LDAP Vulnerability Detection in Web Applications

Hossain Shahriar, Kennesaw State University, Marietta, USA
Hisham Haddad, Kennesaw State University, Marietta, USA
Pranahita Bulusu, Kennesaw State University, Marietta, USA

ABSTRACT

Lightweight Directory Access Protocol (LDAP) is commonly used in web applications to provide lookup information and enforcing authentication. Web applications may suffer from LDAP injection vulnerabilities that can lead to security breaches such as login bypass and privilege escalation. This paper proposes OCL fault injection-based detection of LDAP injection attacks. The authors extract design-level information and constraints expressed in OCL and then randomly alter them to generate test cases that have the capability to uncover LDAP injection vulnerabilities. The authors proposed approaches to implement test case generation, and they used one open source PHP application and one custom application to evaluate the proposed approach. The analysis shows that this approach can detect LDAP injection vulnerabilities.

KEYWORDS

Fault Injection, LDAP, LDAP Query Injection, Object Constraint Language (OCL)

1. INTRODUCTION

LDAP (“Introduction to Lightweight,” n.d.) is used in a large number of web applications. However, poorly implemented LDAP applications suffer from injection vulnerabilities where user inputs are not validated properly (“Open Web Application,” n.d.). An attacker can exploit vulnerabilities by providing malicious inputs and change intended operations through altered LDAP queries. A vulnerable application cannot differentiate a malicious query generated based on attacker’s inputs and a legitimate query generated based on benign inputs. Altered LDAP queries may lead to unwanted activities such as login bypass and disclosure of sensitive information. For example, LDAP injection vulnerability was discovered in *Events Planner - SmarterMail* (Hoyt, n.d.), where input type parameters can be provided to alter a disjunctive (OR) query to conjunctive (AND) query. Many security threats arise when a default configuration of LDAP server is used (“Vulnerable Applications for LDAP Injection,” n.d.).

Various research works have been conducted for the prevention of injection attacks (Coldwind et al., n.d.; DuPaul, 2015; Hafiz & Johnson, 2009; Hafiz, n.d.; Xie, Chu, Lipford & Melton, 2011; Zheng & Zhang, 2013; Almorsy, Grundy, & Ibrahim, 2012). Some of the existing approaches include the administrator and developer techniques and program transformation technique. However, these mitigation techniques have not been designed specifically to prevent LDAP injection attacks. Past

DOI: 10.4018/IJSSE.2017100102

Copyright © 2017, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
research work that focus on code injection vulnerabilities may not be directly applied to mitigate LDAP injection vulnerabilities due to syntactic and semantic differences with other known vulnerabilities like SQL injection. Moreover, unlike SQL injection, LDAP injection vulnerabilities may be exploited by an insider employee of the organization.

In this paper, we propose fault injection-based detection of LDAP query injection attacks. Extract the design-level information and constraints for various functionalities, relying on LDAP query, expressed in Object Constrained Language (OCL) notation. We propose an approach to alter constraints and generate appropriate test cases that produce dissimilar results between original constraints and altered constraints. We show the application of the proposed approach for detecting common types of LDAP query injection attacks using PHP web application. The approach has been found effective in detecting a prior reported LDAP injection vulnerabilities.

The paper is organized as follows: Section 2 provides an overview of LDAP and injection attacks. Section 3 describes related work. Section 4 discusses the proposed approach. Section 5 shows the evaluation of the proposed approach using a PHP web application. Finally, Section 6 concludes the paper.

2. BACKGROUND ON LDAP INJECTION ATTACK

Figure 1 shows an example of LDAP directory tree structure. The tree is subdivided into different Organizational Units (ou) along with common names (cn) for each of them.

For example, the organizational unit of Human Resources has common name as HR. Different entries of each organizational unit are given under the common name such as the employees working in a particular department and any document relevant to the particular department.

Figure 1. Directory tree structure of LDAP server
Threatening the Cloud: Securing Services and Data by Continuous, Model-Driven Negative Security Testing
[www.igi-global.com/chapter/threatening-cloud-securing-services-data/72236?camid=4v1a](www.igi-global.com/chapter/threatening-cloud-securing-services-data/72236?camid=4v1a)

An SMT-based Approach for Generating Coverage Oriented Metamodel Instances
[www.igi-global.com/article/an-smt-based-approach-for-generating-coverage-oriented-metamodel-instances/170518?camid=4v1a](www.igi-global.com/article/an-smt-based-approach-for-generating-coverage-oriented-metamodel-instances/170518?camid=4v1a)