International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 11 Issue: 03 | Mar 2024 www.irjet.net p-ISSN: 2395-0072

Garble Encryption System for Data encryption in website database

Meenakshi A[1], Vinoth G[2], Jayanthan R[3], Shiyam Sundar GS[4]

[1] Head of the Department, Department of Computer Science and Engineering,
[2,3,4] Student, Department of Artificial Intelligence and Data Science,
[1,2,3,4] Kamaraj College of Engineering and Technology, Virudhunagar, Tamil Nadu, India

---

***Abstract*— In today's digital age, the security of sensitive data stored in databases is of paramount importance. One of the most prevalent threats to database security is SQL injection cyber attacks, which can potentially expose confidential information and compromise the integrity of systems. This project delves into the imperative strategy of encrypting database data as a robust defence against SQL injection attacks. The objective is to provide a comprehensive understanding of the role of encryption in mitigating this critical security risk. The project starts out by outlining the principles behind SQL injection attacks and how malevolent actors might use web application vulnerabilities to insert malicious SQL code into database requests. These attacks can lead to unauthorized access, data theft, and data manipulation, making them a significant concern for organizations and individuals alike. Subsequently, the project explores encryption as a countermeasure to SQL injection attacks. It delves into the fundamental concepts of encryption, emphasizing its effectiveness in safeguarding data at rest. Furthermore, this project examines the performance and trade-offs associated with encrypting database data. It presents real-world case studies and benchmarks, which demonstrate that while encryption adds computational overhead, the security it provides is well worth the cost. In summary, this project underscores the significance of encrypting database data as a formidable defence against SQL injection cyber attacks. It provides insights into the mechanisms of SQL injection attacks, the principles of encryption, key management best practices, and performance considerations. By implementing encryption alongside other security measures, organizations can fortify their databases and safeguard their sensitive data from the ever present threat of cyber attacks**

Keywords : Encryption, Secure database, User friendly, Cyber theft

I.INTRODUCTION

Garble Security emerges as an indispensable asset in the realm of database management, offering a locally deployable web application meticulously integrated with the GlassFish server. This seamless integration not only guarantees compatibility but also elevates reliability, providing users with a stable and resilient platform to execute database operations with utmost efficiency and fortified security measures. The amalgamation of Garble Security with GlassFish server forms a symbiotic relationship, ensuring seamless communication and optimized performance within the database environment.

Upon the initiation of Garble Security, users are greeted with a streamlined setup process designed to facilitate swift and hassle-free establishment of a secure connection to their designated database. This process begins with the input of a specified URL, username, and password, enabling the application to promptly authenticate and validate successful connectivity. This initial interaction serves as a testament to Garble Security's commitment to user satisfaction and confidence, instilling users with a sense of assurance from the outset of their engagement with the platform.

The integration of Garble Security with the GlassFish server not only enhances compatibility but also augments the overall reliability of the database management ecosystem. The robust infrastructure provided by GlassFish server serves as a solid foundation for Garble Security, ensuring seamless operation and optimal performance even in the face of demanding tasks and fluctuating workloads. Users can rely on the stability and resilience of the platform to support their database management endeavors with unwavering consistency and efficiency.

Moreover, Garble Security's commitment to robust security measures reinforces its status as an indispensable asset in safeguarding sensitive data and mitigating potential security risks. The application employs advanced encryption techniques to secure data transmission between the application and the database server, effectively shielding sensitive information from unauthorized access and potential breaches. By encrypting data both at rest and in transit, Garble Security provides users with peace of mind, knowing that their valuable data assets are protected with the highest level of security standards.

In addition to its security features, Garble Security excels in streamlining database management operations, offering a comprehensive suite of tools and functionalities tailored to enhance productivity and optimize workflow efficiency.

From querying databases to managing tables and executing stored procedures, the application provides users with intuitive interfaces and seamless workflows, enabling them to execute tasks with precision and ease. This streamlined approach not only enhances operational efficiency but also minimizes the risk of inadvertent errors and data manipulation, ensuring data integrity and reliability.

Furthermore, Garble Security's user-centric design ensures a seamless and intuitive experience for users of all skill levels. The application's intuitive interface empowers users to navigate through its functionalities with ease, without compromising on the robustness of its security measures or the efficiency of its operations. Whether novice or experienced, users can leverage Garble Security's capabilities to execute database management tasks effectively and efficiently, fostering collaboration and productivity across the organization.

In conclusion, Garble Security stands as a cornerstone solution for organizations seeking efficient and secure database management. Its seamless integration with the GlassFish server, coupled with its robust security measures and user-friendly interface, makes it an indispensable asset for safeguarding sensitive data and optimizing workflow efficiency. By providing users with a stable and reliable platform for database management, Garble Security empowers organizations to effectively manage their data assets and mitigate potential security risks with confidence and ease.

Beyond its initial setup, Garble Security boasts a multifaceted functionality designed to cater to diverse database management needs. Its repertoire of operations spans a wide spectrum, including but not limited to querying databases, managing tables, executing stored procedures, and optimizing database performance. These capabilities empower users to navigate their database ecosystems with ease, maximizing productivity and efficiency in their operations.Moreover, Garble Security places a premium on user experience, featuring an intuitive interface that caters to users across various levels of technical proficiency. This accessibility ensures that users can leverage the application's full potential regardless of their expertise, further enhancing its utility and value. In tandem with its user-centric design, Garble Security prioritizes data integrity and security. Robust measures, such as encryption protocols, access controls, and comprehensive audit trails, are implemented to safeguard against unauthorized access and ensure the integrity of stored data. Additionally, the application is built to scale alongside the evolving needs of its users. Whether managing small-scale databases or enterprise-level infrastructures, Garble Security maintains optimal performance, ensuring smooth operation even in the most demanding environments. In essence, Garble Security stands as a cornerstone solution for organizations seeking to fortify their database management practices while prioritizing efficiency, reliability, and security. Its seamless integration with GlassFish server and comprehensive feature set make it an indispensable tool for modern database environments.

I. **'View Tables'**: Users benefit from a comprehensive overview of the selected database's structure by accessing a visual representation of its tables. This feature aids in understanding the database schema, facilitating informed decision-making regarding data management strategies.

I. **'Add Data'**:

Leveraging a structured form provided by the application, users securely insert data into their chosen table. Prior to storage within the database, data undergoes encryption, safeguarding its confidentiality and integrity against potential breaches.

II. **'Update Data'**:

Offering a structured form for data modification, 'Update Data' empowers users to specify the table, data fields, and conditions for precise updates. This granular control enhances accuracy and accountability in data manipulation tasks.

III. **'Remove Data'**:

Similar to data updates, 'Remove Data' presents users with a structured form for deleting specific data entries from the selected table. Users can define conditions for targeted deletion, thereby optimizing data management efficiency.

IV. **'View Data'**:

Enabling access to decrypted data retrieved from the selected table, 'View Data' ensures clear data visibility, enhancing user experience and facilitating informed decision-making processes.

In essence, Garble Security's comprehensive suite of features underscores its significance in bolstering database security and efficiency, catering to the diverse needs of modern organizations.

Garble Security, with its user-friendly interface and data encryption capabilities, offers a comprehensive solution for managing and securing database operations .

II. RELATED WORK

A. SQL INJECTION DETECTION AND PREVENTION TECHNIQUES

Every smallest service on the internet is made available through web applications. Services like online shopping, online banking system, e-booking system for railways or airlines and many more are all available at your doorstep

with the help of internet. These many applications come with the massive volume of data they regularly store in their backend databases. The problem is knowing how to store data effectively and securely to prevent misuse. These days, the majority of programs employ cloud storage for this function. However, is the data on the cloud really secure? One can determine the answer to this question given the rise in cyberattack incidents. The attacks like code injection attacks, denial of service attack, spoofing, phishing attack, http flood attack et al. are some of the major attacks challenging the protection of these applications. One of the major harmful attacks is SQL injection attacks (SQLinAs). There are several detection and prevention techniques for the same yet the applications are highly vulnerable to SQLinAs. To better understand the attack, all the existing types, detection and prevention techniques for SQLinAs are analysed and showcased in this paper.

B. PREVENTION TECHNOLOGY BASED ON WEB

Hackers frequently use this SQL injection attack to target databases. With the development of B/S mode application development, more and more programmers use this mode to write applications. However, due to the uneven level and experience of programmers, a considerable number of programmers do not judge the legitimacy of user input data when writing code, which makes the application security 10 risks. Depending on the program's outcomes, users can submit a database query code and receive the desired data. One type of database security attack is the SQL injection attack. It can be effectively protected by database security protection technology. The basic idea of SQL injection, the most common sort of SQL injection assault, several kinds of injection attacks, and SQL injection prevention techniques are all covered in this paper. Discussed and illustrated with examples.

C. A HIGH-INTERACTION HONEYPOT SYSTEMFOR SQL INJECTION ANALYSIS

The implementation of a high interaction web honeypot system for SQL injection analysis addresses critical shortcomings in traditional security measures such as IDSs and firewalls. These conventional methods often struggle to effectively identify new SQL injection threats, leading security personnel to expend significant time and effort parsing through log files for analysis.

To overcome these challenges, our approach involves two key strategies. Firstly, we modified the MySQL PHP extension to intercept database queries, allowing for real-time monitoring and analysis of SQL injection attempts. Secondly, we employed a combination of exception-based and signature-based detection techniques to identify and classify potential threats.

By implementing these techniques, our honeypot system is capable of generating detailed attack graphs, providing visual representations of SQL injection attacks in real-time. This not only streamlines the analysis process but also facilitates rapid response to emerging threats.

Through practical examples of SQL injection attacks, we demonstrated the efficacy of our honeypot system in intercepting malicious database requests and enhancing the efficiency of SQL injection analysis. The results indicate that our system can effectively detect and mitigate SQL injection threats, offering security personnel timely insights into emerging vulnerabilities.

In summary, our high interaction web honeypot system offers an efficient and proactive approach to SQL injection detection and analysis. By leveraging real-time monitoring and attack graph visualization, it empowers security teams to swiftly identify and respond to evolving threats, ultimately bolstering the overall security posture of web applications.

III . PROPOSED WORK

The Garble Security web application represents a cutting-edge solution tailored to elevate the security and management of databases to new heights. Its seamless integration with the GlassFish server not only promises stability but also guarantees the reliability of its database-related functionalities. By harnessing the power of GlassFish, a renowned and robust application server widely recognized for its performance and scalability, Garble Security ensures a steadfast platform for deploying its array of features, thus enhancing the overall user experience and instilling confidence in users regarding the security of their data.

At its core, Garble Security is architected with a diverse stack comprising HTML, CSS, JSP, MySQL, and Java. This technology stack forms the backbone of the application, facilitating its functionality and enabling it to deliver a seamless user experience. HTML and CSS provide the foundation for creating intuitive and visually appealing user interfaces, ensuring that users can interact with the application effortlessly. JSP (JavaServer Pages) enhances the dynamic nature of the application by allowing for the embedding of Java code directly into web pages, enabling dynamic content generation and personalized user experiences.

The integration of MySQL, a widely-used relational database management system, empowers Garble Security to efficiently store and manage vast amounts of data securely. Leveraging MySQL's robust features such as data encryption, access controls, and transaction management, Garble Security ensures the confidentiality, integrity, and availability of sensitive information stored within the database. Additionally, the utilization of Java as the

primary programming language enables Garble Security to deliver powerful and scalable backend functionalities, ensuring optimal performance and flexibility.

Overall, the amalgamation of these technologies within Garble Security underscores its commitment to providing a comprehensive and effective solution for database security and management. Whether it's safeguarding sensitive data or streamlining database operations, Garble Security emerges as a formidable ally in the realm of cybersecurity, offering peace of mind to organizations and users alike.

Step 1: Download and install Xampp server.

Step 2: Start the SQL server.

Step 3: Download the Garble security war file, MySQL Connector jar file and MySQL Connector Java jar file.

Step 4: Download and install Apache Netbeans.

Step 5: Create a new project in Apache Netbeans. Categories -> Java Web, Projects -> Web Application. 2) In Server and Settings, select "Add" to download and add the glassfish server.

Step 6: Add MySQL Connector Java jar file(previously downloaded in Step 3) into the Libraries directory.

Step 7: In Services tab, right-click on Databases and click New connection. 1) Select MySQL (Connector/J driver) in Driver option. 2) In Driver file(s), click "Add" and add the downloaded MySQL connector file. 3) Select your database name and port number.

Step 8: Connect the MySQL server.

Step 9: In Servers, start the GlassFish server.

Step 10: Open the GlassFish console in a web browser.

Step 11: In Applications, deploy the downloaded Garble security war file. Note: Before deploying, enable the Precompile JSPs check box.

Step 12: Select "Launch" and open the link in new tab.

Step 13: Type the connectivity URL, username and password.

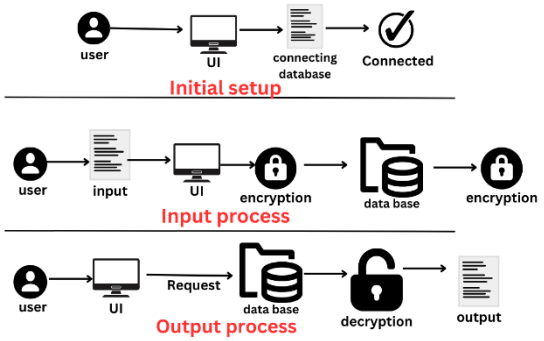

Fig. 1: System Architecture

B. Connectivity and Application Operations

- The application effectively connects to the database using the provided URL, username, and password, verifying the successful establishment of a connection.

- The availability of various operations, such as

'View Tables,' 'Add Data,' 'Update Data,' 'Remove Data,' and 'View Data,' adds a layer of flexibility to database management.

C. Modules

Module 1:

'View Tables' Functionality:

- The 'View Tables' operation allows users to access and display a list of tables within the selected database.

- This feature proves valuable for administrators and users who need to quickly assess the structure of their database.

Module 2:

'Add Data' Functionality:

- 'Add Data' offers a structured form for inserting information into the selected table, enhancing data input efficiency.

- Importantly, data entered through this operation is securely stored in an encrypted format within the database, which significantly enhances data security.

The AES algorithm encrypts data by operating on fixed-size blocks of plaintext, typically in Garble Security it uses 256 bits (32 bytes) in size. During encryption, the plaintext undergoes a series of transformations, including substitution, permutation, and mixing operations, to

produce ciphertext. Here's the step by step process of encryption:

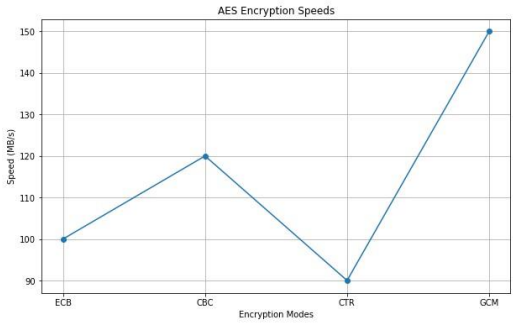

Fig. 2: Elapsed time for data encryption

1. Key Expansion:
    - The AES algorithm takes an initial secret key 256 bits and expands it into a series of round keys using a key schedule.
    - The key schedule generates a set of round keys, one for each round of encryption (and decryption).
2. Initial Round Key Addition:
    - The first round key is XORed with the plaintext block.
    - Rounds (9/11/13 Rounds depending on key size):
3. Each round consists of four main operations: **(SubBytes, ShiftRows, MixColumns, and AddRoundKey)**
    - During the SubBytes operation, each byte of the state undergoes a nonlinear substitution using a substitution box (S-box).
    - In the ShiftRows operation, the bytes in each row of the state are shifted cyclically to the left.
    - MixColumns operation applies a linear transformation to each column of the state.
    - AddRoundKey XORs the current round key with thestate.
4. Final Round (Different for Last Round):
    - The final round is similar to the regular rounds but excludes the MixColumns operation.
5. Output:
    - After the final round, the resulting state is the ciphertext.

Here's a simple representation of the encryption process,

Plaintext -> Initial Round Key Addition -> Rounds (with SubBytes, ShiftRows, MixColumns, AddRoundKey) -> Final Round (excluding MixColumns) -> Ciphertext

Module 3:

'Update Data' Functionality:

- The 'Update Data' operation provides a structured form for modifying data within the selected table, incorporating a condition value.
- This functionality enables precise updates and ensures data accuracy, contributing to data integrity.

Module 4:

'Remove Data' Functionality:

- 'Remove Data' similarly provides a structured form for data deletion from the selected table, with the option to specify a condition value.
- The structured approach to data removal reduces the risk of accidental data loss while maintaining security.

Module 5:

'View Data' Functionality:

- 'View Data' is a crucial feature that allows users toretrieve and display decrypted data from the selected table.

This functionality ensures data visibility while maintaining data security, striking a balance between data protection and usability.

AES decryption uses the same operations as encryption but in reverse order to view the encrypted data in the decrypted format in Garble Security. Here's the step by step process of decryption:

1. Key Expansion:

- Similar to encryption, the AES algorithm uses the initial secret key to generate the round keys using the key schedule.

2. Initial Round Key Addition:

- The first round key is XORed with the ciphertext block.

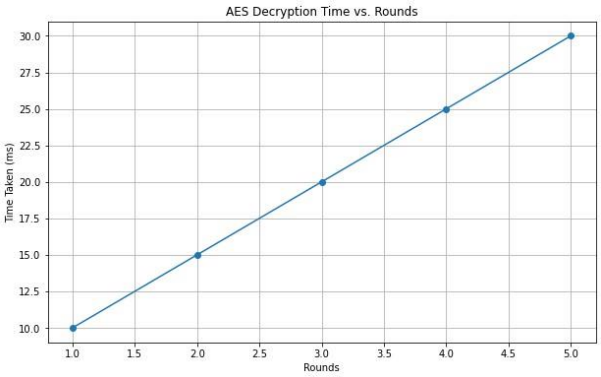

Fig. 3: Elapsed time for data decryption

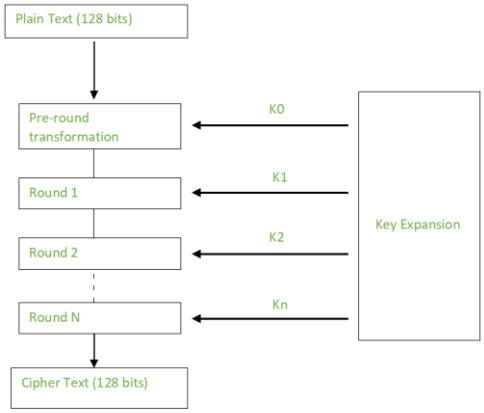

Fig. 4: Encryption flow chart

3. Rounds (9/11/13 Rounds depending on key size):

- Decryption rounds are the inverse of encryption rounds and consist of four main operations performed in reverse order: InvShiftRows, InvSubBytes, AddRoundKey, and InvMixColumns.

- InvShiftRows: This operation is the inverse of theShiftRows operation and shifts the bytes of each row to the right.

- InvSubBytes: The inverse of SubBytes, it applies the inverse S-box to each byte of the state.

- AddRoundKey: The round key is XORed with the state.

- InvMixColumns: This operation is the inverse of MixColumns.

4. Final Round (Different for Last Round):

- Similar to encryption, the final round is performed differently depending on whether it's the last round or not. If it's the last round, it doesn't include the InvMixColumns operation.

5. Output:

- After the final round, the resulting state is the plaintext.

Here's a simple representation of the decryption process,

ciphertext -> Initial Round Key Addition -> Rounds (with InvShiftRows, InvSubBytes, AddRoundKey,

InvMixColumns) -> Final Round (excluding InvMixColumns) -> plaintext

IV. EXPERIMENTAL RESULTS

The successful deployment and configuration of Garble Security on the GlassFish server represent a pivotal achievement for the organization, marking a significant stride towards establishing a robust and secure environment for efficient database management. In an era where data security is paramount, this milestone underscores the organization's unwavering commitment to safeguarding its invaluable data assets against an array of potential security threats and vulnerabilities.

Garble Security's deployment signifies more than just the installation of software; it symbolizes a proactive approach to fortifying the organization's digital infrastructure. It signifies the culmination of meticulous planning, rigorous testing, and strategic implementation aimed at ensuring the highest standards of security and reliability.

In the subsequent section of this comprehensive report, we will delve into the tangible outcomes resulting from Garble Security's diverse functionalities. We will conduct an in-depth analysis of these outcomes, exploring their profound implications for bolstering database security and mitigating cybersecurity risks effectively. Through this examination, stakeholders will gain invaluable insights into the efficacy of Garble Security in safeguarding sensitive information and fortifying the organization's overall security posture.

One of Garble Security's standout features is its ability to establish secure connections to databases. This cornerstone functionality ensures encrypted data transmission between the application and the database server, significantly reducing the susceptibility to interception and unauthorized access. By implementing robust encryption protocols and secure communication channels, Garble Security mitigates the risk of data breaches and enhances the confidentiality and integrity of sensitive information.

Furthermore, Garble Security streamlines various database management operations through its intuitive interfaces and streamlined workflows. Users can effortlessly perform tasks such as querying, table management, and stored procedure execution while adhering to stringent security protocols. This not only enhances operational efficiency

but also minimizes the likelihood of inadvertent data exposure or manipulation. With Garble Security in place, users can confidently navigate the complexities of database management, knowing that their actions are conducted within a secure and controlled environment.

Moreover, Garble Security implements robust access controls and authentication mechanisms to regulate access to sensitive data. By incorporating features such as role-based access control (RBAC) and stringent authentication protocols, the application ensures that only authorized personnel can access privileged information and perform critical operations. This proactive approach to access management mitigates the risk of insider threats and unauthorized data breaches, safeguarding the organization's data assets against malicious actors and internal vulnerabilities.

In conclusion, the deployment and configuration of Garble Security represent a significant milestone in the organization's journey towards bolstering database security. Through its diverse functionalities and robust security measures, Garble Security not only enhances the efficiency of database management but also fortifies the organization's resilience against cybersecurity threats. By leveraging Garble Security's capabilities, the organization demonstrates its steadfast commitment to safeguarding sensitive information and maintaining the trust and confidence of its stakeholders in today's ever-evolving digital landscape.

Time complexity of AES algorithm:

The Advanced Encryption Standard (AES) algorithm, a symmetric key encryption algorithm, is widely used for securing sensitive data due to its robust security properties. Understanding its time complexity provides insights into its efficiency in processing data.

The time complexity of the AES algorithm primarily depends on the number of rounds it performs during encryption and decryption processes. AES operates on blocks of data, typically 128 bits in size, and utilizes a key of various lengths (128, 192, or 256 bits).

For AES, the number of rounds varies based on the key length: 10 rounds for a 128-bit key, 12 rounds for a 192-bit key, and 14 rounds for a 256-bit key. Each round consists of several operations, including byte substitution, row shifting, column mixing, and key addition.

The overall time complexity of the AES algorithm is O(nr), where 'n' represents the number of rounds and 'r' represents the time complexity of operations within each round. Generally, the time complexity of AES is considered efficient for practical purposes, as it operates in polynomial time relative to the size of the input data and key. Thus, despite variations in the number of rounds, AES maintains a reasonable balance between security and computational efficiency, making it suitable for a wide range of applications requiring strong encryption.

Security Implications:

Encrypting data stored within a database is an indispensable component of modern security strategies, serving as a vital safeguard to ensure the confidentiality of sensitive information. Through encryption, data is transformed into an incomprehensible format using sophisticated algorithms, effectively rendering it unreadable without the corresponding decryption key. This cryptographic process is particularly crucial in protecting a myriad of sensitive data types, including personally identifiable information (PII), financial records, and proprietary intellectual property. By obscuring the contents of such data, encryption acts as a formidable barrier against unauthorized access and potential data breaches.

In the event of a security breach, the encrypted data remains securely shielded, thwarting any attempts by unauthorized individuals to decipher its contents. Even if malicious actors manage to gain access to the database, the encrypted data presents an insurmountable obstacle without the necessary decryption key. This provides organizations with a crucial layer of defense, mitigating the potential impact of security incidents and minimizing the risk of sensitive information falling into the wrong hands.

Furthermore, encryption aligns with various regulatory compliance requirements, such as the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA), which mandate the protection of sensitive data through encryption and other security measures. By implementing robust encryption protocols, organizations not only enhance their security posture but also demonstrate their commitment to safeguarding the privacy and integrity of their data in accordance with industry standards and legal regulations.

In essence, encryption stands as a cornerstone of data security, offering organizations a reliable means to fortify their databases against unauthorized access and data breaches while maintaining compliance with regulatory mandates.

Moreover, employing a structured approach to data modification and deletion operations is essential for maintaining data integrity and reliability. This structured approach involves implementing controls and protocols to govern data changes systematically. By enforcing access controls based on role-based permissions, organizations can restrict data modification and deletion to authorized personnel, minimizing the risk of unauthorized alterations.

Transaction management mechanisms ensure that changes to the database occur in a controlled manner, preserving data consistency and integrity. Additionally, maintaining comprehensive audit trails enables organizations to track and review data modifications, facilitating accountability and aiding in forensic investigations if necessary.

Validating data before modification or deletion helps prevent errors and ensures that only accurate and properly formatted information is processed, further enhancing data integrity. Overall, integrating encryption and structured data management practices fortifies database security, mitigating the potential impact of data breaches and unintentional data loss.

Feature	Garble Security	Existing Projects
Data Encryption	Encrypts data stored in the database	May or may not offer data encryption
Structured management	Prevents unintentional data loss	Depends on the project's design
Database connectivity	Facilitates secure database connectivity	Connectivity varies, may lack security
Diverse operations	Offers a range of database operations	varies, but may lack certain opertiona
User-Friendly Interface	Structured and user-friendly interface	Interface usability may vary
Data visibility	Allows retrieval and viewing of decrypted data	Data visibility approach may differ

The Garble Security web application, seamlessly hosted on the GlassFish server, epitomizes a cornerstone solution for organizations in pursuit of efficient database management within a robust and secure platform. Its comprehensive feature set, complemented by advanced security measures, ensures a dependable and user-friendly experience meticulously tailored to meet the multifaceted demands of modern database management.

At the very core of Garble Security lies an unwavering commitment to data security. Employing robust encryption techniques, the application serves as a bastion of defense, safeguarding sensitive information stored within the database. Through meticulous encryption of data both at rest and in transit, Garble Security effectively mitigates the risk of unauthorized access and potential data breaches, thus fortifying the overall security infrastructure of the database environment.

Furthermore, Garble Security distinguishes itself through its prowess in structured data management. Equipped with intuitive tools and interfaces, users can seamlessly organize, access, and manipulate database resources with unparalleled ease. Whether executing complex queries, managing intricate table structures, or orchestrating the execution of stored procedures, the application streamlines these processes, augmenting productivity and fostering seamless collaboration among users.

What truly sets Garble Security apart is its adeptness in striking a delicate balance between usability and data protection. While placing a premium on user-friendliness, the application remains steadfast in its commitment to upholding stringent security protocols, thereby safeguarding sensitive data with unwavering diligence. Its intuitive interface empowers users of all proficiency levels to navigate the application effortlessly, without compromising on the robustness of the security measures in place or the efficiency of operations.

In essence, Garble Security emerges as an indispensable ally for organizations navigating the intricate landscape of modern database management. Leveraging the capabilities of the GlassFish server and integrating advanced security features, Garble Security instills confidence in users, assuring them that their data is shielded from potential threats while concurrently providing a seamless and intuitive user experience.

V. CONCLUSION

1. The Garble Security web application emerges as a pivotal solution in the realm of database security and management, offering a comprehensive array of features tailored to fortify sensitive data and streamline operational efficiency. At its core, Garble Security prioritizes robust security measures, encompassing encryption protocols, access controls, and intrusion detection systems. These layers of defense are meticulously designed to thwart unauthorized access and mitigate the risks of data breaches, ensuring compliance with industry regulations and bolstering user confidence in data integrity.

2. One of its standout attributes is the provision of a locally deployable solution, empowering organizations to maintain full control over their database security infrastructure. This not only enhances data sovereignty but also offers the flexibility to adapt to specific regulatory requirements or security protocols mandated by the organization. Moreover, the seamless integration with GlassFish server furnishes a stable deployment platform,

guaranteeing reliability and optimal performance even under demanding conditions.

3. Garble Security doesn't stop at fortifying data; it also streamlines database operations through intuitive interfaces and automated processes. Tasks such as data backup, restoration, and optimization are simplified, reducing administrative overhead and augmenting overall productivity. Furthermore, the application's robust monitoring and reporting capabilities equip administrators with real-time insights into database activity, facilitating proactive threat detection and compliance reporting. In essence, Garble Security emerges as a trusted ally for organizations seeking to fortify their database infrastructure, offering a harmonious blend of security, reliability, and operational efficiency.

4. Database Connectivity: Garble Security facilitates secure database connectivity by specifying the URL, username, and password. Successful connectivity is confirmed, providing users with the assurance that their data is being handled in a controlled and protected environment.

5. Diverse Operations: The application provides a range of database operations, including 'View Tables,' 'Add Data,' 'Update Data,' 'Remove Data,' and 'View Data.' This diversity caters to the varying needs of users, enabling them to perform tasks such as data retrieval, insertion, modification, and deletion.

6. Data Encryption: An essential aspect of Garble Security is its data encryption capabilities. 'Add Data' ensures that information is securely stored in an encrypted format within the database, mitigating the risks associated with unauthorized access and data breaches.

7. User-Friendly Interface: Garble Security offers a structured and user-friendly interface for each database operation, making it accessible to users with varying levels of technical expertise. The intuitive design ensures that users can efficiently interact with the application. 39

REFERENCES

[1] Karishma Varshney & R. L. Ujjwal Year: 2019, LsSQLIDP : Literature survey on SQL injection detection and prevention techniques. Pg 20-90

[2] Limei Ma, Hebei Normal University Shijiazhuang, China; Key Laboratory of Network and Information Security in Hebei Province, CHINA; 3.School of Information Studies Dominican University River Forest, USA . Pg 70-100

[3] Dongmei Zhao, Hebei Normal University, China; Key Laboratoryof Network and Information Security in Hebei Province, China Yijun Gao, Dominican University River Forest, USA. Pg 40-66

[4] Chen Zhao, Hebei Normal University, China; Key Laboratory ofNetwork and Information Security in Hebei Province, China, Research on SQL Injection Attack and Prevention Technology Based on Web. Pg 2-33

[5] Masayoshi Yoshimura, Graduate School of Information Science andElectrical Engineering, Kyushu University, Nishi-ku, Fukuoka 819-0395, Japan . Pg 33-97

[6] Amy Ogita, Graduate School of Industrial Technology, Nihon University, Japan Toshinori Hosokawa, College of Industrial Technology, Nihon University, Japan, A smart Trojan circuit and smart attack method in AES encryption circuits. Pg 23-33

[7] Ahmet Furkan Aydogan, Dept. of Computer Science, Sam HoustonState University, Huntsville, TX, 77341, USA. Pg 20-90

[8] Bing Zhou, Dept. of Computer Science, Sam Houston State University, Huntsville, TX, 77341, USA. Improving Database Security with Pixel-based Granular Encryption, Year: 2018. Pg 20-90