[FRTSSDS- June 2018]
DOI: 10.5281/zenodo.1288417

ISSN 2348 – 8034
Impact Factor- 5.070

GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

PREVENTING LDAP INJECTION FROM MALICIOUS ATTACKS

Rajesh Shah & Maya Rathore
Department of Computer Science, Christian Eminent College, Indore

ABSTRACT

It is common in web application to provide interactions between end-users and databases. These applications may be at risk to code injection attacks which come first of weak validation in the parameters that the users use. If one of these applications accept input from a client and execute it without first validating it, attackers have the potential to execute their own queries and thus extract sensitive information from the lightweight directory access protocol (LDAP) directory. LDAP services are the key components in companies. It is an Internet standard for accessing directory services.

Lightweight Directory Access Protocol (LDAP) is an open-standard protocol for both querying and manipulating X.500 directory services. The LDAP protocol runs over Internet transport protocols, such as TCP. Web applications may use user-supplied input to create custom LDAP statements for dynamic web page requests. LDAP defines a standard method for accessing and updating information in a directory. LDAP has gained wide acceptance as the directory access method of the Internet and is therefore also becoming planned within corporate intranets. It is being supported by a growing number of software vendors and is being incorporated into a growing number of applications. For example, the two most popular web browsers, Netscape Navigator and Microsoft Internet Explorer support LDAP functionality as a base feature.

But sometimes it has been observed that our critical data may not be safe. These data can be hacked by unauthenticated user through the different techniques. One of these techniques is Lightweight Directory Access Protocol (LDAP) injection. LDAP injection is an attack technique used to exploit web sites that construct LDAP statements from user-supplied input. In this paper we explain about LDAP injection technique and the various problems associated with it. It also discusses some prevention methods through which you can secure your web application from LDAP injection.

Keywords- *SQL injection, X.500, LDAP directory, RBAC*

I. INTRODUCTION

Although the awareness is increasing day by day and it is common in web application to provide interaction between end users and databases. But sometimes these applications may be at risk to code injection attacks which occurs due to weak validation in the parameters used by users. If your web application accepts input from the client and executes it without validating it then there is a possibility for attackers to execute their own queries and thus extract the sensitive information from the LDAP directory. Preventing the consequences of these kinds of attacks requires the study of different code injection possibilities and in making them public and well known for all programmers and administrators. In this paper the LDAP injection techniques are analyzed in detail, because all the web applications based on LDAP tree might be vulnerable to these kinds of attacks.

The key to exploiting the injection techniques with LDAP is to manipulate the filters used to search in the directory services. Using these techniques, an attacker may obtain direct access to the database underlying an LDAP tree, and thereby to important corporate information. An LDAP injection attack still poses a serious security exposure.

This paper discusses what LDAP injection is all about, as well as the preventions to protect secure our important data from these attacks.

ISSN 2348 – 8034
Impact Factor- 5.070

LDAP

The Lightweight Directory Access Protocol (LDAP) provides a mechanism for connecting to, searching, and modifying internet directories. LDAP statements (or Queries) used to retrieve data from information directories.

LDAP Injection

LDAP injection is a specific form of attack that can be used to compromise web sites that construct LDAP (Lightweight Directory Access Protocol) statements from data provided by users. This is done by changing LDAP statements, so dynamic Web applications can run with invalid permissions, allowing the attacker to alter, add or delete content. LDAP is a protocol that facilitates the location of organizations, individuals and other resources in a network. It is a streamlined version of DAP (Directory Access Protocol), which is part of X.500, a standard for network directory services.

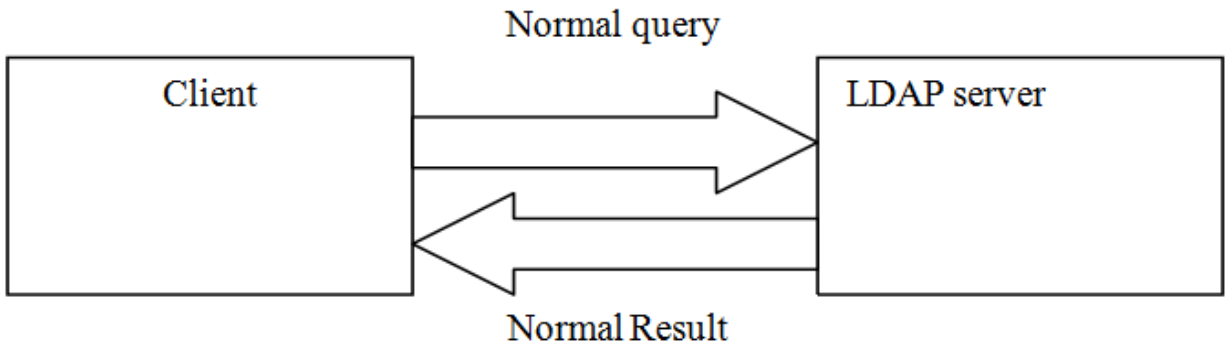
LDAP injection works in much the same manner as SQL injection, a type of security exploit in which the attacker adds SQL (Structured Query Language) code to a Web form input box to gain access to resources or make changes to data. The main reason that LDAP injection and similar exploits are on the rise is the fact that security is not sufficiently emphasized in application development. To protect the integrity of Web sites and applications, simple precautions during development must be implemented, such as controlling the types and numbers of characters that are accepted by input boxes.

LDAP injection in web applications

LDAP injection attacks are based on similar techniques to SQL injection techniques, Therefore the primary concept is to take advantage of the parameter introduced by the user to generate the LDAP query. A secure web application should cleanse the parameters introduced by the user before constructing and sending the query to the server. In a vulnerable environment these parameters are not properly filtered and the attacker can inject malicious code.

II. DISCUSSION

The typical test to know if an application is vulnerable to code injection consists of sending to the server a query that generates an invalid input. Therefore if the server returns an error message, it is clear for the attacker that the server has executed the query and that he/she can exploit the code injection techniques.



ISSN 2348 – 8034
Impact Factor- 5.070

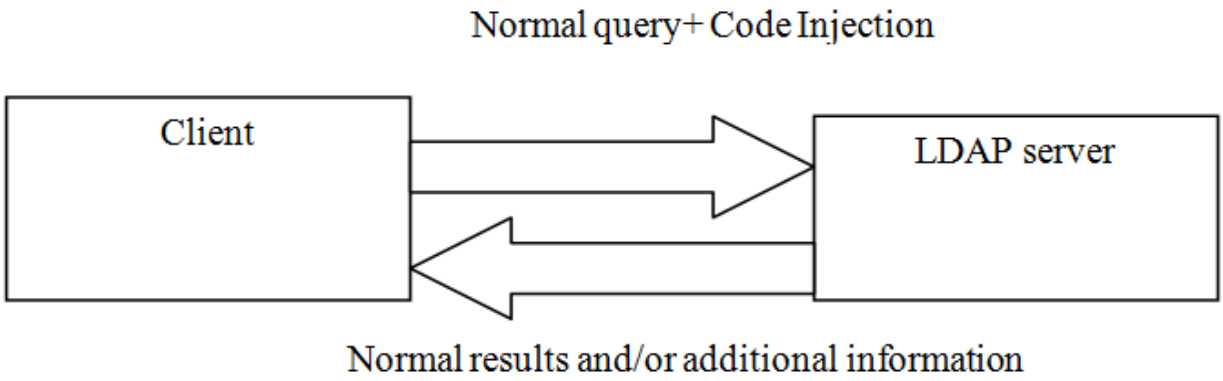


Fig 1. LDAP Injection

Detecting LDAP Injection:
The most common way to detect the LDAP Injection attack is:
Identify entry points that collect user input such as text boxes, query string parameters, etc.
Insert any character ('(', '|', '&') as an input and submit the request.
Identify whether an exception/error message was generated relating to LDAP (e.g., Page cannot be displayed).

Preventing LDAP injection
Protecting LDAP-enabled web applications demands the effort of developers as well as the LDAP administrators. LDAP injection is an application-specific vulnerability that commonly occurs due to missing or weak input validation functions prior to processing and allowing persistence of data in LDAP. This weakness would allow a hacker to use malicious LDAP attributes to inject or manipulate or steal personal information from an LDAP repository. LDAP injection is also possible due to exploits of an insecure LDAP lookup configuration (using "Directory Manager") and missing LDAP access control policies.

To prevent LDAP injection, it is suggested to enforce strict input validation functions before processing data for LDAP persistence. In the case of an application that relies on client-side data validation, it becomes important to re-verify and validate them on the server side as well. The data validation should verify the input in terms of required LDAP attributes and its known data type, meta characters, format, length, legal values, etc. To prevent issues with insecure LDAP configuration and access control policies, it is often suggested to verify LDAP configuration and enforce principle of least privilege and role-based access control (RBAC) policies.

The different ways through which you can prevent from LDAP injections are:

Incoming Data Validation

a) All client-supplied data needs to be cleaned of any characters or strings that could possibly be used maliciously. This should be done for all applications, not just those that use LDAP queries.
b) Stripping quotes or putting backslashes in front of them may not be enough. The best way to filter data is with a default deny regular expression that includes only the type of characters that you want.
c) If you need to include symbols or punctuation of any kind, make absolutely sure to convert them to HTML substitutes (such as "&quote; " or " > ").For instance, if the user is submitting an email address, allow only the "at" sign, underscore, period, and hyphen in addition to numbers and letters, and only after those characters have been converted to their HTML substitutes.

Outgoing Data Validation
All data returned to the user should be validated and the amount of data returned by the queries should be restricted as an added layer of security.

ISSN 2348 – 8034
Impact Factor- 5.070

LDAP Configuration

Implementing tight access control on the data in the LDAP directory is imperative, especially when configuring the permissions on user objects and even more importantly if the directory is used for single sign-on solution. Understand how each object class is used and decide if the user should be allowed to modify it. Allowing users to modify their uid Number attribute may let the user change access levels when accessing systems. The access level used by the Web application to connect to the LDAP server should be restricted to the absolute minimum required. That way, even if an attacker manages to find a way to break the application, the damage would be limited.

III. CONCLUSION

LDAP services facilitate access to networks information organizing it in a hierarchical database that allows authorized users and applications to find information related to people, resources and applications. LDAP injection techniques are an important threat for these environments, specially, for the control access and privileges and resources management. These attacks modify the correct LDAP queries, altering their behavior for the attacker benefit.

It is very important to filter the variables used to construct the LDAP queries before sending them to the server. In this paper the various security problems of LDAP injection are discussed and keeping it in mind the preventive measures have been suggested like incoming data validation, All data returned to the user should be validated and the amount of data returned by the queries should be restricted as an added layer of security.

REFERENCES

[1] LDAP Injection-Are your web application vulnerable, Sacha Faust.
[2] LDAP Injection vs. Blind LDAP Injection, José María Alonso1.
[3] LDAP Injection and Blind LDAP Injection in web application, Chema Alonso