Using blockchain technology to improve security against SQL injections

UDK 004.056

Iryna Zamrii[1], Ivan Shakhmatov[2]

State University of Information and Communication Technologies,
[1]irinafraktal@gmail.com, [2]ivan.shakhmatov@gmail.com

The threat of SQL injections remains an actual problem in the field of web development, as they reveal system vulnerabilities and allow attackers to gain unauthorized access to data [1-2]. This type of attack consists of injecting malicious SQL code through the user interface, which can lead to unauthorized access, leakage or loss of information. Attackers are actively looking for weak points in protection in order to gain access to sensitive information, delete critical data or even gain control over servers.

The BlockchainSQLSecure model is a SQL security solution that combines blockchain technology and a Bloom filter to detect and prevent SQL injections. What's unique is that the model is not limited to tracking standard SQL injection scenarios, but instead uses blockchain to securely store information about previous queries. This allows you to create a log that is resistant to changes and can be checked at any time to ensure data security.

The developed model does not require large computing resources for its work and can be easily integrated into existing systems. The Bloom filter, in turn, provides high processing speed and efficiency, which makes it ideal for working with large volumes of data. In the BlockchainSQLSecure model, a key aspect is the identification of similarities between SQL queries to effectively detect and prevent SQL injection attempts. To achieve this goal, the Jacquard distance formula is used:

$$d(X, Y) = \frac{\sum_{i=1}^{n} X_i Y_i}{\sum_{i=1}^{n} X_i^2 + \sum_{i=1}^{n} Y_i^2 - \sum_{i=1}^{n} X_i Y_i} \quad (1)$$

where X_i, Y_i are the coordinates of vectors X and Y respectively.

The identification between two SQL queries is by formula (1), where a value leading to 1 indicates similarity and a value leading to 0 indicates relativity. This distance is an important element of the algorithm for removing duplicate data based on the Bloom filter and increases the efficiency and accuracy of the model. The metric allows you to quickly determine whether a new SQL query is a variant of an existing SQL query that contains a potential injection attempt. Using the Jacquard distance in combination with the Bloom filter increases the efficiency and accuracy of the system in detecting and preventing SQL injections.

The proposed UML diagram (Fig. 1) shows the class structure of the innovative system, which combines the functions of processing SQL queries, managing blockchain transactions, monitoring user actions, and using the Bloom filter to optimize queries. The developed UML diagram contains five main classes: User, SQLQuery, BlockchainManager, AuditLogger and BloomFilter, each of which plays an important role in the functioning of the system and establishment of relationships and dependencies between classes.

The development of the system in the Python programming language made it possible to

adopt the principles of clean coding, ensuring quality, ease of understanding and the possibility of further expansion of the project. The system architecture was developed in the form of modules and classes, which ensured a clear structure and division of responsibilities.

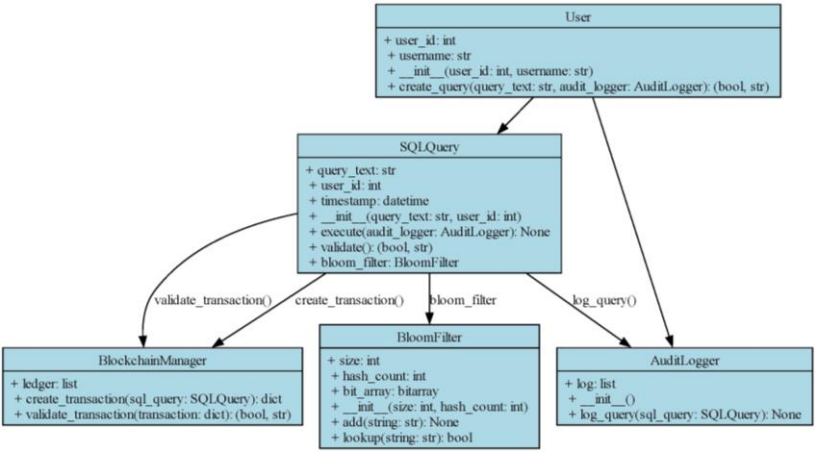

Fig. 1 UML class diagram for database management system and blockchain transactions

The created system demonstrates a high level of structure, security and efficiency. Using the UML diagram allowed us to carefully plan the architecture of the system, identifying the key components and their interactions. Each class plays a unique role that contributes to the creation of a complete and reliable system.

1. Tanrıverdi M., Tekerek A. Implementation of Blockchain Based Distributed Web Attack Detection Application. Feminist Press at CUNY. 2021. 102 p.
2. Alghawazi M., Alghazzawi D., Alarifi S., Detection of SQL Injection Attack Using Machine Learning Techniques: A Systematic Literature Review. *Journal Cybersecurity and Privacy*, 2022, 2(4), pp. 764-777.