Information Systems 56 (2016) 135–156

Contents lists available at ScienceDirect

Information Systems

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

Preventing database schema extraction by error message handling

Sepideh Naghdi, Morteza Amini

Sharif University of Technology, Tehran, Iran

ARTICLE INFO

Article history:
Received 15 November 2014
Received in revised form
16 September 2015
Accepted 29 September 2015
Recommended by: F. Naumann
Available online 8 October 2015

Keywords:
Error handling
Database security
Database schema extraction
Error message modification

ABSTRACT

Nowadays, a large volume of an organization's sensitive data is stored in databases making them attractive to attackers. The useful information attackers try to obtain in the preliminary steps, is the database structure or schema. One of the popular approaches to infer and extract the schema of a database is to analyze the returned error messages from its DBMS. In this paper, we propose a framework to handle and modify the error messages automatically in order to prevent schema revealing. To this aim, after identifying and introducing an appropriate set of categories of error messages, each error message that is returned from a DBMS is placed in a proper category. According to the policy specified for each category, corresponding rules are applied for removing/modifying/obfuscating the sensitive data in the error messages of that category before submitting them to the application. The general way proposed to determine the category of an error message is employing the keyword based categorization approach, which is 95% accurate for Microsoft SQL Server 2012.

1. Introduction

Databases are attractive to attackers because they store sensitive data of organizations. Different types of attacks with different purposes are executed against the databases. Attackers attack the DBMSs directly, or indirectly through cracking the applications [1]. The *database structure or schema* is generally the useful information that attackers try to obtain in the preliminary steps of an attack against the databases. One of the more popular attacks to extract the database structures and access its stored data is the *SQL injection* attack. DBMSs are used as the infrastructure of many applications. If an application gets some inputs from its users and directly generates the required SQL queries with them (without any validation) and sends the queries to the DBMS, it will be vulnerable to the SQL injection attacks.

There are different techniques to form SQL injection attacks against a database, where *Inference* and *Illegal/Logically Incorrect Queries* techniques are used to extract the database schema. In inference attacks, attackers ask some questions with true or false responses and obtain the structural information according to the changes in the operation of the application. In many situations, when certain types of errors occur in the DBMSs, predefined or user-defined error messages are displayed to the users at the application level. In Illegal/Logically Incorrect Query attack, the attackers send the queries to a DBMS which causes an error to be raised and then they obtain the structural information from the content of error messages that are returned from the DBMS [2].

E-mail addresses: snaghdi@ce.sharif.edu (S. Naghdi), amini@sharif.edu (M. Amini).

http://dx.doi.org/10.1016/j.is.2015.09.010
0306-4379/

The main goal of this research is to deal with the Illegal/Logically Incorrect Query or similar attacks which reveal sensitive information about a database by creating error conditions. The error messages that are defined in the DBMSs for error and exception conditions, help the database administrators and application developers to debug and audit the error conditions. However, displaying them to the application end-users is not necessary or secure. As a result, to prevent structural information being revealed via error messages, we should prevent the original error messages from being displayed to the users and instead handle and modify/customize them.

Error handling can be done at two levels, the DBMS level and application level. It is also possible to handle the errors in middlewares, such as database firewalls, which are developed for similar purposes. If the features and methods developed for preventing SQL injection attacks are not employed correctly in an application and the raised errors are not handled correctly, the application and its back-end DBMS are vulnerable to Illegal/Logically Incorrect Query attacks. Actually, none of the current solutions against this sort of injection attacks are suitable to protect the vulnerable applications. In this paper, we propose a framework to automatically customize the error messages and prevent disclosure of structural information via the error messages. To achieve this goal, after identifying and introducing an appropriate set of categories of error messages, each error message that is returned from a DBMS is placed in the proper category. According to the sensitive data that exists in the error messages of each category, some rules are defined to be leveraged for modifying or obfuscating the error messages of the category before submitting them to the application.

In fact, the proposed approach can be useful for systems where there is no security mechanism at either application or database level to prevent showing error messages to the end-users. Hence the whole system is vulnerable to revealing the database schema via the returned error messages from the DBMS. As an example, consider a web application which is connected to MSSQL, when an attacker inserts the following text in a field of the web site which needs number values [2].

```
"CONVERT(int,(SELECT TOP 1 name FROM sysobjects WHERE xtype='u'))"
```

If the application does not perform input validation, the following query is formed in the application and is sent to the back-end MSSQL.

```
SELECT id FROM customers WHERE username='' AND password='' AND id=convert (int,(SELECT
top 1 name FROM sysobjects WHERE xtype='u'))
```

In this query, the name of the first user-defined object is extracted from the system table "sysobject" and is converted into an integer number. Since this conversion is invalid, an error message is created by MSSQL and the following error message is shown to the end-user by the web application.

```
"Microsoft OLE DB Provider for SQL Server (0x80040E07) Conversion failed when
converting the nvarchar value
'CustomerDetails' to data type int."
```

So the displayed error message reveals the DBMS type (here, MSSQL) and the name of the first user-defined object in the database (here, CustomerDetails). In this attack example, there is no security action in DBMS or the application to prevent showing the user the generated error message. In this condition, a system based on our proposed approach can be involved to automatically modify the error messages to the benign and secure ones. Here, a secure error message means an error message without any sensitive information about the database schema; however it would contain the minimum information required for debugging purposes. As an instance, the above error message string would be changed to the following one:

```
"Error generated because of objects or constraints on a table."
```

In the rest of this paper, in Section 2, error handling in software systems and the current solutions to deal with SQL injection attacks and especially Illegal/Logically Incorrect Query attacks are reviewed. Section 3 describes our proposed framework for handling the error messages generated by attackers. In Section 4, a general approach to categorize the error messages is introduced. The evaluation of the accuracy, effectiveness, and response time of the proposed method is presented in Section 5. Finally, Section 6 concludes the paper.