Pune Institute of Computer Technology
Dhankawadi, Pune

Mini Project Report
on
Sql Injection Prevention System

SUBMITTED BY

Chetan Atole 41307
Prem Bansod 41310
Shailesh Borate 41315

DEPARTMENT OF COMPUTER ENGINEERING

Academic Year 2020-2021

Contents

1 Introduction

A SQL injection attack consists of insertion or "injection" of a SQL query via the input data from the client to the application. A successful SQL injection exploit can read sensitive data from the database, modify database data (Insert/Update/Delete), execute administration operations on the database (such as shutdown the DBMS), recover the content of a given file present on the DBMS file system and in some cases issue commands to the operating system. SQL injection attacks are a type of injection attack, in which SQL commands are injected into data-plane input in order to affect the execution of predefined SQL commands.

SQL Injection attacks allow attackers to spoof identity, tamper with existing data, cause repudiation issues such as voiding transactions or changing balances, allow the complete disclosure of all data on the system, destroy the data or make it otherwise unavailable, and become administrators of the database server. SQL Injection is very common with PHP and ASP applications due to the prevalence of older functional interfaces. Due to the nature of programmatic interfaces available, J2EE and ASP.NET applications are less likely to have easily exploited SQL injections. The severity of SQL Injection attacks is limited by the attacker's skill and imagination, and to a lesser extent, defense in depth countermeasures, such as low privilege connections to the database server and so on. In general, consider SQL Injection a high impact severity.

An online shop that allows users to check for different cloths for women's available at the online store and can purchase cloths online. The project consists of list of cloths displayed in various materials and designs. The user may browse through these products as per categories. If the user likes a product, he/she can add it to his/her shopping cart. Once user wishes to checkout he must register on the site first. Once the user makes a successful transaction admin will get report of his bought products. The objective of this project is to develop a secure path for transaction done by the user. Using AES (Advanced Encryption Standard) encryption technique, the transaction and user account details can be made secured. AES encryption is also used to encrypt the user's card and password information while transaction.

SQL-injection, depending on the type of database implementation and the conditions may allow the attacker to execute arbitrary database query (e.g., read the contents of any table, remove, change or add data), get the opportunity to read and / or write local files and execution of arbitrary commands on the server. The attack by using SQL-injection becomes possible because of incorrect handling of input used in SQL-queries.

Advantages

1. Secured transaction while doing card payment.
2. Less risk of data getting hacked.
3. The system is very secure and robust in nature.
4. SQL Injection prevention mechanism is used.

Disadvantages

1. Does not keep track of stock/order.
2. System may provide incorrect results if data not entered correctly.

2 Software Requirement Specification

2.1 Project Plan

This project scope is to prevent the commonly occurring SQL injection. Create the system for web systems which help to resolve the sql injections. SQL Injection has become a common issue with database-driven web sites. The flaw is easily detected, and easily exploited, and as such, any site or software package with even a minimal user base is likely to be subject to an attempted attack of this kind.

2.2 Hardware Requirement

1. Computer/Laptop 32/64 bit
2. 4 GB RAM

2.3 Software Requirement

1. Java 8
2. PHP
3. VS Code

3 Result and Output

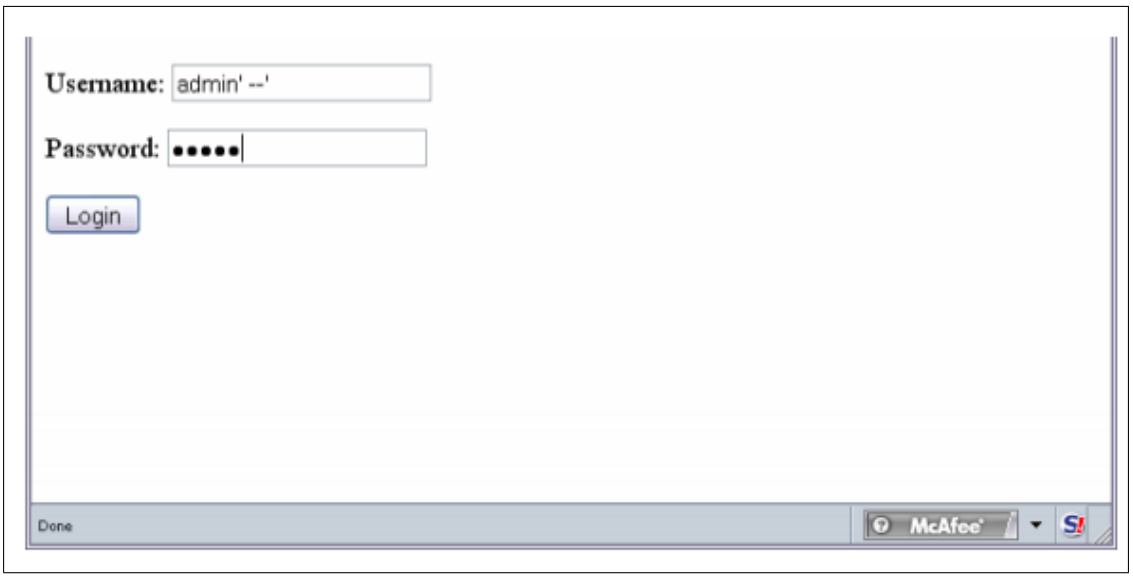

Figure 1: Output 1

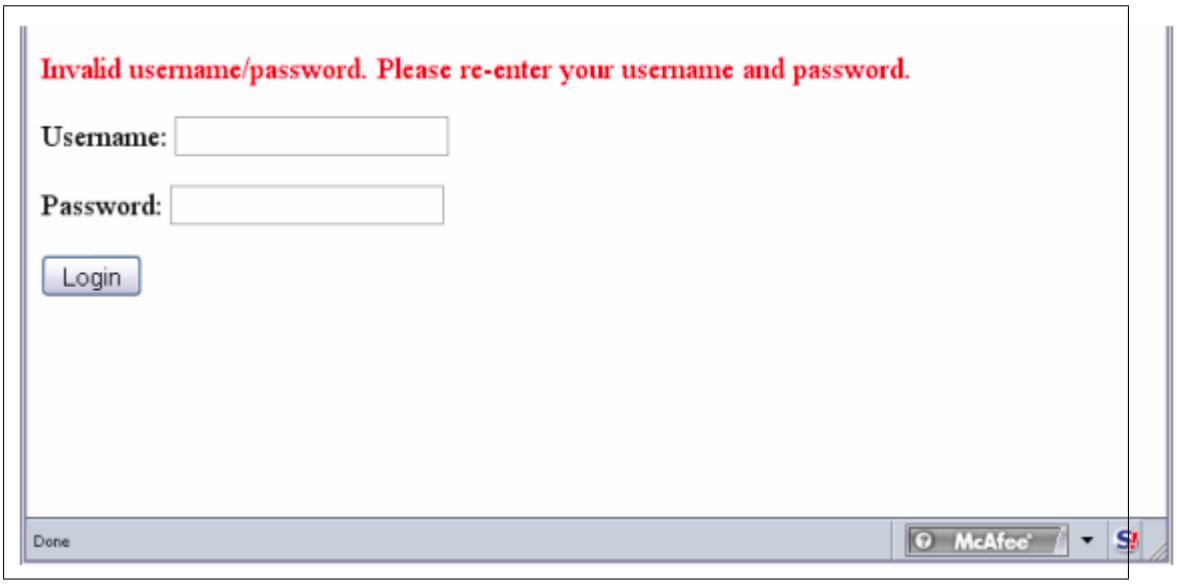

Figure 2: Output 2

Attack Type	Username input	Password input	Result
AND/OR	admin	' OR '1'='1	Blocked
AND/OR	admin	' OR '0'<>'1	Blocked
White Space Manipulation Variation of AND/OR	admin	'OR'1'='1	Blocked
White Space Manipulation Variation of AND/OR	admin	'OR'0'<>'1	Blocked
Comments	admin'--	abcd	Blocked
Comments	admin'#	abcd	Blocked
Comments	admin'/*	abcd	Blocked
Comments	' or 1=1--	abcd	Blocked
Comments	' or 1=1#	abcd	Blocked
Comments	' or 1=1/*	abcd	Blocked
Comments	') or '1'='1--	abcd	Blocked
Comments	') or ('1'='1--	abcd	Blocked
Comments	/*!32302 10*/	abcd	Blocked
Comments	10; DROP TABLE members --	abcd	Blocked
Comments	10; DROP TABLE hackers2; /*	*/ OR 'abcd' <> '	Blocked
Comments	admin'); drop table hackers2; --	abcd	Blocked
Comments	admin'--	' OR '1'='1	Blocked
UNION	' UNION SELECT 1, 'anotheruser', 'doesnt matter', 1--	abcd	Blocked
UNION	' UNION select all from dummy --'	abcd	Blocked
String Concatenation	' UNI' + 'ON' + ' select "test" from dummy'	abcd	Blocked
String Concatenation	' UNI' || 'ON' || ' select "test" from dummy'	abcd	Blocked
String Concatenation	CONCAT(CHAR(65),CHAR(68),CHAR(77), CHAR(73),CHAR(78))	abcd	Blocked
Comments variation of UNION	' UNI/*breakUpThisKeyword*/ON select "test" from dummy'	abcd	Blocked
Hex/Bin/Dec	1 UNION SELECT ALL FROM WHERE	abcd	Blocked
Case-insensitivity, white-space manipulation	admin	U n I o N	Blocked

Figure 3: Results

4 Conclusion

Internet applications that make use of a database system face in terms of security and protection the private data. SQL-I attacks is a legitimate threat that endangers the confidentiality of data and may cost an organization a great deal of money and even their reputationWe believe that SQLrand is a very practical system that solves a problem heretofore ignored, in preference to the more “high profile” buffer overflow attacks. Our plans for future work include developing tools that will further assist programmers in using SQLrand and extending coverage to other DBMS back-ends.

References

[1] O. P. Voitovych, O. S. Yuvkovetskyi and L. M. Kupershtein, "SQL injection prevention system," 2016 International Conference Radio Electronics Info Communications (UkrMiCo), 2016, pp. 1-4, doi: 10.1109/UkrMiCo.2016.7739642.

[2] Aulakh, T. Intrusion Detection and Prevention System: CGI Attacks, 2009. San Jose State University master's thesis project

[3] Webpage "About page for dotDefender from Applicure" http://www.applicure.com/About dotDefender Retrieved on 2009-12-15.

[4] Webpage "Wikipedia.org: SQL" http://en.wikipedia.org/wiki/SQL Retrieved on 2010-03-01.

[5] Webpage "SQL Injection Cheat Sheet" http://ferruh.mavituna.com/sql-injection-cheatsheet-oku Retrieved on 2010-03-01.