availability and omnipresence of web applications have made them a prime target for attackers. A simple implementation flaw in the application could allow an attacker to steal sensitive information and perform adversary actions, and hence it is important to secure web applications from attacks. Defensive mechanisms for securing web applications from the flaws have received attention from both academia and industry.

Objective: The objective of this literature review is to summarize the current state of the art for securing web applications from major flaws such as injection and logic flaws. Though different kinds of injection flaws exist, the scope is restricted to SQL Injection (SQLI) and Cross-site scripting (XSS), since they are rated as the top most threats by different security consortiums.

Method: The relevant articles recently published are identified from well-known digital libraries, and a total of 86 primary studies are considered. A total of 17 articles related to SQLI, 35 related to XSS and 34 related to logic flaws are discussed.

Results: The articles are categorized based on the phase of software development life cycle where the defense mechanism is put into place. Most of the articles focus on detecting the flaws and preventing the attacks against web applications.

Conclusion: Even though various approaches are available for securing web applications from SQLI and XSS, they are still prevalent due to their impact and severity. Logic flaws are gaining attention of the researchers since they violate the business specifications of applications. There is no single solution to mitigate all the flaws. More research is needed in the area of fixing flaws in the source code of applications.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Over the years, web application has evolved from a simple, static, and read-only system to a complex, dynamic, and interactive system that provides information and service to the users. Web applications have become an integral part of the daily life since they are freely available and accessible from any machine through the Internet. They often handle sensitive data, and are being used for carrying out critical tasks such as banking, socializing, online shopping and online tax filing. However, web applications have become a prime target for attackers due to their ease of use, omnipresence, demand and growing user-base.

Fig. 1 illustrates the most widely used three-tier architecture of a web application along with the software components and technologies involved in each tier. The advancements in architecture and technologies to provide sophisticated functionalities increase the complexity of the web application, and make them more prone to various attacks. The evolving technologies fail to consider security of the application due to the following factors: (i) the availability of business processing logic on the client-side, for reducing the interaction between client and server, assists the attacker in gaining more knowledge about the web application in order to trigger an attack against the end-user, (ii) the limited security support offered by the current widely used application development frameworks such as Django, Ruby on Rails, etc. makes them prone to attacks, even though the frameworks favor easy and quick

[*] Corresponding author. Tel.: +918951261510.
E-mail addresses: gdeepabalu@gmail.com, ganesan.deepa@yahoo.in (G. Deepa), santhisocrates@gmail.com (P.S. Thilagam).

http://dx.doi.org/10.1016/j.infsof.2016.02.005
0950-5849/© 2016 Elsevier B.V. All rights reserved.

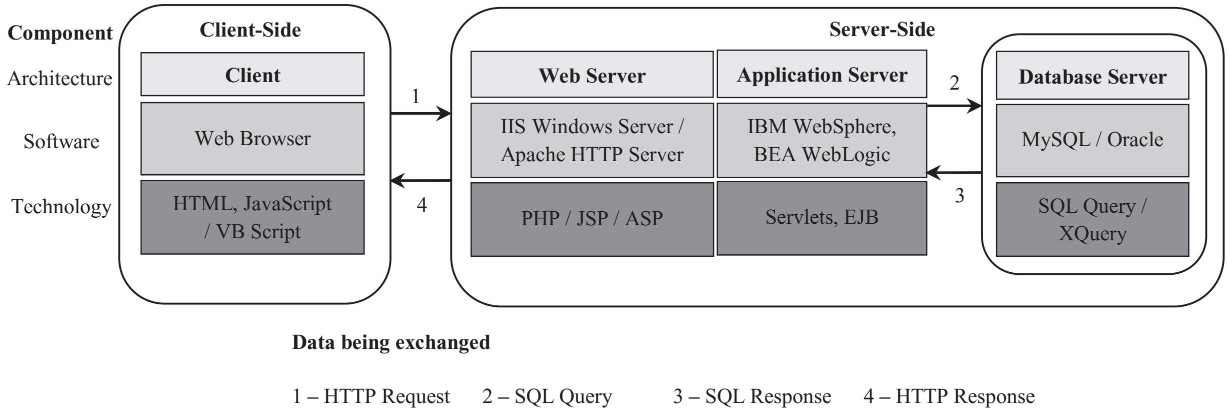

Fig. 1. Web application architecture.

Table 1
Data breaches in the recent years.

Year	Company	Data breach
2015	Bitcoin exchange [3]	5 million dollars
2015	Premera Blue Cross [4]	Personal information of users
2014	Healthcare system [5]	Personal information of patients
2014	eBay [6]	Personal information of active users
2012	Bitcoin exchange [7]	24,000 Bitcoins
2011	Sony Corporation [8]	Login credentials
2009	Heartland payment systems [9]	Credit card information

implementation of the application, (iii) the interoperability and openness of XML used for providing interaction between heterogeneous web applications make them an easy target for attackers, and (iv) web applications are implemented by developers focusing on implementing the features and functionality of the application rather than the security aspects. As a result, existing web applications are more vulnerable to attacks, and the exploitation of these vulnerabilities compromises the confidentiality, integrity and availability of data.

The security breach reports from various organizations signify the importance towards securing web applications. According to Verizon's Data Breach Investigation Report [1], 35% of the security incidents in 2013 were due to attacks on web applications. A report by the Identity Theft Resource Center (ITRC) [2] states that the reported number of breaches increased by 27.5% in 2014 as compared to the previous year, and most of them targeted business, military, banking and medical applications. Table 1 lists some of the data breaches reported in the recent years.

These breaches occur due to attacks that propagate through the weakness in the application itself rather than the weakness in the network. These weaknesses are referred to as software security vulnerabilities, and arise due to implementation defects or design flaws in the programming language, development framework, architecture and code library (i.e. APIs) [10]. The conventional security measures such as Secure Socket Layer, cryptographic techniques, etc. employed for protecting the web applications ensure the security of online network traffic and message in transit, and do not protect them against attacks that exploit vulnerabilities existing in the application [11].

Web application security has attracted much attention from both academia and industry. A substantial amount of research efforts has been dedicated in the past to secure web applications by preventing vulnerabilities and extenuating attacks. Injection and logic vulnerabilities are ranked as the most potent vulnerabilities affecting the security of web applications as reported in OWASP [12], SANS Institute [13], and Trustwave [14]. Hence, it has become necessary to analyze the recent literature addressing these two flaws, and build a comprehensive knowledge-base, which would help to identify future research directions in this domain. This paper mainly aims to explore the following:

- It discusses various kinds of vulnerabilities and attacks that exploit these vulnerabilities in web applications
- It analyzes the pros and cons of mitigation approaches available for securing web applications from injection and business logic vulnerabilities
- It provides information on the capabilities of existing vulnerability scanners, and the challenges faced by them
- It highlights the open-source web applications that can be used for testing and evaluation.

The remainder of the paper is organized as follows. In Section 2, the different types of web application vulnerabilities and attacks are discussed. Section 3 describes the related work. The method employed for conducting the review is summarized in Section 4. Section 5 reviews the approaches devoted to detection and prevention of injection and business logic vulnerabilities. Section 6 presents information on existing commercial and open-source vulnerability scanners, and Section 7 highlights the available open-source web applications. The paper is concluded in the last section.

2. Background

This section describes the various types of vulnerabilities that lead to a variety of attacks on web applications.

2.1. Classification of vulnerabilities

A vulnerability is a flaw in the application that stems from coding defects, and causes severe damage to the application upon exploitation. These vulnerabilities could be exploited by injecting malicious code into input supplied by a user for interacting with the application. The malicious code may violate the syntactic and semantic restrictions imposed on user-input, issue queries at inappropriate application states, and modify the HTTP responses for capturing session information of the users. The malicious input propagates through the application due to the existence of implementation flaws and results in attacks. Majority of the attacks are possible due to the following implementation flaws: improper input validation, improper authentication and authorization mechanisms, improper management of session information, and other implementation bugs that compromise the intended functionality of the application [12,13,15–18].

Improper input validation refers to absence of validation or erroneous validation of input supplied by a user through user

Download English Version:

https://daneshyari.com/en/article/550479

Download Persian Version:

https://daneshyari.com/article/550479

Daneshyari.com