Detecting and Defeating SQL Injection Attacks

Sangita Roy, Avinash Kumar Singh and Ashok Singh Sairam, *senior member IACSIT*

***Abstract*—The increasing dependence on web applications have made them a natural target for attackers. Among these attacks SQL Injection Attacks (SQLIA) are the most prevalent. In this paper we propose a SQL injection vulnerability scanner that is light-weight, fast and has a low false positive rate. These scanners prove as a practical tool to discover the vulnerabilities in a web application as well as to test the efficiency of counter attack mechanisms. In the latter part of our work we propose a security mechanism to counter SQL Injection Attacks. Our security methodology is based on the design of a filter for the HTTP request send by clients or users and look for attack signatures. The proposed filter is generic in the sense that it can be used with any web application. Finally we test our proposed security mechanism using the vulnerability scanner developed by us as well as other well known scanners. The proposed security mechanism is able to counter all the vulnerabilities that were previously reported before the deployment of our security framework.**

***Index Terms*—SQL Injection Attacks, URL filter, Web Application Vulnerability Scanner.**

I. Introduction

Web applications such as blogs, social network, webmail, bank etc have become our way of life. The omnipresence of web applications has made them a natural target for malicious minds. Web applications are susceptible to a number of vulnerabilities which can be due to a design flaw or an implementation bug. Among the top ten web application vulnerabilities published by Open Web Application Security Project [10], SQL Injection Attack (SQLIA) is the most vulnerable. According to OWASP, SQL injection vulnerabilities were reported in 2008, making up 25% of all reported vulnerabilities for web applications. An SQLIA occurs when an attacker changes the intended effect of an SQL query by inserting (or injecting) new SQL keywords or operators into the query thereby gaining unauthorized access to a database in order to view or manipulate restricted data. The root cause of SQLIA is insufficient user input validation. Although there is an increasing awareness about security, there are several significant factors that make securing web applications difficult. First web applications are growing at a frantic pace largely fuelled by the simplicity with which one can develop such applications using the numerous tools available. Secondly the developers and administrators do not have the requisite knowledge and experience in the area of security.

A logical approach to tackle the problem of SQLIA is to scan the vulnerabilities present in a webpage and subsequently launch attack counter measure tools. There are a number of open-source as well as commercial tools called Web Application Vulnerability Scanners [7, 8, 9] that perform security testing as well as assessment and finally report the vulnerabilities present. In spite of their continuous evolution,these automated scanners still have some problem with regard to the high number of undetected vulnerabilities and high percentage of false positives [2]. A web vulnerability scanner is not a panacea but an useful tool to access the security of web applications. The methodology proposed in this paper is to first scan a webpage in a controlled environment and discover the vulnerabilities present. Next we provide a framework to prevent SQLIA. To test the performance of our security framework we again run the vulnerability scanner after its implementation. Empirical results on a realistic environment show that our counter security mechanism is able to prevent all the vulnerabilities previously reported by the scanner.

The remainder of the paper is organized as follows. In section II we study existing scanners and review their performance. Section III presents overview of our proposed web vulnerability scanner named CSRScanner and section IV describes the implementation details of CSRScanner and in section V we have done the analysis of our designed tool. In section VI we perform a detailed analysis of the different types of known SQLI attacks and identify a unique pattern or signature for each. Based on these signatures we propose a URL filter to prevent SQLI attacks in section VII. In Section VIII we have checked the effectiveness of our filter approach by using our CSRScanner tool. Finally we conclude the paper in section IX.

II. Web Application Vulnerability Scanners

Web application vulnerability scanners are designed to test security mechanisms applied to web applications [4]. The general methodology of these scanners is based on the discovery of *vulnerable spots* that is positions in the HTTP request where an attacker can inject maliciously crafted SQL codes. In the second step the tool performs a controlled exploit of the vulnerabilities at these vulnerable spots. Finally it verifies the success of the attack and reports the result. The vulnerability scanners are designed such that they perform the same attack as one would do manually and hence they provide a practical environment to test counter measure mechanisms against SQLIA.

A host of vulnerability scanners both commercial as well as open-source are available. A brief review of some of the best known scanners is given below.

- Acunetix([8]): automatically checks web applications for vulnerabilities such as SQL Injections, cross site scripting, arbitrary file creation/deletion and weak password strength

Manuscript received June 16, 2011; revised June 30, 2011.

Sangita Roy is with the Indian Institute of Technology, Patna, Bihar, India (e-mail: r_sangita@ iitp.ac.in).

Avinash Kumar Singh is with the Gwalior Engineering College, Gwalior, MP , India(e-mail: avinashkumarsingh1986@ gmail.com).

Ashok Singh Sairam is with the Indian Institute of Technology, Patna, Bihar, India (e-mail: ashok@iitp.ac.in).