Detecting Insider Attacks in SQL DB Using Blockchain

Naga Srividya Varanasi
nvnxp@mail.umkc.edu
Fall 2019
December 13th,2019

Nikita Goyal
ngoyal@mail.umkc.edu
Fall 2019
December 13th,2019

Sreevalli Tata
st8d7@mail.umkc.edu
Fall 2019
December 13th,2019

Sai Srinivas Vidiyala
Sai.srinivas.vidiyala@mail.umkc.edu
Fall 2019
December 13th,2019

Abstract **— Applications that are built on a centralized database are often at risk for insider attacks. Any user with administrative privileges can tamper the database easily and it is difficult to predict. We implemented a solution in which detecting insider attacks using the SQL database in the multichain environment, which is a blockchain technology that follows tamper resistance property. We have developed a student management system and performed create, update, select and delete operations.**

Keywords— Integrity, Security, Blockchain, Multichain, Tamper resistance, Centralized database.

I. INTRODUCTION

The Database system is used in almost all web-based and mobile-based applications to perform day to day activities. Data is generating at a faster pace and maintaining the security of the data from unwanted modification is of utmost importance. It is difficult to secure database servers and define security control policies maintaining the read/write access for various users of the system. The application server takes care that these policies are not violated. However, detecting insider attacks is still a strenuous job.

A. Detailed Introduction

Our student management system has been developed and implemented so that the insider attack is detected. We have implemented the student database and created a signup/login page for both students and instructors. We have used REST API calls which is JSON RPEC to interact with the multichain server using HTTP request/response. If somebody tries to exploit the student data, and also tries to manipulate the logs, our system will detect with ease and will generate a pop-up message. The technology stack we have used to implement this system includes Python Tkinter based GUI and for backend, PHP is used as it is best suited with the MYSQL database.

B. Current State

We have implemented the 'insider threat' using blockchain and multichain by including the MYSQL database. For every query issued to the database, a check is performed to see if the database is consistent with what is present in the Multichain stream. This stream is used to track the latest transaction. We have also implemented the security check as well as a user-friendly interface.

C. Structure of paper

In further topics, we are going to discuss more in detail about abbreviations in section[II], already existing work in section[III], detail description of our problem in section[IV], our approach towards the problem in section[V], block diagram and working explained in section[VI], our in-depth scheme and algorithm used in implementing our thought in section[VII] and followed by expected behavior in section[VIII], future work in section[IX], conclusion in section[X] and lastly references[XI].

II. ABBREVIATIONS

A. Blockchain

The Blockchain is a new and rapidly developing technology. The blockchain is a reliable way of storing data about transactions, contracts, basically, everything that needs to be written down and verified. Members of the network are anonymous persons, called nodes. A block is an array of data, containing information about the transactions that entered the system after the creation of the previous block. Every block is connected chronologically, each new data block is attached to the previous one employing complex mathematical and cryptographic algorithms, which makes it possible to consolidate the blocks. A base and a header represent each block. The base of the block is a particular order of information records. The header is the key to the previous block sequence and a hash function. When information needs to be updated in a particular a new block will be added with information such as date and time of updating. Thus, the blockchain concept provides integrity, safety, and transparency to the data.

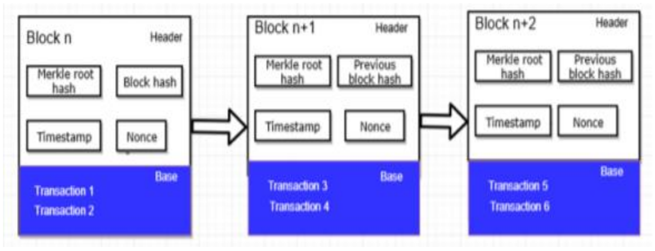

Fig: Internal structure of blockchain

B. Multichain

Multichain technology is a platform which is rapidly growing to help users to deal with financial or organizational transactions by creating a private blockchain. Multichain assists firms to build applications with high speed. It also allows to deploy their own applications. It has a fine-grained permission property, and it optionally controls who can send and accept transactions, streams, and blocks. Multichain provides us with a simple application program interface and command-line interface, which helps to preserve and set up the chain. Whenever different nodes in blockchain connect, multichain takes place.

Each node has an address that has a set of permissions. When each node sends a message to the other users to have peer to peer connection, it should satisfy all the rules, if not the connection will be aborted.

- Hundred/thousands of instructions executed per minute.
- Multiple application program interface functions.
- It can simultaneously run multiple nodes by using the same server.
- Node permission is flexible.

C. Centralized Database

In this database, there will be central control of all applications, data. Since information is hidden inside the server it saves the processing costs. However, the safety of data may be compromised as there is only one single unit of control. This type of database can be easily accessed or modified or changed easily and can become a victim of such insider attacks.

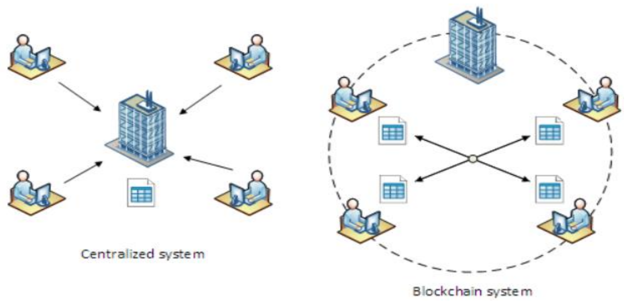

Fig: Centralized system and blockchain system.

III. EXISTING WORK

Verity is a framework that is more advanced in detecting insider threats or attacks in centralized or distributed systems. It not only handles the CRUD operations but also handles most of the SQL functional queries like joins, aggregated queries, and nested queries. Because it uses Hyperledger fabric technology of blockchain it takes only 0.8 seconds – 2.5 seconds to query per tuple. Durability does not apply to verity as it does not store any data with itself.

IV. DETAIL DESCRIPTION

Insider attack has become a very crucial issue in protecting the data integrity in any system. These attacks can be either passive or active. When someone unethically tries to access a database is called passive attack while the attacks that deal with tampering of the data and the logs are called active attacks.

We have used Merkle trees which is most commonly used in blockchain technology. Because of its widespread usage, it is most efficient and secure for verification and authentication. These hash codes are used as a replacement for pointers and thus make the hashing process easy. Repeated hash pairs are generated for different nodes till there is only one hash node left i.e., root of the tree.

This Merkle root is responsible for adding all the data and summarizing it into a block header. This maintains the integrity of the data in the blockchain technology.

We have used PHP MYSQL combination as it has several advantages. Because of PHP's dynamic nature of web application, it doesn't store any information by itself. MySQL, on the other hand, automates the most frequent tasks for retrieving and storing unique user information. Together they deliver faster performance, greater scalability, and high reliability.

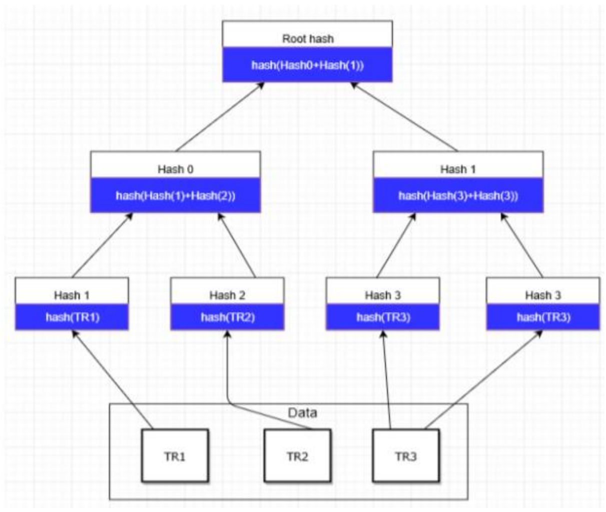

Fig: Implementation of Merkle trees.

We have created different nodes in a multichain server and on the top of each node we have created a stream for each course. From our PHP code, we are publishing our stream in the form of a hash which is a 16-bit binary randomly generated key and is broadcasting to the server.

V. APPROACH

In our approach, we have created a stream for each course that the instructor teaches. A stream consists of the ledger where it stores all information regarding a particular transaction. This ledger has several columns like timestamp, retrieval key, and public access key. This stream is generated using a multichain server for each of the courses and is sent to different multichain nodes. This multichain server takes care of authentication and verification of credentials and publishing of the student data. The multichain node takes care of transaction confirmation. Here, we are not migrating the entire database to the multichain server rather we are just passing a stream of the data to the server. This stream consists of metadata which is used when someone tries to access the entries of the database. Streams in a multimode chain can implement the following:

1. Document storage
2. Time series database using timestamp which is used to the ordering of the transactions.
3. All entries are classified according to user convenience.

We have to make a few modifications to implement our protocol.

- ➢ It takes values of all columns and rows and forms a tuple.
- ➢ A pointer is present in the transaction field (k+ 1).

- The transaction is initially signed by the user and then using a public-key digital signature from already existing PKI key and verifies.
- If the signature is matched, the hash value of tuple will be generated.

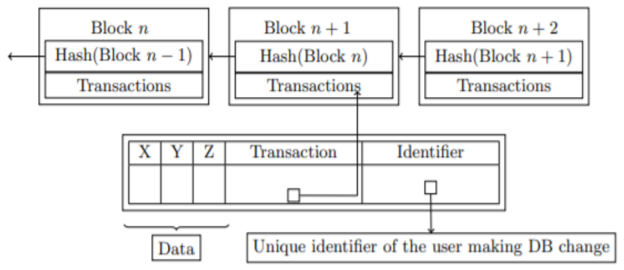

Fig: Internal database schema modification of blockchain.

VI. BLOCK DIAGRAM

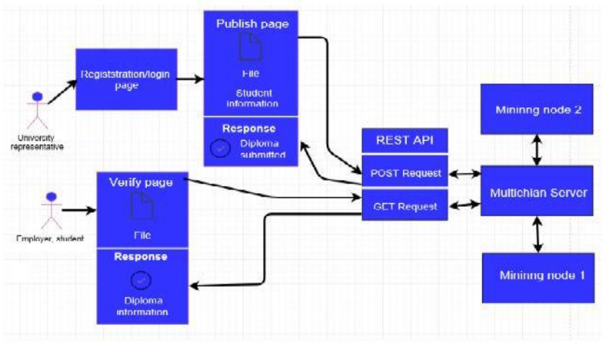

Fig: Application workflow

When the new university representative accesses the application, the registration request must be submitted for university authorization. After carrying out the necessary checks, the login and password are provided for the use of the system. Every university connected to the system fills in a form with the name, student ID number, diploma ID and upload a scanned copy of the degree certificate. The document is timestamped according to the order received, the information is hashed and converted to hexadecimal form and published to the blockchain. As soon as the transaction is confirmed by the system, the diploma information is visible in the application. Any further forgery attempt causes a discrepancy between the hash functions of the original document and the document with unauthorized changes.

A student, graduating from an educational institution, receives a copy of the original document and the information stored in the system. It includes a hash function that acts as a digital signature and a sequence number in the system, the transaction number, and the time when the document was loaded into the system. If it is necessary to confirm the passage of the studies at the university, this information can be provided to the employer for further verification.

There are two ways to verify the authenticity of the diploma:
• Database search using a hash function
• Upload the document to the system

The First method involves entering a hash function in the search field. The system looks for matches in the database and outputs the results in the form of brief information and a link to the extended version with the complete report. The Second method involves document upload into the system and forming a hash function. Next, the system compares the hash functions of existing documents in the blockchain and the hash function of the uploaded file. Whenever a match is found, the user will be provided with a full report.

VII. SCHEME

A. *SignUp and Login*

We have implemented our application using Python Tkinter based GUI. Instructors can sign up and then log in to the application. When they sign up, an RSA key pair is generated. The Private Key is encrypted using the passphrase that the instructor enters and is stored on the person's system. The Public Key is sent to the server where it is broadcast as a transaction on a public-key stream.

B. *Insert Query*

An instructor can enter grades of his students in the courses that he teaches. The courses that a particular instructor teaches is not stored on the database but present as a transaction on an instructor stream.

An instructor enters a student's ID, course, and grade with each string separated by a newline. On submitting a bunch of grades, each line of grade is digitally signed using the instructor's Private Key and sent to the server. The server then retrieves the public key of the instructor from the public-key stream and verifies the signature.

For each grade about to be entered, the database is retrieved* (ordered by txID) and each tuple (txID + uid + course + grade + identifier) # is concatenated and a hash is obtained. To this hash, the present grade is concatenated (uid + course + grade + identifier) and again hashed. This final hash is then published to the stream which gives us a transaction for that grade.

$s = "INSERT INTO scores VALUES (":uname, :course, :score, :txid, :idfr)";

$stmt = $conn ->prepare($s);

C. *Update Query*

The instructors also have an option to update a student's grade. Unlike Inserts, updates happen one at a time. When an update is issued, the whole database is retrieved and the old grade tuple is removed from it. It is now hashed, the new data is concatenated and again hashed (similar to the insert query above) and this is sent as a transaction to the stream. The old grade is then updated with the new grade and the new transactionID.

$s =   "UPDATE scores SET score=:scr, txid =: tid WHERE uid =: uname AND course =: cors";

$stmt = $conn -> prepare($s);

D. Select Query

When we run a select query, the whole database is retrieved. The new data is hashed is sent as a transaction to stream.

```
$q = “SELECT txid, uid, course, score, identifier from grades WHERE uid = :uid AND course = :cors”;

$stmt = $conn -> prepare($q);
```

VIII. EXPECTED BEHAVIOR

Whenever a student views his grades, the student has to log in initially to his student account using his credentials. The credentials that the student gives should match with the credentials that are already present in the database. After login-in successfully, the server fetches information about the grades from the central database. The application retrieves the public key using an identifier and the signature of the transaction is then verified. In case of any failure in the system, it will be notified in the application itself and a notification will be seen at the top.

IX. FUTURE WORK

1. Storage of files in the blockchain. Now, files are not loaded to the blockchain and only the hash of a document is stored. It is possible to implement this using the Multichain platform, but it will slow down the work of network nodes which are confirming the transactions and creating new blocks. To perform this task, it is necessary to select the optimal document size and document storage algorithms in the blockchain.

2. Improvement of safety. Basic security matters in this application version do not guarantee the absence of unauthorized publications in the blockchain by hackers getting access to a university account. Adding a two-factor authorization (2FA) mechanism and electronic identification can significantly improve this parameter.

3. Use of public blockchains. In the future, the use of public blockchain projects such as Ethereum can be considered. In contrast to private blockchains, in theory, the degree certificate can be stored forever in the public network. However, this significantly increases the cost of the system since in public networks each transaction costs a certain amount.

X. CONCLUSION

We have developed a system using blockchain technology which is one of the most secure platforms in the security area and has been very strong to break it. As compared to other frameworks, the execution time of CRUD operations is more and the overhead storage is low because we are not fetching the entire database rather then we are only considering a stream of the query that has to be processed.

XI. REFERENCES

[1] Blockchain Hub. (n.d.). Blockchain Glossary for Beginners. Retrieved from https://blockchainhub.net/blockchain-glossary/

[2] Bear, J., & Ezell, A. (2012). Degree Mills: The Billion-Dollar Industry That Has Sold Over a Million Fake Diplomas. New York: Prometheus Books.

[3] Satoshi, N. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Retrieved from https://bitcoin.org/bitcoin.pdf

[4] Swan, M. (2015). Blockchain: Blueprint for a New Economy (1 edition). Sebastopol: O'Reilly Media.

[5] Imran, B. (2017). Mastering Blockchain: Deeper insights into decentralization, cryptography, Bitcoin, and popular Blockchain frameworks. Packet Publishing.

[6] Dzone. (2017). An Introduction to Blockchain. Retrieved May 16, 2018, from https://dzone.com/articles/how-blockchain-technology-works-andgives-maximum

[7] Invest in Blockchain. (2017). How Does Cryptography Protect the Blockchain? Retrieved from https://www.investinblockchain.com/howdoes-cryptography-protect-blockchain/

[8]Shaan, R. (15 December 2017). Merkle Trees. Retrieved from https://hackernoon.com/merkle-trees-181cb4bc30b4

[9]Lee, D. K., & Deng, R. H. (August 18, 2017). Handbook of Blockchain, Digital Finance, and Inclusion, Volume 1: Cryptocurrency, FinTech, InsurTech, and Regulation. London: Academic Press; 1st edition.

[10] Lamport, L., Pease, M., & Shortak, R. (n.d.). The Byzantine Generals Problem [PDF]. SRI International. Retrieved from http://people.eecs.berkeley.edu/~luca/cs174/byzantine.pdf

[11]Dwork, C.,& Naor, M. (n.d.). Pricing via processing or combatting junk mail [PDF].

[12] Castro, M., & B., Liskov. (n.d.). Practical Byzantine Fault Tolerance [PDF]. Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge. Retrieved from http://pmg.csail.mit.edu/papers/osdi99.pdf

[13] Byterin, V. (2015, August 7). On Public and Private Blockchains [Web log post]. Retrieved May 8, 2018, from https://blog.ethereum.org/2015/08/07/on-public-and-privateblockchains/

[14] Williams, S. (2017, December 11). 5 Big Advantages of Blockchain, and 1 Reason to Be Very Worried [Web log post]. Retrieved May 8, 2018, from https://www.fool.com/investing/2017/12/11/5-big-advantages-ofblockchain-and-1-reason-to-be.aspx

[15] Rosic, A. (2017). 17 Blockchain Applications That Are Transforming Society [Web log post]. Retrieved May 8, 2018, from https://blockgeeks.com/guides/blockchain-applications/

[16] Greenspan, G. (2015). Multichain Private Blockchain — White Paper [PDF]. Retrieved from https://www.multichain.com/download/MultiChainWhite-Paper