CS - 431

CNS Project Design - 12

Chakradhar Basani (22110054), Koleti Eswar Sai Ganesh (22110123),
Sriman Reddy (22110124), Manav Jain (22110140), Nakka Naga Bhuvith (22110163),
Pavan Deekshith (22110190), Venigalla Harshith (22110283)

Security Guard

Project Overview:

In this project, you will implement a *secure log* to describe the *state of an institute*: the guests and employees who have entered and left, and persons that are on campus in different buildings or rooms. The log will be used by *two programs*. One program, logappend, will append new information to this file, and the other, logread, will read from the file and display the state of the institute according to a given query over the log. Both programs will use an authentication token, supplied as a command-line argument, to authenticate each other. Specifications for these two programs and the security model are described in more detail below.

Languages/Frameworks:

- C/C++ for logic implementation/file handling etc
- Bash for automation and to script executables
- openSSL for cryptographic functions
- Git for version control
- SQL for MySQL Database management.

The Architecture and Functionality of the system (Components/Modules):

- **Main Server**
  - Maintain visitor information and handle transaction requests.
  - Receives and validates input data from `logappend`, applies encryption, and stores it securely.
  - Authenticate requests and ensure secure communication.
  - Handles requests from clients, maintaining ACID constraints.
- **Guard Client**
  - Maintaining Input validation from the client request.
  - The client and main server communicate over TCP/IP. The client sends transaction requests to the server, which processes the requests and responds with the transaction status or query information.

- **Logappend module**

Functionality	**Description**
`-T timestamp`	Time the event is recorded
`-K token`	Token used to authenticate
`-E employee-name`	Name of Employee
`-G guest-name`	Name of guest
`-A`	Used to signify the arrival of an employee
`-L`	Used to signify the departure of an employee
`-R room-id`	Specifies the room ID of an event
`log`	Path to the file
- B	Specifies the batch file of commands

- **Logread module**

Functionality	**Description**
`-K token`	Token used to authenticate
`-S`	To print the current state of the log to stdout

`-R`	List of all rooms entered by a guest or an employee.
`-T`	Total time spent inside the campus by a guest or an employee.
`-E`	Name of the Employee
`-G`	Name of the guest
`log`	Path of the file log used for recording events

- **Security Module and features**
  - **Authentication and Validation:** Each `logappend` and `logread` command requires a key, which acts as a password. Only authorized keys are allowed to perform these operations.Commands are validated against a list of permissible actions associated with the key to ensure proper authorization.
  - **Data Encryption:** Logs are encrypted before being stored in the database to ensure that unauthorized access to the database does not compromise sensitive information.Both data in transit (between client and server) and data at rest (stored logs) are encrypted using strong encryption algorithms (e.g., AES-256).
  - **Audit Logging:** All actions (log additions, reads, failed attempts) are recorded in a secure audit log that is protected from tampering. This helps in tracking all actions performed in the system.
  - **Security Alerts:**. Security alerts are generated for suspicious activities, such as repeated failed access attempts, unauthorized data retrieval.
  - **Rate Limiting:** Limits the number of requests a user can make within a specific time period to prevent database overload and protect against denial-of-service (DoS) attacks, thereby maintaining system stability and security.
  - **Timestamp Consistency Check:** The program checks that the timestamp of a new entry is later than the most recent timestamp in the log before appending it. If the timestamp is not valid, the program will display "invalid" and terminate with an error.
- **Communication Module:**
  - Handles the communication between the client and the server, ensuring that all requests are securely transmitted. Includes functions to set up connections, send and receive requests, and handle any communication errors.
    - Socket setup and management
    - Data serialization and deserialization
    - Error handling for communication failures

- **Database Module:**
  - Implement functions for storing and retrieving and updating information.

- Maintain a log of all requests for auditing and rollback in case of errors.
- Ensure ACID constraints of requests to prevent issues like double spending or unauthorized access.

Work Distribution :

Name	**Responsibility**
Chakradhar Basani (22110054)	Client module
Koleti Eswar Sai Ganesh (22110123)	Client module
Sriman Reddy (22110124)	Main Server module
Manav Jain (22110140)	Main Server module
Nakka Naga Bhuvith (22110163)	Main Server module
Pavan Deekshith (22110190)	Security and integration module
Venigalla Harshith (22110283)	Security and integration module