Cyber Security Policy: GenStore

Submitted by,

CHAARU MANJURAJ KOMALA (ID: 2406827)

1. Introduction

Genstore provides a cloud based secure storage repository/database for medical DNA data. There are three actors, Cloud provider, Data Publishers, who can read and write and Data Readers, who can only read data.

2. Cyber security threats

Threats identified (in the order of ranking):

1. **Man-in-the-Middle**
   The man-in-the-middle attack is an active attack where the attacker positions himself in the middle of the network. Since it mostly happens between the user and the server, GenStor is highly vulnerable to the MiTM attacks.Mitigations:
   - Tamper Detection helps to identify whether the data has altered or not.
   - Enabling HTTPs, TLS Encryption encrypts the HTTP traffic, making it impossible to read/alter by the attacker.

2. **Injection attacks**
   In an injection attack, an attacker supplies an untrusted input to a program. Injection attacks can lead to data loss, loss of data integrity as well as the previously described denial of service.
   Mitigations:
   - All inputs to the database should be sanitised, with parameterised queries and proper formatting is extremely important.
   - Apply the principle of least privilege to prevent any data publishers and data readers from modifying/extracting restricted data.

3. **Insecure Identity and Access Management (IIAM)**
   GenStore holds medical DNA data which is classified as high risk PII. When IAM is not configured correctly, it will allow processing of illegitimate requests giving adversary access to the data and system impacting confidentiality and integrity.
   Mitigations:
   - Multi Factor authentication
   - Using strong passwords or token based authentication
   - Setting up policies and rules correctly in IAM system

4. **Insecure APIs**
   Genstore may expose APIs for accessing the data stored in DB. Unprotected endpoints will allow adversaries to exploit and read/modify sensitive data.

Mitigations:

- All endpoints should go through authentication and authorisation.
- Server-side input validation should be enabled
- Setup an API gateway.

5. **DoS Attacks**

   Denial of service (DoS) attacks aim to conquer a target by exhausting all, or most of its available resources. DoS Attacks can also be interpreted as any attack that prevents users or owners of services from accessing or managing the service. Network DoS attacks leverage the request-response architecture. GenStore is susceptible to DoS attacks as it is a network-based service.

   Mitigations:

   - Having an intelligent network border filter can prevent many of these requests from entering the network.
   - Working with ISPs to block suspiciously large streams of data from even being forwarded at the service would help prevent a majority of DoS attacks.

6. **Port scan attacks**

   Adversary can consistently scan cloud providers IP ranges to locate machines with exposed management endpoints. Once detected, they can try to brute force and exploit these end points, taking advantage of new machines with insecure configurations.

   Mitigations:

   - Install a Firewall: A firewall can help prevent unauthorised access to private networks. It controls the ports that are exposed and prevents port scan attacks.
   - TCP Wrappers: TCP wrapper gives administrators the flexibility to allow or deny access to the servers based on IP addresses or domain names.

7. **Ransomware**

   Ransomware is malware that prevents the user from accessing their filesystem by encrypting the files. This attack affects the availability of the data hosted in GenStore.

   Mitigations:

   - Perform regular back-ups: Regular back-ups are necessary to restore the data in case of an attack.
   - Harden the endpoints: Network endpoints should be secured and monitored.
   - Implement Intrusion Detection Systems: IDSs help in detecting unusual behaviour of apps and network traffic.
   - Keep the security patch up to date: Keep the systems patched all the time. If possible, can use the automated patch management system.

8. **Social Engineering**

   Adversary can exploit an existing user's credentials or send emails to known system users with malware embedded to gain access to Genstore DNA database and then proceed with a correlation attack.

   Mitigations:

   - Multi Factor Authentication to protect user accounts. GenStore should provide the users with a specific/ registered device to complete their logins, so that the credentials which were stolen in phishing attacks become useless.

We have ranked by which attack may be executed more frequently as well as taking into consideration the nature of the service.

3. Suitable security protocols/frameworks/systems/etc

Initial Architecture:

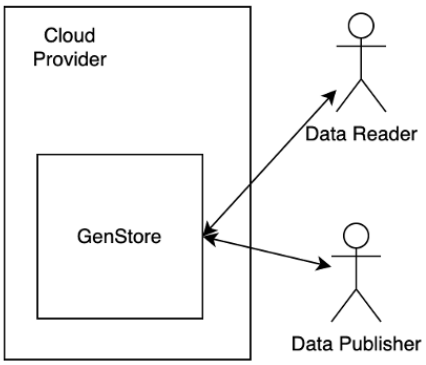

Figure 1: Initial Architecture

Proposed Architecture:

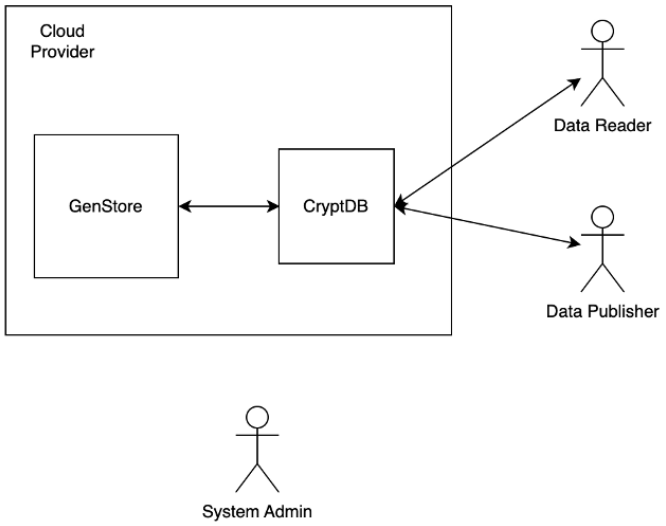

Figure 2: Proposed Architecture

1. CryptDB

CryptDB is a secure middleware between the application server and the database. It encrypts the data ensuring confidentiality of the data is maintained. It also encrypts the DB schema i.e tables and column names.

CryptDB uses the below key approaches:

- Query is executed on encrypted data
- Uses symmetric key encryption
- Runs on unmodified database software and uses user-defined functions
- Uses onions of encryption to ensure there is no data leak at any step
- User passwords are chained with encryption keys

2. Searchable Encryption for DNA Searching

Searchable encryption enables a server to search for documents in a database given a query from a client. Each encrypted document is seen as a set of keywords and an inverted index is used to search for documents. The result of the search is a set of encrypted documents that contain the keyword (inverted index) .

Genstore can deploy this using CryptDB to search the DNA data in the database.

3. Multi-Factor Authentication

It's a verification method that involves two or more factors to verify and gain access to the information.
Multi-Factor Authentication (MFA) is the combination of any of the following methods (including the same class) of user authentication.

1. Something you know (Passwords)
2. Something you have (Digital Signature)
3. Something you are (Biometrics)

The benefit of MFA is the requirement for the users to validate themselves with more than a username and a password. Passwords are vulnerable to brute-force attacks or can be stolen by a third party. MFA can prevent password attacks by enhancing the security of the organisation.

4. Secure Protocol Choice

Security Protocols mainly cryptographic or encryption protocols help protect sensitive data, medical data, etc.

The best practice is to encrypt all database connections using the Transport Layer Security (TLS) protocol, protecting data in transit. It establishes a secure connection by performing a handshake between two machines using a key exchange, cipher, and authentication, then encrypts the data.

IPsec helps protect the confidentiality and integrity of data as it travels across less-trusted networks. Virtual Private Network (VPN) is established to implement Network-based encryption using the IPsec protocol. VPNs are encrypted network connections which allow remote users to securely access an organisation's services.

5. Active system administration

Active System Administration is vital to keep the system up and running. Manually monitoring resources like CPU Usage, Disk Usage, Latency, and DNS can prevent some attacks like MiTM, and DDoS. System Administration also involves managing user permissions and roles that can help to create the barrier and absolute encapsulation between the layers. Other tasks like managing SSOs and Passwords and defining system usage policies will greatly help prevent misuse of the resources and secure the system.

An active system administration will help in preventing attacks, thefts, and misuse of resources. Administrators can ensure uptime, resources, services, and security that they manage will meet the users.

6. Anonymous data download/upload

Anonymous access to servers, including read and write capabilities can be done in a couple of ways. One way is to allow read and write to the database without any authentication, this is

clearly not within our scope and an alternative approach needs to be used. Using symmetric keys between each client and CryptDB ensures that communication between the client and the proxy is only with trusted users.

To prevent being able to tell what operation the identity has performed on the database, an algorithm can be implemented to distribute read and writes throughout the memory of the server. This includes writing to a random memory slot during every read, as well as reading from a random memory slot during a memory write.

Data flow diagram:

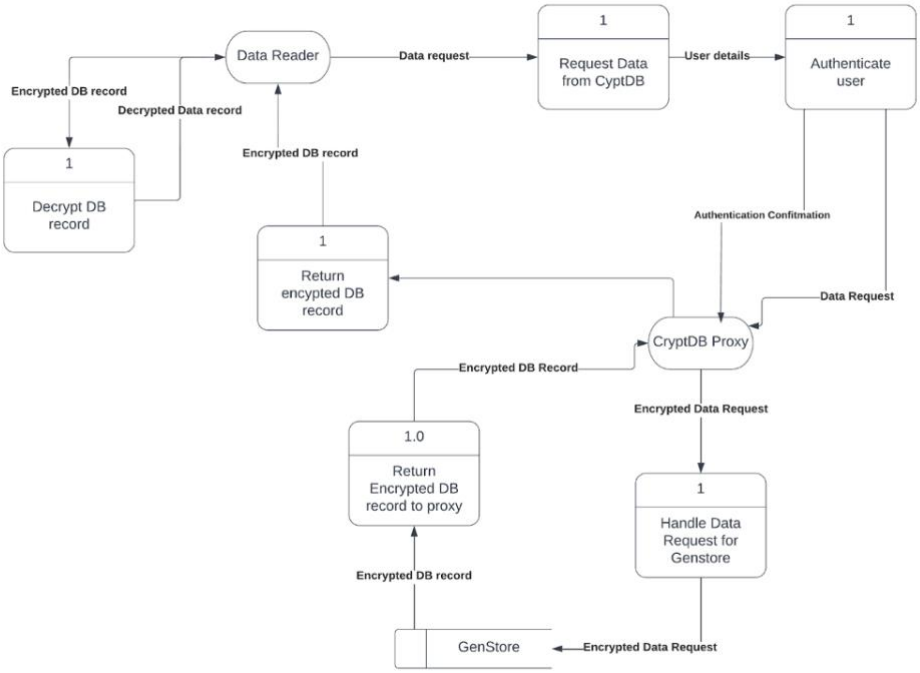

Figure 3: Data Reader workflow

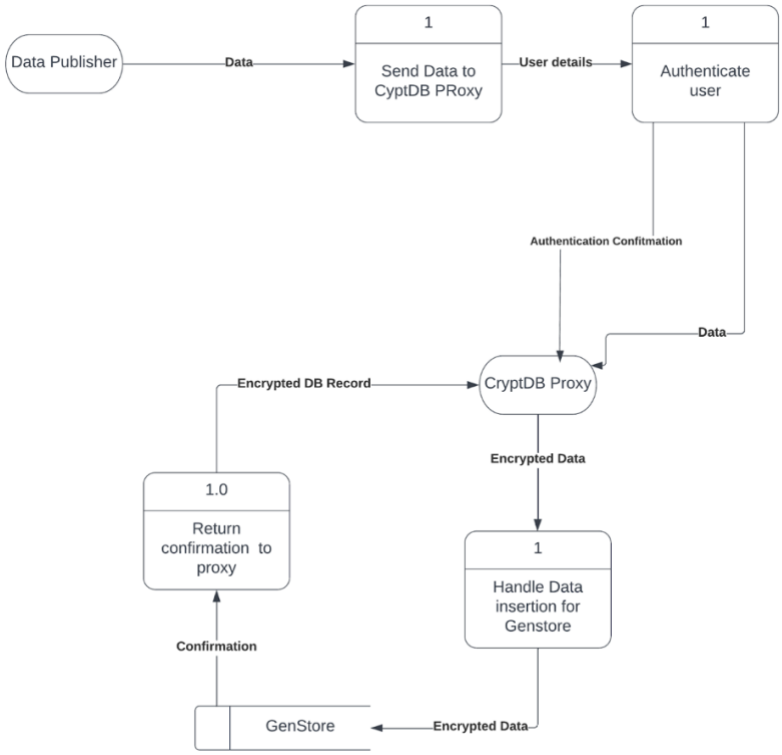

Figure 4: Data Publisher workflow

Important workflows:

Data Publisher Writing Data

1. The Data Publisher will verify his identity by a *Multi-Factor Authentication* method. Multi-Factor Authentication allows the users to validate themselves with more than just a username and a password.
2. Once the user identity is verified, the publishable data is sent to the CryptDB Proxy with the TLS encryption enabled.
3. CryptDB Proxy will have access to the *Master Key* and the *Schema Onion Levels*. With the help of them, it encrypts the data.
4. Within the normal SQL interface, the proxy stores the encrypted data in an unmodified DBMS (E.g., PostgreSQL).

Data Reader Reading Data

1. The data reader issues a query, which the CryptDB proxy intercepts and rewrites it
2. Proxy anonymises each table and column name. CryptDB's proxy stores a secret master key MK which is used to encrypt the query. It uses the following encryption types:
    - Random (RND) - Random encrypts the query using a randomly generated initial vector (IV). This is the most secure layer in CryptDB encryption.
    - Deterministic (DET) - This layer is less secure than Random. It uses deterministic encryption to encrypt the query. Deterministic encryption can reveal whether or not 2 encrypted texts have identical values.

- Order-preserving encryption (OPE)- It preserves the order of the ciphertext as it was in plaintext. For example, if x<y then enc(x)<enc(y)
- Equality-join and range-join- It uses same key for multiple database columns

3. The proxy then forwards the encrypted query to the Database server, which executes it using standard SQL
4. CryptDB Proxy will decrypt it using the Private Key and fetch the data from GenStor Database. Here the private key is chained to the user password to enable data decryption so that no unauthenticated user can access the data.
5. The Database server then returns the encrypted query result, which the proxy decrypts and returns it to the data reader

Data Reader Searching Data

1. The search query will be encrypted by the proxy server and sent to the DB
2. The searchable encryption scheme searches for the specified keyword and returns the encrypted results. Searchable encryption is implemented in the following way-
    - Setup - In this step, a security parameter is taken as input and symmetric key is generated as output
    - Token - In this step, the security parameter and symmetric key is taken as input and a search token is given as output
    - Search - In this step, a search algorithm is used which takes the search token as input, generated in Step 2 and outputs the encrypted data that matches with the search token
3. The results will be decrypted and shown using the private key

4. Evaluation of the security of the enhanced system

Attacker Model:

The threats have been identified assuming that GenStore's adversary follows a malicious but cautious attacker model. Below are the assumptions of this model:

- Adversary is an active attacker
- Will try to remain undetected and leave no trace
- May not obey the protocol
- Will try to get access to the database and data by listening over the network and modify it
- Will try to impact availability of the database

What security our proposed system provides/prevents:

CryptDB:

CryptDB does not affect users and application developers, this allows for some freedom when creating and updating a security system as the user end can be segregated. The implementation of CryptDB introduces a middleware for all communication between users and the cloud provider, meaning that the CryptDB proxy intercepts all SQL queries between the user and GenStore. CryptDB adapts the regular SQL queries of the data reader into an encrypted format so that it can query over encrypted data, thus hiding the search patterns of a user to any adversary. This addresses two

threats, a curious database administrator who has control of the DBMS. CryptDB leverages the idea of chain encryption keys, this means that each record in GenStore can only be decrypted using a series of passwords related to the user that is trying to access the data.

Anonymous data download/upload:

With all the security considerations listed, an adversary that has complete control may still be able to infer things about particular user's and their search/query patterns as they may be able to discern whether a read or write has been performed. An implementation of ORAM prevents an adversary from being able to extract non-trivial information about the accessing of the database. Additional overhead is added to each access, this is to prevent more timing analysis on each access. If every access to the database appears the same, including the amount of time that it takes for that access then it will be increasingly difficult for an attacker to discern between the different operations.

Searchable Encryption:

Previously, in the original architecture, Genstore was not encrypting the data before storing it into the database, thereby exposing the data to the adversary. When the database is not encrypted, the attacker can easily obtain and tamper the confidential data using SQL injection or Man in the Middle attacks. They can run malicious queries over the DNA database or can effectively take over the communication process and intercept data, either by copying it or altering it before it is sent along to the patient. Once they intercept the information in this way and unencrypt it, they can either use it to harm the patient or they could perform ransomware attacks on the database.

In the proposed architecture, Genstore is using searchable encryption to search data in the DNA database. In this case, both the search token and DNA data is encrypted. To access the database and generate the search token, the user needs to have the security parameter and the symmetric key. The database becomes less prone to attacks like SQL injection or MITM because no unauthorised user can access the data without these two parameters. After giving these two parameters as input, the cloud looks for the search token in the database and sends the output to the proxy server. The user then needs to give their private key to decrypt the data.

Secure Protocol Choice:

In the proposed architecture, Genstore is using TLS protocol to protect the DNA data from MITM attack. TLS certificates safeguards the data by encrypting it with the secret key which is only known to Genstore. So, MITM attackers will not be able to read or tamper the encrypted DNA data without the knowledge of the secret key.

TLS provides the below mentioned advantages:

1. Private Connection: TLS encrypts the data using symmetric key cryptography. Keys for each connection are generated uniquely based on the shared secret during the TLS handshake process. The client/server negotiates an encryption algorithm and cryptographic key to use before the data is transmitted.
2. Communication Party Identity: Authenticated using the public-key cryptography.
3. Reliable Connection: Each message has an integrity check using a message authentication code (MAC) to prevent modification of data during the transmission.

Multi-Factor Authentication:

The architecture shows direct user interaction with the CryptDB. Users must authenticate themselves with the CryptDB to do operations on the data that is stored on the unmodified DBMS.

The primary authentication works with a username and a password. The passwords are stored as keys in the CryptDB Proxy in their *hashed* form. Users are compelled to use a strong password at the time of registration. A real-time verification will be performed at the time of logging in. As a secondary factor of authentication, either a biometric thumbprint or an OTP to the registered mobile number is recommended in the proposed architecture.

Password and MiTM Attack primarily target during the request and response. As the traffic is encrypted with TLS, the attacker will not be able to see the response in plain-text mode. Additionally, the introduction of a secondary authentication factor will help in preventing Social Engineering attacks or poor handling of passwords by the user.

Active System Administration:

An efficient role of System Administration is proposed as it is a vital component for overall system security. Active attacks such as port scans, DDoS, and MiTM involve sending modified packets to the network. There is a good chance that they may be modified to bypass the automated firewalls and Intrusion Detection Systems. System Administrators keep an eye on the incoming traffic and any sort of unusual behaviour of the memory which will lead to immediate notice and mitigation of the attacks.

System Administration also strictly controls the passwords and security policies of the users and the crew to keep the system workflow easier and secure from malicious insiders.

5. Conclusion

The proposed architecture adds layers of security to GenStore. It offers confidentiality of the stored data from the cloud provider, ability to search without revealing the search pattern and anonymous data download/upload.

6. References

1. CryptDB-sosp11.pdf (berkeley.edu)
2. Searchable symmetric encryption capable of searching for an arbitrary string - Uchide - 2016 - Security and Communication Networks - Wiley Online Library
3. CryptDB: A Practical Encrypted Relational DBMS - Raluca Ada Popa, Nickolai Zeldovich, and Hari Balakrishnan
4. Common Vulnerability Scoring System Version 3.1 Calculator
5. https://www.ncsc.gov.uk/collection/10-steps/identity-and-access-management
6. https://www.directives.doe.gov/terms_definitions/personally-identifiable-information-pii
7. https://owasp.org/www-project-top-ten/2017/A2_2017-Broken_Authentication
8. https://www.isc2.org/Articles/the-threat-of-insecure-interfaces-and-apis#
9. https://www.optiv.com/insights/discover/blog/insecure-api-cloud-computing-causes-and-solutions
10. https://cloudsecurityalliance.org/blog/2022/07/30/top-threat-2-to-cloud-computing-insecure-interfaces-and-apis/