Research Journal of Applied Sciences, Engineering and Technology 14(5): 203-208, 2017
DOI:10.19026/rjaset.14.4290
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: February 9, 2017 Accepted: March 23, 2017 Published: May 15, 2017

Research Article

Integrated Password-based Algorithms with Auditing Capability for Database Applications

Ayman Mohamed Mostafa and Faten Ayied Almutairi
Faculty of Computers and Informatics, Zagazig University, 44519, Egypt

Abstract: The aim of this research is to maintain the confidentiality and integrity of database by building a security application used for protecting sensitive information stored in a database from disclosure. The first layer in security application is based on a password-based system. The password serves to authenticate the ID of the user logging on to the system. In order to prevent passwords from offline dictionary attacks and specific account attacks, an encryption process is executed using Message Digest 5 (MD5) hashing and salt hashing algorithms for concealing passwords stored in a database. The salt hashing prevents duplicate passwords from being visible in the password file and increases the difficulty of offline dictionary attacks. If an adversary tries to retrieve the password system data, a different password will be generated which is completely different from the original one. For maintaining the integrity of data, an auditing mechanism is embedded into the password-based system for monitoring all transactions and operations inside the database.

Keywords: Auditing, database security, hashing, salt hashing cryptography

INTRODUCTION

Protecting the privacy of the data is one of the most important criteria to be taken into account when building a new application. This is due to the importance of data and its negative effects on the economy in the event of damage or loss of data stored. As presented in Kumar and Al Hasani (2016), databases has increasingly become the target of cyber criminals which can be lost or stolen, Hence, like data breaches in production environment, non-production environment data breaches can incur significant cost and irreparable harm to reputation and brand. This study presents a password-based algorithm for reducing security breaches in database in addition to protecting users' password stored in database from any disclosure. An auditing mechanism is embedded into the system for monitoring all user transactions in database. This study modifies different categories presented in Kumar and Al Hasani (2016) and Lu and Miklau (2009) for increasing the security of passwords stored in database. The contribution of this study is as follows:

- Building a user verification algorithm using CAPTCHA test for recognizing whether the login operation is executed by human being or computer generation.
- Building a password-based Algorithm using Salt Hashing for encrypting passwords file stored in database from adversary users.
- Enforce a password complexity requirement by merging the secret key of the user and the auto generated activation key into one distinct key.
- Building an auditing mechanism for monitoring user transactions in database.
- Using the auditing mechanism to register user logs in database and verify normal users from malicious user activities.

LITERATURE REVIEW

For developing a strong security system, database security applications depend on three main dimensions: Confidentiality, integrity and availability. Different research papers present hybrid mechanisms for maintaining confidentiality and integrity of data. These mechanisms are used as pre-security procedures for securing database. Auditing database is considered the post-security procedure for controlling and monitoring database after performing transactions.

Database triggers for auditing database are presented in Fabbri *et al*. (2013). In this research, the auditing system is a key component of a database security infrastructure. Database auditing requires monitoring the access to sensitive data. A database auditing logging component is presented in Liu and Huang (2009). The first step in database auditing is the logging procedure. It is used to determine if security parameters and other policies are being violated and if so, provide evidence of violation, discover attacks to database and help recover a database in case of any

Corresponding Author: Ayman Mohamed Mostafa, Faculty of Computers and Informatics, Zagazig University, 44519, Egypyt

damage. Retention restriction of a database auditing framework is presented in Lu and Miklau (2009). A historical model is implemented in this framework to perform an accurate audit to check the old and new changes in database under retention restrictions rules.

Securing dynamic auditing protocol is presented in Yang and Jia (2012). This research focuses in designing an auditing framework for cloud storage system to implement privacy preserving and efficient storage auditing protocol to ensure data privacy. As presented in Huang and Liu (2009), a logging schema for database auditing is implemented for monitoring and logging database activities which increase the security of sensitive data in network.

A database auditing framework is presented in Wu *et al*. (2014). An auditing framework is designed and implemented to reduce security risks. System design of unified auditing on complex network is presented in Liu *et al*. (2012). In this research, auditing is performed to ascertain the validity and reliability of information and providing an assessment of a systems internal control. The authors of Dai *et al*. (2012) present a framework to eliminate backdoors from response computable authentication. An authentication module is used for addressing the problems of backdoors using a two party authentication model (RCA). Database forensic analysis is presented in Khanuja and Adane (2012) for validating, analyzing, collecting, detecting avoiding and reporting any strange behavior in a database.

Securing outsourced database architecture is presented in Althneibat *et al*. (2010). In this research, database architecture is built for reducing computation and communication overhead by partially encrypting data and information and increase data confidentiality through using deferent encryption techniques. Authors of Chahar *et al*. (2015) present an approach for biometric authentication using a Leap Password. The leap password consists of a string of successive gestures performed by the user during which physiological as well as behavioral information is captured. The authors of Howe *et al*. (2012) study the psychology of security for home computer users.

An analysis of field data on web security vulnerabilities is presented in Fonseca *et al*. (2014). In this research, some faults appeared in source code can help the attacker in performing unauthorized access, gain access to privileged database account. An Input password method for handicapped people is presented in Ito *et al*. (2016). The main idea of this research is to input a password system with only three keys and show an analysis of the security evaluation and results of short user test.

SYSTEM ARCHITECTURE

The main architecture of the security system is based on two stages. The first stage is used for system user registration and the second stage is used for users logging in into the system.

System registration stage: In system registration stage, each user creates a username, password, email and registration code. The registration code is a 3 digit distinct key that is created by the database administrator and is given to each user. This key is kept private by the user for later login operation. After the user submits all required information, a 5 digit activation code is sent to the user' email for later login. An 8-digit authentication code is created by concatenating the 3-digit private key created by the DBA and the 5-digit activation code that was sent to the user' email. The activation code is created uniquely using salt hashing encryption technique. Each time the authorized user enters the system, a new activation code is auto-generated and is sent to his email.

$$\text{8 digit Authentication Code (AC)} = \text{Private Key} \,||\, \text{Activation Code} \quad (1)$$

Access control stage: In access control stage, access control parameters and captcha are submitted by each user. Captcha is a challenge-response test used to ensure whether the response is generated by a human being or computer generation. The process involves a computer asking a user to complete a simple test. This test is designed to be easy for a computer to generate but difficult for a computer to solve. If a correct solution is received, then it is assumed that the data is entered by a human. This is presented in Algorithm 1.

Algorithm 1: Intrusion recognition

1. If Captcha is valid Then
2. {
3. Retrieve all usernames and passwords from Security. Users
4. Put usernames and passwords into system cache
5. If entered username and password is authentic Then
6. {
7. Go to 8-digit Authentication Code Session
8. }
9. Else
10. {
11. Raise Error Alarm
12. }
13. }

The security system requires a username and password from any user entering the system to classify and filter out intrusion attempts. The username and password for each authorized user must be defined by the Database Administrator (DBA) during user creation and must be known to the legitimate users only. As presented in Algorithm 1, if the username and password

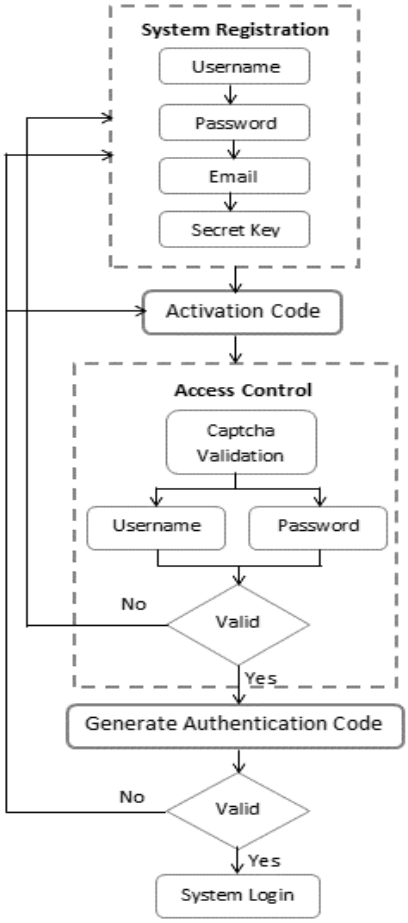

Fig. 1: System registration and access control procedure

are authentic, the authorized user must enter 8-digit authentication code as presented in formula 1. The overall system architecture is presented in Fig. 1.

MATERIALS AND METHODS

This research was conducted on 2016 and the beginning of 2017 for Kuwait ministry of education. The developed password-based application is used for storing student records with their grades. The officials of the educational process are responsible for tracking student grades and their confidential information. These officials are logging on to the system using their authentication ID.

The front line of defense against intruders is the password-based system. All multiuser systems require not only a name or identifier (ID) of the user but also a password. The password serves as an authentication to the ID of the individual logging on to the system.

In order to prevent any threats to password-based systems, the encryption process is executed using MD5 hashing and salt hashing algorithms. The main objective of the encryption procedure is encrypting not only the authorized user password but also protecting the password stored in the database from any disclosure. Algorithm 2 explains the MD5 hashing and salt hashing algorithm for the security system.

Algorithm 2: MD5 and Salt Hashing

```
Create MD5_Hash Function
 {
Input Hash = MD5. Compute_Hash (Input)
Return string (Hash)
 }
Create MD5_Hash_Salt Function
 {
Input Hash = MD5. Compute_Hash (Input, Salt)
Return string (Hash)
}
Create SHA1_Hash Function
{
Input SHA1 Service
 Input Hash = SHA1. Compute_Hash (Input)
Return string (Hash)
 }
Create SHA1_Hash_Salt Function
{
Input SHA1 Service
 Input Hash= SHA1.Compute_Hash (Input, Salt)
Return string (Hash)
 }
Generate Random_Salt_Password
{
 Input integer I
Execute MD5_Hash Function
Execute MD5_Hash_Salt Function
Execute SHA1_Hash Function
Execute SHA1_Hash_Salt
ReturnEncrypted_Password
}
```

By performing a salt hashing encryption, all users' passwords are kept private in the database. If a malicious user tries to retrieve the password system data, a different password will be generated which is completely different from the original one. So, salt hashing is a one-way function when it is hashed, it can't be retrieved to obtain original input.

After performing the cryptographic Algorithm, the encrypted password is kept private in the database. An activation code is generated using Triple DES encryption algorithm and is sent to the authorized user email. As presented in Fig. 2, an authentication procedure is generated by merging the activation code with the private password. For logging in into the system, the authorized user must write the correct concatenation.

As presented in Fig. 2, the generated authentication code is unique and random key. It is created based on the concept of one-time pad mechanism. The authentication code is used for a single sign in into the

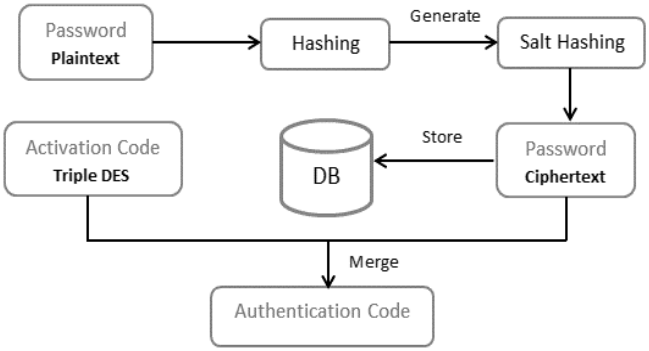

Fig. 2: Authentication procedure

system and then is discarded. Every time the user enters into the system, a new authentication code is generated making the whole application unbreakable.

RESULTS AND DISCUSSION

As presented in Algorithm 1 and 2, different access control policies are added to increase the robustness of the password-based system. The results and mechanism of the password-based system are compared with the results presented in Ito *et al*. (2016). Different contradictions and added data are presented to enhance the efficiency of the password-based file from adversary attacks.

Enforce password history: As presented in Ito *et al*. (2016), a number of 24 unique passwords are used before using a password history. Hackers can frequently bypass controls and gain access to the file. The attacker obtains the system password file and compares the password hashes against hashes of commonly used passwords. If a match is found, the attacker can gain access by that ID/password combination. This is considered a contradiction point.

In the developed password-based system, every attempt to enter the system is based on a unique password without the need to use password history. In addition, if a malicious user tries to retrieve the password system data, a different password will be generated which is completely different from the original one.

Maximum password age: As presented in Ito *et al*. (2016), the expiration date for each password is set to 90 days. The adversary may perform a brute-force attack until the password is breached. If the user' password is breached, the intruder can retrieve and perform different transactions for a long time. This is considered a security violation and contradiction point.

In the developed password system, each sign in into the system requires a unique password for every user which is completely different from the previous password for the same user. All password logs are encrypted to prevent malicious users from accessing the password history and breach the encryption pattern for each user.

Maximum password length: The password length presented in Ito *et al*. (2016) is based on 8 characters length.

In the developed password-based system, the same password length is set but the key is portioned from two sources. A 3 digit distinct key that is created by the database administrator is merged with a 5 digit activation code created randomly by Algorithm 2. This makes the password a unique one in each sign in. This can increase the complexity requirement for the developed application.

By applying the password-based system, an auditing mechanism is embedded into the system. The principle key for implementing efficient auditing mechanisms is to understand the major auditing requirements and the main categories for reporting auditing operations.

The main idea is to present efficient detection and prevention strategies to obtain efficient, flexible and solid database security system. Different database auditing mechanisms are developed to report all auditing operations inside database whether the transactions are executed by normal users or malicious users.

If normal users perform different database transactions, the auditing mechanisms are used to create a report to list all user operations inside the database. If malicious users succeed in breaching the database security prevention and detection algorithms, the auditing mechanisms are used to report all transactions. The Database Administrators (DBAs) can perform error containment operation by rolling back all malicious transactions in the database.

As presented in Table 1, the auditing mechanisms include: Auditing ID, action type whether it is DDL, DML, or login operations, the audit action name whether it is a query on database or a modification of data, the kind of action whether it is positive action or negative action, the source table name, the page URL of actions, the username and the timestamp for the transaction. The developed algorithms and application

Table 1: Auditing mechanism

Audit id	Audit action type	Audit action name	Spname
1519	System DML	Update in database	User_admin_loginfo
1520	System DML	Update in database	User_admin_loginfo
1521	System DML	Sign in	Auditlog-ddl
1522	Login	Check information of sign in	User_pk-logininfo
1523	Login	Sign in	User_pk-logininfo
1524	Login	Check information of sign in	User_pk-logininfo
1525	Login	Sign in	User_admin_loginfo
1526	Admin Login	Check information of sign in	User_admin_loginfo
1527	Admin Login	Check entry information	User_admin_loginfo
1528	Admin Login	Sign in first page	User_admin_loginfo
1529	Admin Login	Check entry information	User_admin_loginfo
1530	Admin Login	Sign in first page	User_admin_loginfo
Audit id	Table name	Sql type	Page url
1519	Lookup_Audit Action Type	update	http://localhost:1310/pl/public/logain.aspx
1520	Lookup_Audit Action Type	update	http://localhost:1310/pl/blic/logain.aspx
1521	test	Drop_table	http://localhost:1310/pl/public/logain.aspx
1522	V_pk_login	select	http://localhost:1310/pl/public/logain.aspx
1523	V_pk_login	select	http://localhost:1310/pl/Admin/Adminlogin.aspx
1524	V_pk_login	select	http://localhost:1310/pl/Admin/Adminlogin.aspx
1525	V_pk_login	select	http://localhost:1310/pl/Admin/Adminlogin.aspx
1526	V_admin_login	select	http://localhost:1310/pl/Admin/Adminlogin.aspx
1527	V_admin_login	select	http://localhost:1310/pl/Admin/Adminlogin.aspx
1528	V_admin_login	select	http://localhost:1310/pl/Admin/Adminlogin.aspx
1529	V_admin_login	select	http://localhost:1310/pl/Admin/Adminlogin.aspx
1530	V_admin_login	Select	http://localhost:1310/pl/Admin/Adminlogin.aspx
Audit id	By user name	Audit date	Audit time
1519	System user	11/06/2016	22:47:33
1520	System user	11/06/2016	22:47:41
1521	System user	14/06/2016	22:52:53
1522	Anonyms	21/06/2016	23:37:40
1523	Anonyms	12/07/2016	02:06:40
1524	Anonyms	25/08/2016	02:19:16
1525	Anonyms	27/09/2016	03:18:27
1526	Anonyms	06/10/2016	02:20:57
1527	Anonyms	12/10/2016	09:24:15
1528	Anonyms	15/11/2016	11:37:14
1529	Anonyms	20/12/2016	12:59:24
1530	Anonyms	05/01/2017	03:01:19
Audit id	Auditing source name		
1519	Via database dml		
1520	Via database dml		
1521	Via database dll		
1522	Via application		
1523	Via application		
1524	Via application		
1525	Via application		
1526	Via application		
1527	Via application		
1528	Via application		
1529	Via application		
1530	Via application		

are implemented using Microsoft Visual Studio 2010 for building the web application with SQL Server 2008 for the database system.

CONCLUSION

The main goal of this study is to build password-based algorithms for encrypting password system data in a database. All authorized users are logging into the system using a random and unique password. This password is stored in the database password file for a single sign in. This information may be breached by an adversary to obtain the mechanism for generating passwords. For preventing the password file from disclosure, a hash encryption algorithm is implemented for concealing passwords making the reverse operation invalid. Any attempt to obtain the system password file will generate a new password which is not recorded and not completely different from the original one. An auditing mechanism for tracking all database transactions is also presented for monitoring all users' actions in the database.

ACKNOWLEDGMENT

This study is partially supported by the ministry of higher education in Egypt and we are grateful to them for their help and support. The authors are grateful to

the anonymous referee for a careful checking of the details and for helpful comments that improved this study.

CONFLICT OF INTEREST

The development of the manuscript is financially supported by Faculty of Computers and Informatics - Zagazig University as one of the manuscripts for post-doctor promotion.

REFERENCES

Althneibat, A.M.A., B.E.M. Hasan, A.E.F.A. Hegazy and N. Hamza, 2010. Secure outsourced database architecture. Int. J. Comput. Sci. Netw. Secur., 10(5): 246-255.

Chahar, A., S. Yadav, I. Nigam, R. Singh and M. Vatsa, 2015. A leap password based verification system. Proceeding of the IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS), pp: 1-6.

Dai, S., T. Wei, C. Zhang, T. Wang, Y. Ding, Z. Liang and W. Zou, 2012. A framework to eliminate backdoors from response-computable authentication. Proceeding of the IEEE Symposium on Security and Privacy Conference, pp: 3-17.

Fabbri, D., R. Ramamurthy and R. Kaushik, 2013. SELECT triggers for data auditing. Proceeding of the IEEE 29th International Conference on Data Engineering (ICDE), pp: 1141-1152.

Fonseca, J., N. Seixas, M. Vieira and H. Madeira, 2014. Analysis of field data on web security vulnerabilities. IEEE T. Depend. Secure., 11(2): 89-100.

Howe, A.E., I. Ray, M. Roberts, M. Urbanska and Z. Byrne, 2012. The psychology of security for the home computer user. Proceeding of the IEEE Symposium on Security and Privacy (SP), pp: 209-223.

Huang, Q. and L. Liu, 2009. A logging scheme for database audit. Proceeding of the 2nd International Workshop on Computer Science and Engineering (WCSE'09), pp: 390-393.

Ito, A., Y. Kumazawa and M. Okamoto, 2016. Input password method for handicapped people. Proceeding of the IEEE SAI Computing Conference (SAI), pp: 1306-1308.

Khanuja, H.K. and D. Adane, 2012. A framework for database forensic analysis. Comput. Sci. Eng. Int. J., 2(3): 27-41.

Kumar, B. and M.H.S. Al Hasani, 2016. Database security - risks and control methods. Proceeding of the IEEE International Conference on Computer Communication and the Internet (ICCCI), pp: 334-340.

Liu, L. and Q. Huang, 2009. A framework for database auditing. Proceeding of the IEEE 4th International Conference on Computer Science and Convergence Information Technology (ICCIT'09), pp: 982-986.

Liu, L., C. Li and X. Li, 2012. System design of unified auditing and monitoring based on complex network. Proceeding of the IEEE 2nd International Conference on Intelligent System Design and Engineering Application (ISDEA), pp: 1144-1147.

Lu, W. and G. Miklau, 2009. Auditing a database under retention restrictions. Proceeding of the IEEE Conference on Computer Science and Convergence Information Technology, pp: 1-12.

Wu, K., L. Hua, X. Wang and X. Ding, 2014. The design and implementation of database audit system framework. Proceeding of the 5th IEEE International Conference on Software Engineering and Service Sciences (ICSESS), pp: 553-556.

Yang, K. and X. Jia, 2012. An efficient and secure dynamic auditing protocol for data storage in cloud computing. IEEE T. Parall. Distr., 24(9): 1717-1726.