Studio Project
Design and Evaluation

Team 1

June 2021

1. Table of Contents

Index of Tables

Index of Figures

2. Document Overview

2.1 Purpose and Scope

This document is the result of the Studio Project given a task during the CMU's Security Specialist course, and is written for evaluation of the task. The main content mainly deals with system design and vulnerability analysis, evaluation of implemented results, and summarizes and reconstructs outputs.

Table 1. Terminology definition

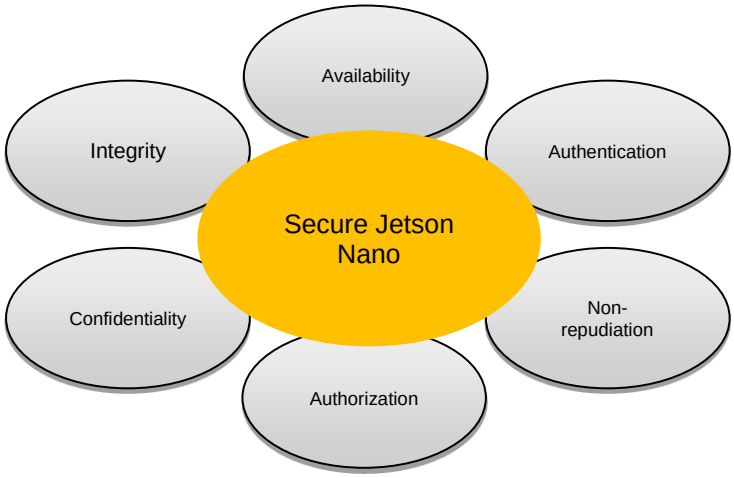
Terminology	Description
Availability	The degree to which a conferencing system is in a specified operational and committable state when requested at any time, and the probability that the conference system will perform satisfactorily. Conferencing software and systems must be available 24/7.
Vulnerability	Common types of software flaws that lead to vulnerabilities include: Memory safety violations, Input validation errors, Privilege-confusion bugs, Privilege escalation, Race Condition, Side-channel, User Interface Failure The content should not be known even if snipping is attempted by ensuring that data is encrypted between participants of the meeting.
Quality	Video quality is a characteristic of a video passed through a video transmission or processing system that describes perceived video degradation. Video processing systems may introduce some amount of distortion or artifacts in the video signal that negatively impacts the user's perception of a system. For many stakeholders in video production and distribution, assurance of video quality is an important task. Video and voice Quality of the conferencing system should be clearly recognizable at any case.
Non-repudiation	Non-repudiation refers to a situation where a statement's author cannot successfully dispute its authorship or the validity of an associated contract. Non-repudiation in conferencing software includes non-repudiation of logins and non-repudiation of incoming/outgoing chat messages. For this, the participant log and message sending/receiving log must remain on the server.
Denial-of-service (DoS)	Although this item appears to be overlapped with availability, here it means availability by intentional attack. The server must be available even for attacks by malicious users.
Intrusion	No other uninvited user should be aware of the meeting.

3. **Project Overview**

3.1 Introduction

LGE security specialist Team 1 developed a Secure Jetson Nano for the project of '2021 LGE Security Specialist course' in Carnegie Mellon University.

This document contains a security concept that aims to fulfill the security development process. The secure Jetson Nano is developed to cover the following processes



3.2 Stakeholders

The stakeholders for this project are as follows. The stakeholders are the team members needed to develop the system, the CMU professors who will evaluate it, and the people who developed the system we will evaluate during Phase 2.

Table 2: Stakeholders

Role	**Description**	**Name**	**Responsibility**
Project Manager	Project Coordinator Schedule management	Jonghyun Park	Organize the team and manages project resources by planning. Manages the relationship with project stakeholder and mentor
Architect	System Design Implementation review	Sungjin Lee	Create strategy of system architecture development Detailed design of server and communication protocols
Developer	Definition protocol Implementation secure channel	Seokwang Kim Jaewon Lee	Have ownership of development server application

	UX Design Client Application Protocol Design	Yeonbi Shin Sungsoo Kim	Detail design of client application
Tester	Create Test case Fuzz testing & Static Analysis	Jonghyun Park Sungjin Lee Jaewon Lee	Create test strategy and test cases Ensure the products meet standards of quality
Customer	The person who commissioned the project.	CMU faculty. Other Team	These are the people who commissioned and appraised this project.
Team2	People who developed a system to evaluate Phase2.	Team 2 members	Implement and provide us with tasks based on customer requirements.

3.3 Project Schedule

The schedule for the entire project is as follows. For the first three weeks, the company will identify customer requirements and proceed with design of safe systems. The second half of the week evaluates the implementation of other competing teams.

Week1 (Planning & Analysis)

- Project Planning - Team Role Assignment	- System Architecture & Requirement	- Risk Assessment - Threat Analysis	- Functional Requirement - Quality Attribute

Week2 (Analysis & Design)

- Security Requirement - Mitigation	- Overall Architecture Design - Protocol Design - Sample code analysis	- Secure Design - Server and Client Implementation	- Test case - Implementation

Week3 (Implementation & Test)

- Secure coding	- Interoperability test - Static Analysis	- Test - Modification	- Packaging - Release - Presentation

Week4 (Phase2 Planning and Analysis)

- Project Panning	- Role assignment - Architecture Review	- TARA - Functional Test	- Code Review - Static Analysis

Week5 (Test and Finalize)

- Open Source check - Dynamic test	- Penetration Test	- Documentation	- Finalize project

4. Phase1 – System Design

4.1. System Design

4.1.1 Overall System Context

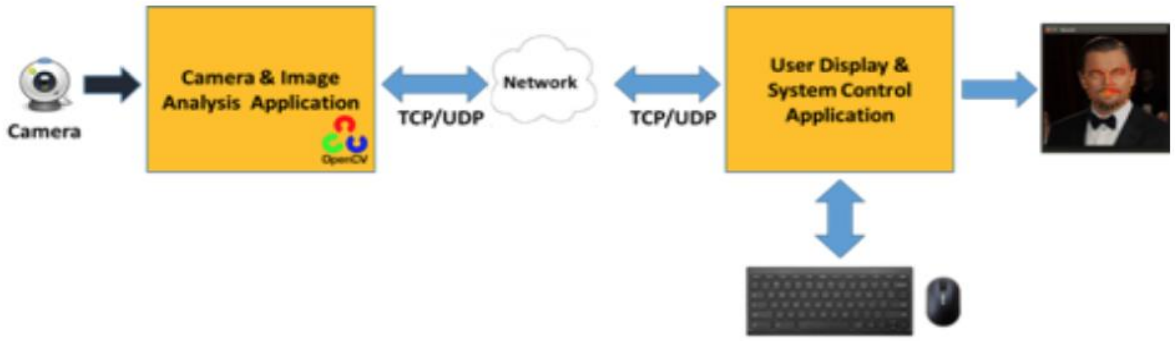


Figure 1. Overall System Context

The entire system is to connect to a server using NVIDIA Jetson Nano platform using a PC client app or mobile app and then operate a face recognition system via Camera connected to the server.

The following are the resources given to build a system.

Table 4. System Context

	Jetson Nano	Client
HW	• Jetson Nano Board • Camera module	• PC
Interface	• Wi-Fi • USB	• Wi-Fi
OS	• Ubuntu 18.04	• Windows 10
SW Module	• FaceRecDemoTCP	• RecvImageTCP
Data	• Video Stream • Captured Image • User ID/PW	• Video Stream • Captured Image • User ID/PW

4.2 Architectural Drivers

4.2.1 Use Case

We derive a Use Case of the system by analyzing customer requirements. The complexity of the system is not high because it aims to practice in the Security Specialist process. The function is also simple.

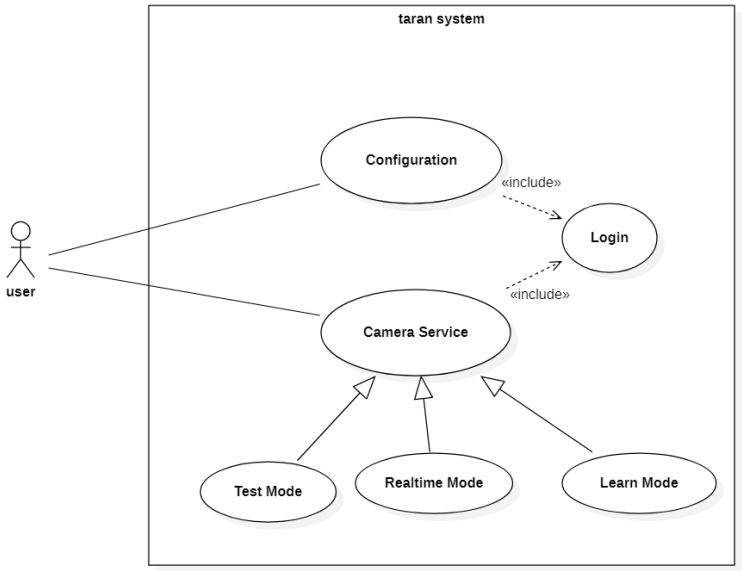


Figure 2. System Use Case

The user who is an actor can be a system administrator or a regular user.

- **Camera Service**
  We service the Client with the results of face recognition in the video entered from the camera.
- **Configuration**
  It provides the operation of the camera service and necessary parameters.
- **Test Mode**
  It provides camera images or streams for testing to camera services.
- **Realtime Mode**
  Send image recognition and results to all clients.
- **Learn Mode**
  The face is extracted from the video and recorded in the database.

4.2.2 Functional Requirement

Based on the above use case, we derive the following functional requirements: Functional requirements are largely divided into the entire system, the Client, and the Server.

Table 5: Functional Requirement - system

Part	**No.**	**Requirement**
System	RQ-GEN-01	Server and client should communicate normally.
	RQ-GEN-02	Secure or non-secure mode should be implemented between the server and the client.

	RQ-GEN-03	The status of connection should be reported (eg. fault/error detection, recovery, etc)

The requirements of the entire system point of view of the table, such as the three were identified.

Table 6: Functional Requirement - server

Jetson Nano (Server)	RQ-SVR-01	Server should be run as Learning, Run, Test Run Mode.
	RQ-SVR-02	In learning mode, the name of the person in front of the camera should be input, registered in the DB, and the result should be transmitted to the client.
	RQ-SVR-03	In Run Mode, the server should perform face recognition using the camera and transmit the result to the client.
	RQ-SVR-04	In Test Run Mode, face recognition should be performed using the given video file and the result should be transmitted to the client.
	RQ-SVR-05	The server should be able to map multiple photos of a user to a single ID.
	RQ-SVR-06	The server must listen to the Secure/Non Secure port for secure communication.

The requirements from the server perspective are identified from six tables as shown above.

Table 7: Functional Requirement - client

PC (Client)	RQ-CLI-01	In the client authentication screen, the SW should provide a UI that determines the Secure/Non Secure mode among the communication methods with the server.
	RQ-CLI-02	The client SW should provide a UI that determines the operation mode of the server - Learning, Run, Test Run Mode.
	RQ-CLI-03	In learning mode, the client must pass the username as input to the server .

Three requirements from the Client perspective are identified as shown in the table above.

4.2.3 System Architecture Design

Architecture is designed based on customer requirements as follows:

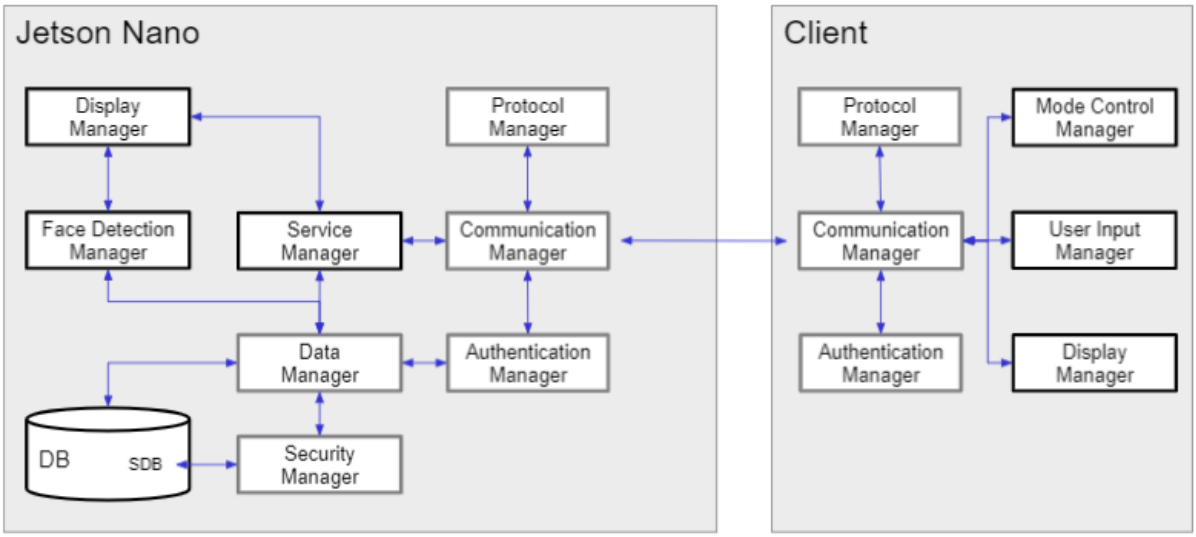


Figure 3. System Architecture

The Server system uses Jetson Nano, and the Client can use a PC or a Mobile Phone. In this assignment, we use PC and win32 app to save time.

Server Side

- **Service Manager**
  It is responsible for Lifecycle management, operation mode, connection management, etc. such as starting, shutting down, restarting services, etc..
- **Communication Manager**
  It is responsible for TCP communication.
- **Protocol Manager**
  Responsible for managing and de/serializing protocols to and from the Client.
- **Face Detection Manager**
  It is responsible for face recognition in the video inputted from the camera.
- **Display Manager**
  It processes camera input and Face Detection results on Display Buffer, and is responsible for Jpeg encoding for communication.
- **Data Manager**
  Manages database
- **Security Manager**
  It is responsible for Encryption/Decryption in the database.
- **Authentication Manager**
  Authentication Manager is responsible for user login..

Client Side

- **Communication Manager**
  It is responsible for TCP communication.
- **Protocol Manager**
  It is responsible for managing and de/serializing protocols to and from the Server. Inspect the packet through check the length and timestamp.
- **Authentication Manager**
  Send the user's id and password to the server and check if the user is a valid user. Also, check whether it is an admin account or a normal user account.
- **User Input Manager**
  Validates the user input value. Only letters and numbers can be entered, and there is a 10-character length limit.

- **Display Manager**
  Display the jpeg image received from the server on the screen.
- **Mode Control Manager**
  Allows the user to select one of Run mode, Learning mode, and test run mode.

4.2.4 System design – Sequence Diagram

The sequence diagram identifies the flow of Data/Control between the required protocols of the Server and Client and the SW Component. The first is the Sequence Chart on Secure Mode settings.

Case #1 set secure mode

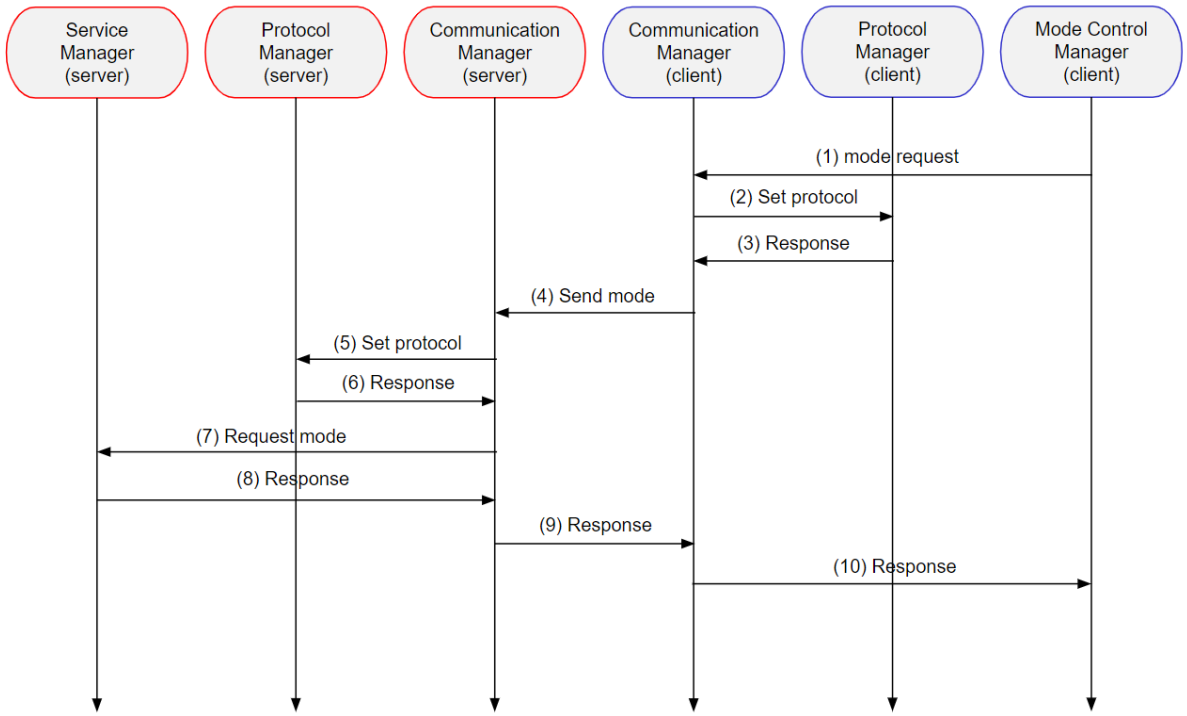


Figure 4 setting secure mode

The following is a sequence chart of the Login process when a secure channel is connected.

Case #2 login process in secure mode

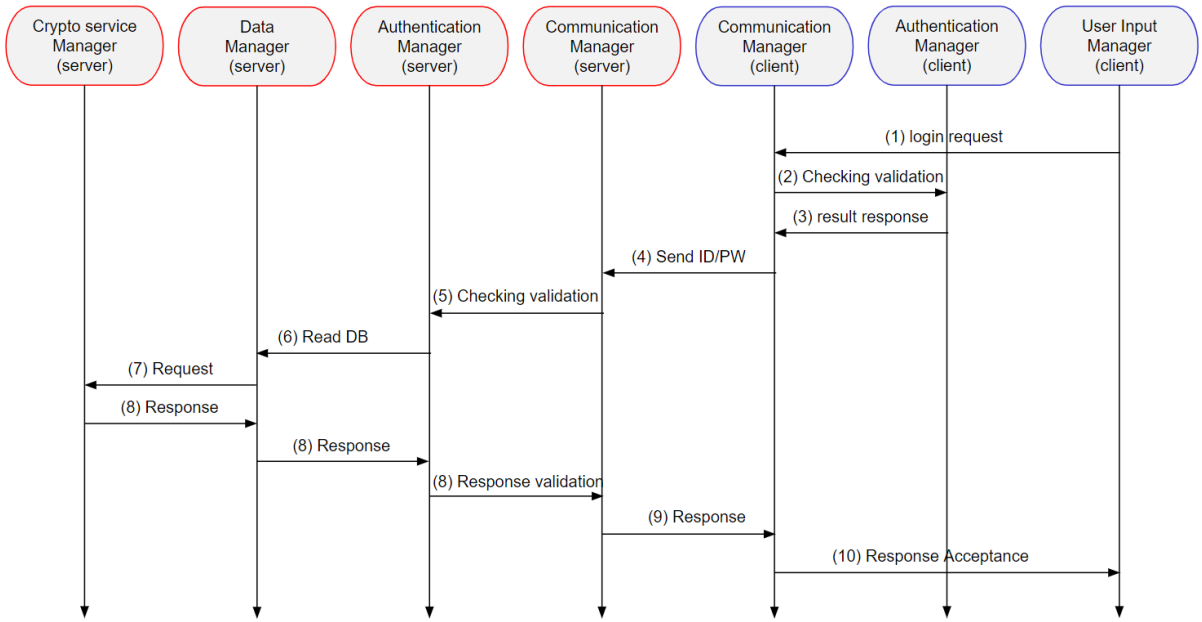


Figure 5. login process in secure mode

The following is a sequence chart of setting the operating mode of a server.

Case #3 mode selection in secure mode

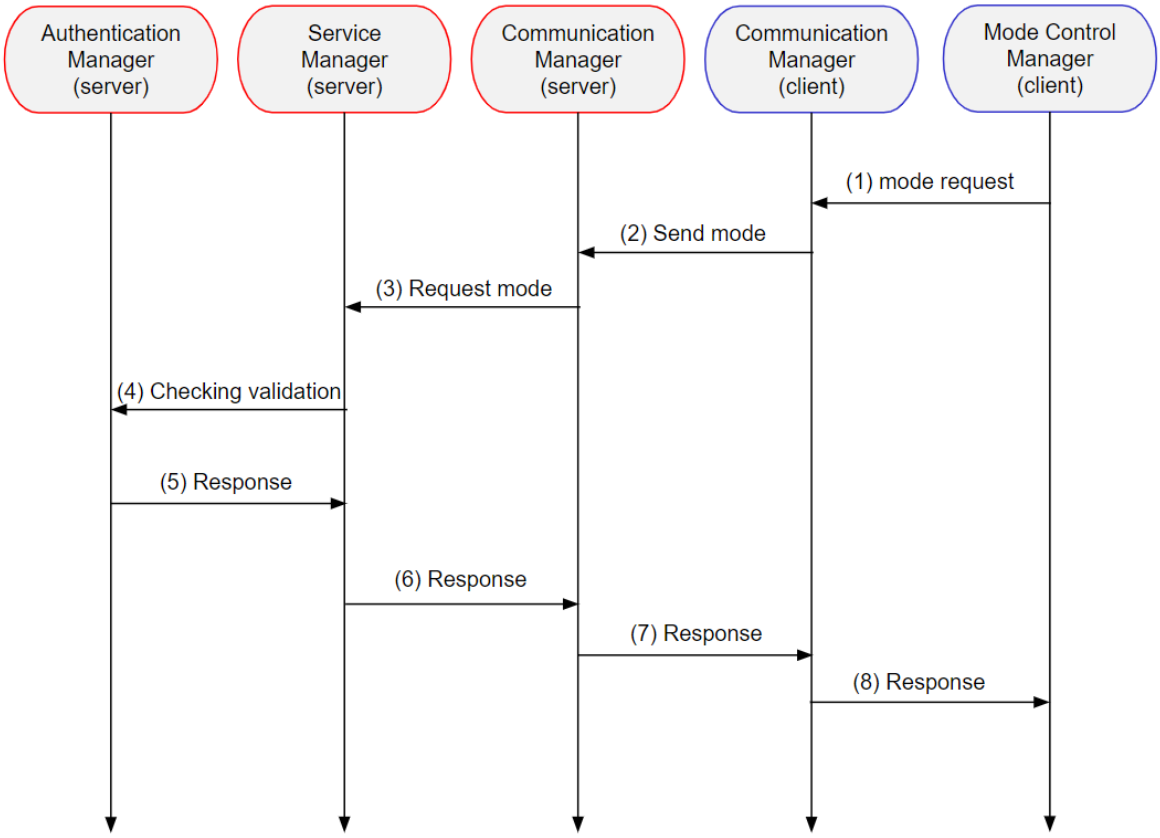


Figure 6. mode selection in secure mode

Case #4 learning mode in secure mode

This is a sequence diagram of learning mode. In Learning mode, the name and number of images to be used for learning are input from the user and delivered to the server. The server does image processing and, in the case of a new person, stores as many images as the number of images delivered to the user.

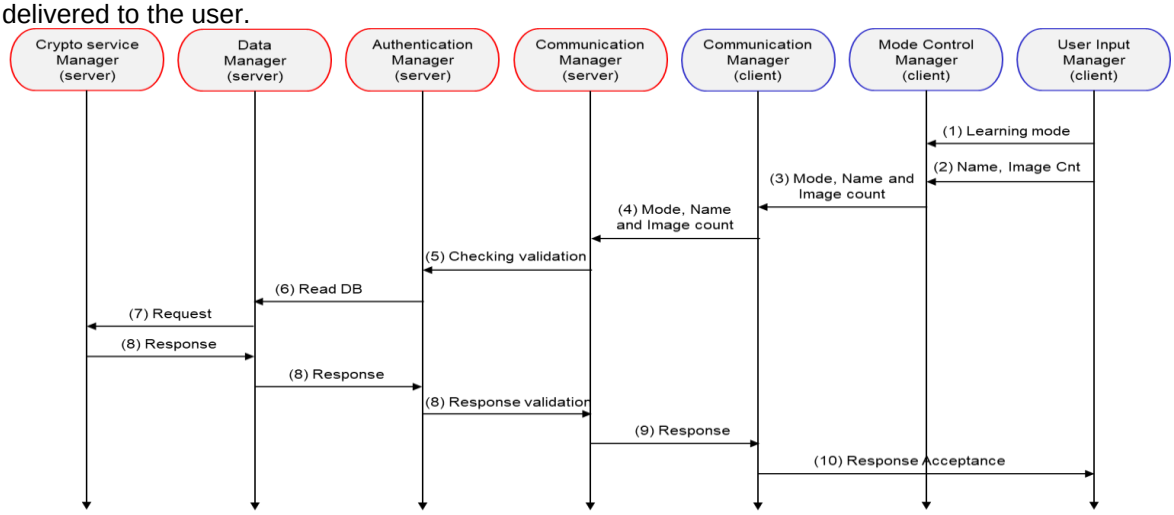


Figure 7 Learning Mode

Case #5 test run mode in secure mode

This is a sequence diagram of test run mode. In test run mode, it receives a list of videos to be played from the server. The user selects a file to play from the list and delivers it to the server, and the server transmits the selected video to the client.

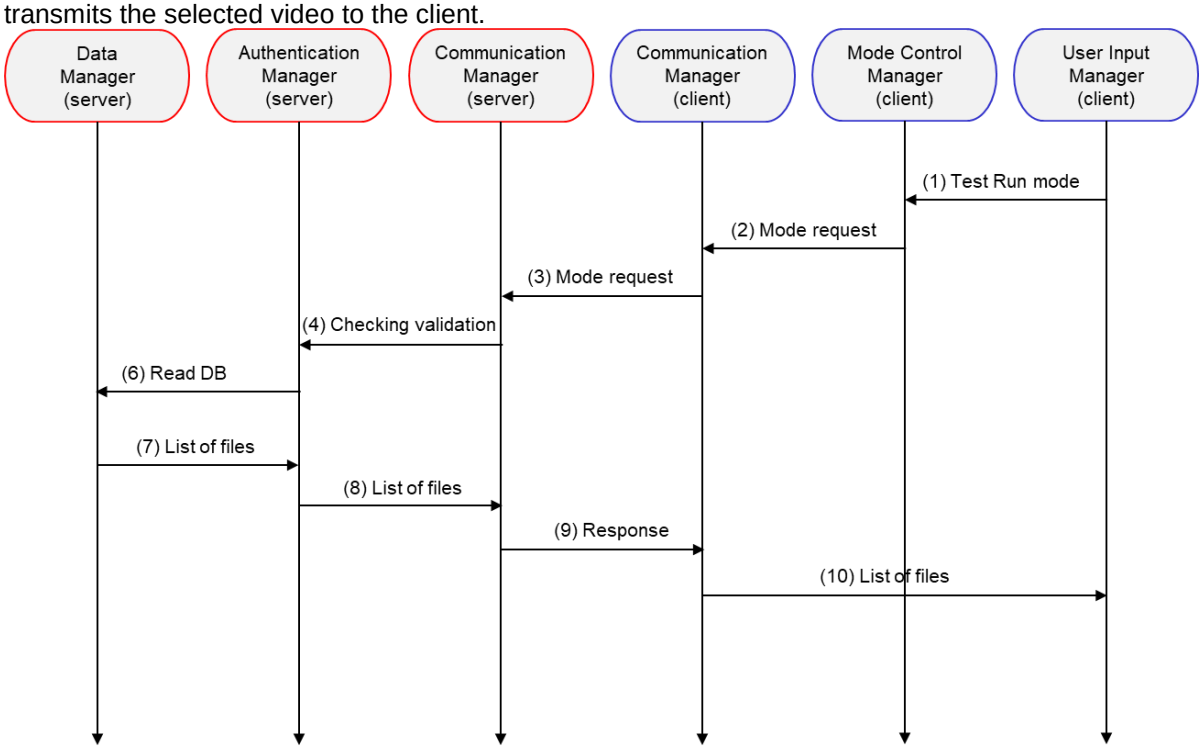


Figure 8 Test Run Mode

5. Phase1 - Secure Design

For the design of the secure system, the following steps are followed in turn to establish and incorporate Threat into the design.

5.1 Asset Identification

We derive from H/W and S/W that make up the system to identify assets that need to be protected from attackers. Both H/W and S/W are important assets, but H/W has been excluded due to the nature of this project.
Assets can be seen as what an attacker wants to steal, what a customer wants to protect, and what is needed or a means to access them.

- Things attackers want
- Things you want to protect
- Stepping stones to either of these

Table 8. Asset Identification

Part	**No.**	**Asset**
HW	Asset-HW-01	Camera
	Asset-HW-02	Jetson Nano
	Asset-HW-03	PC
SW	Asset-SW-01	User Information
	Asset-SW-02	Face Data Base
	Asset-SW-03	Image Information data (name, count, etc)
	Asset-SW-04	Control system data (mode, error, etc)

5.2 Risk Assessment

The following risks have been identified to identify, plan and incorporate them into the design. These risks are reflected in the requirement and reflected in the system implementation.

Table 9. Risk Identification

RISK ID	Condition	Consequence	Probability	Impact	Rating
RSK-01	Noise data from the network	The control system cannot quickly set the normal data.	9	2	18

RSK-02	Replay attack for Network communication	Attacker can use old packet for replay attack	5	10	50
RSK-03	Data flow sniffing	Attacker can see the data from camera to control system	7	8	56
RSK-04	Someone stole Jetson Nano board	The project will be stopped, and it takes a long time to buy a new one.	2	5	10
RSK-05	Face image and names stored in DB leaked	Personal information leakage	3	7	21
RSK-06	The network packet between the Jetson Nano board and the client pc is sniped	Information from DB data is leaked and hacker can be tricked into a registered member and access is successful.	9	9	81
RSK-07	One of your team members is ill or infected with Covid-19. Vacation due to emergency.	The patient could not participate in the project until fully recovered, increasing the stress on the remaining members. In the worst case, schedule delays or reduced project completion	6	7	42
RSK-08	Jetson Nano board or camera module is physically broken	Project cannot be completed because there is no replacement HW	5	10	50
RSK-09	Lost or broken development PC	Work is delayed	2	6	12
RSK-10	Design or threat modeling that differs from project goals	Failed to submit project results and must be completely redesigned and redeveloped	3	8	24
RSK-11	Saved DBs can be tampered with by hackers.	Security cameras recognize faces differently. For example, consider A as B.	2	10	20
RSK-12	Normal user get administrator right (root privilege) and change the system.	A normal user can put a malicious backdoor on behalf of the normal system.	3	10	30
RSK-13	Jetson Nano responds that it did not receive the data sent by the client.	The mode setting data sent by the client cannot be transmitted and may be set to another mode.	3	8	24

5.3 Threat Analysis

STRIDE is a model of threats developed by Microsoft. It provides a mnemonic for security threats in six categories. The threats are Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, and Elevation of privilege. It is used to identify more specific and systematic threats.

To use STRIDE, the DFD is first derived based on the design architecture and inputted into the MS tool. If a threat is found based on DFD, the review identifies the effective threat one by one.

The picture below shows DFD in MS Tool. By default, since threat is the interface part that can be intervened by three parties, the server and client are set to trust boundaries around the network.

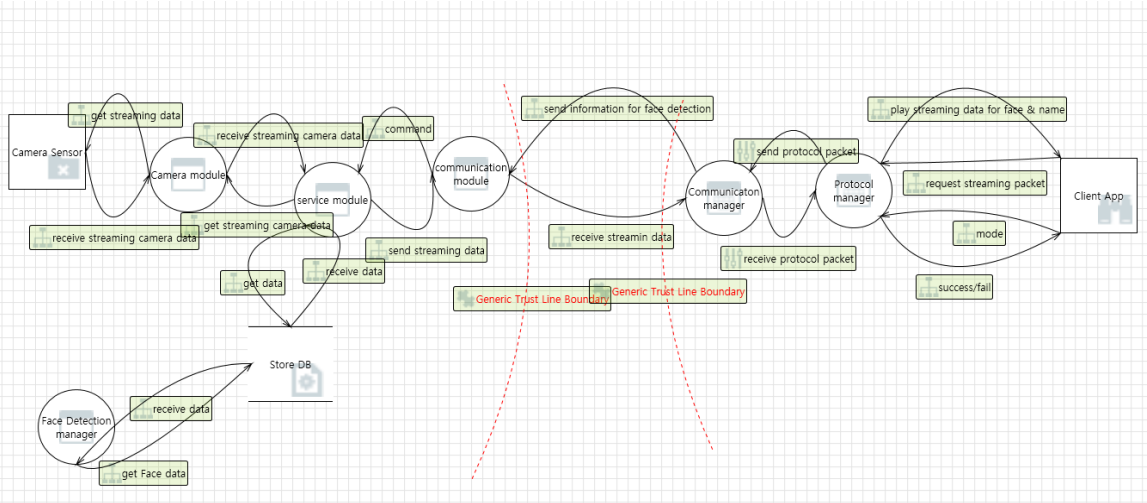


Figure 9. DFD for threat analysis

5.3.1 Threat modeling - STRIDE

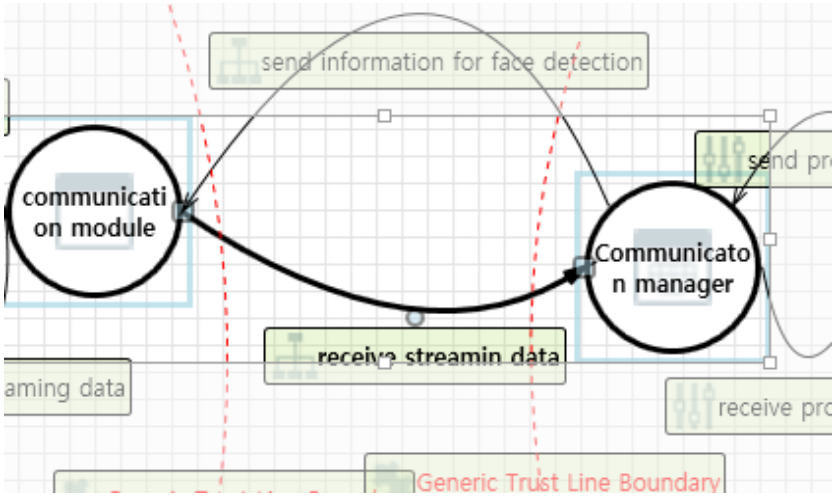


Figure 10. Threat Modeling #1

The assets associated with each candidate threat are listed in the table below. If it is not an asset to protect, it is excluded from the candidate list.

Table 10. Threat Modeling #1

Category	Description	Asset No.
Tampering	If communication module is given access to memory, such as shared memory or pointers, or is given the ability to control what Communication manager executes (for example, passing back a function pointer.), then communication module can tamper with Communication manager. Consider if the function could work with less access to memory, such as passing data rather than pointers. Copy in data provided, and then validate it.	Asset-SW-01 Asset-SW-02 Asset-SW-03 Asset-SW-04
	Data flowing across send information for face detection may be tampered with by an attacker. This may lead to a denial of service attack against communication module or an elevation of privilege attack against communication module or an information disclosure by communication module. Failure to verify that input is as expected is a root cause of a very large number of exploitable issues. Consider all paths and the way they handle data. Verify that all input is verified for correctness using an approved list input validation approach.	
	Data flowing across receive streaming data may be tampered with by an attacker. This may lead to a denial of service attack against Communication manager or an elevation of privilege attack against Communication manager or an information disclosure by Communication manager. Failure to verify that input is as expected is a root cause of a very large number of exploitable issues. Consider all paths and the way they handle data. Verify that all input is verified for correctness using an approved list input validation approach.	
Spoofing	Communication manager may be spoofed by an attacker and this may lead to unauthorized access to communication module. Consider using a standard authentication mechanism to identify the source process.	
	communication module may be spoofed by an attacker and this may lead to information disclosure by Communication manager. Consider using a standard authentication mechanism to identify the destination process.	
	communication module may be spoofed by an attacker and this may lead to unauthorized access to Communication manager. Consider using a standard authentication mechanism to identify the source process.	
	Communication manager may be spoofed by an attacker and this may lead to information disclosure by communication module. Consider using a standard authentication mechanism to identify the destination process.	
Repudiation	Communication manager claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.	

	communication module claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.	
Information Disclosure	Data flowing across receive streaming data may be sniffed by an attacker. Depending on what type of data an attacker can read, it may be used to attack other parts of the system or simply be a disclosure of information leading to compliance violations. Consider encrypting the data flow.	
	Data flowing across send information for face detection may be sniffed by an attacker. Depending on what type of data an attacker can read, it may be used to attack other parts of the system or simply be a disclosure of information leading to compliance violations. Consider encrypting the data flow.	
Elevation Of Privilege	An attacker may pass data into Communication manager in order to change the flow of program execution within Communication manager to the attacker's choosing.	
	communication module may be able to remotely execute code for Communication manager.	
	An attacker may pass data into communication module in order to change the flow of program execution within communication module to the attacker's choosing.	
	Communication manager may be able to remotely execute code for communication module.	
	Communication manager may be able to impersonate the context of communication module in order to gain additional privilege.	
	communication module may be able to impersonate the context of Communication manager in order to gain additional privilege.	
Denial Of Service	An external agent interrupts data flowing across a trust boundary in either direction.	
	Communication manager crashes, halts, stops or runs slowly; in all cases violating an availability metric.	
	An external agent interrupts data flowing across a trust boundary in either direction.	
	communication module crashes, halts, stops or runs slowly; in all cases violating an availability metric.	

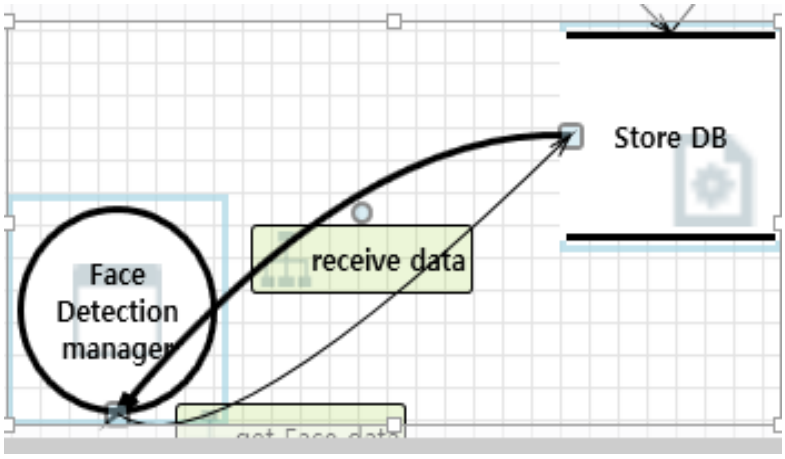


Figure 11. Threat Modeling #2

This is a threat analysis of the module that stores the picture in the image DB on the server. DB information is the most important information of this system.

Table 11. Threat Modeling #2

Category	Description	Asset No.
Spoofing	Store DB may be spoofed by an attacker and this may lead to incorrect data delivered to Face Detection manager. Consider using a standard authentication mechanism to identify the source data store.	Asset-SW-02 Asset-SW-03
	Store DB may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of Store DB. Consider using a standard authentication mechanism to identify the destination data store.	
Information Disclosure	Improper data protection of Store DB can allow an attacker to read information not intended for disclosure. Review authorization settings.	
Denial Of Service	Does service module or Store DB take explicit steps to control resource consumption? Resource consumption attacks can be hard to deal with, and there are times that it makes sense to let the OS do the job. Be careful that your resource requests don't deadlock, and that they do timeout.	

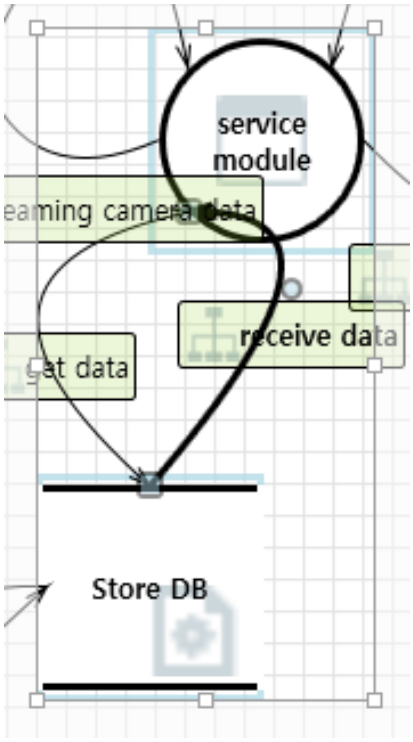


Figure 12. Threat Modeling #3

This is a threat analysis of the module that stores the username in the user DB on the server. DB information is the most important information of this system.

Table 12. Threat Modeling #3

Category	Description	Asset No.
Spoofing	Store DB may be spoofed by an attacker and this may lead to incorrect data delivered to service module. Consider using a standard authentication mechanism to identify the source data store.	Asset-SW-01 Asset-SW-04
Information Disclosure	Improper data protection of Store DB can allow an attacker to read information not intended for disclosure. Review authorization settings.	
Denial Of Service	Does Face Detection manager or Store DB take explicit steps to control resource consumption? Resource consumption attacks can be hard to deal with, and there are times that it makes sense to let the OS do the job. Be careful that your resource requests don't deadlock, and that they do timeout.	

5.4 Mitigations

Categorizing each risks according to the STRIDE and perform mitigation measure on them.

Table 13. Mitigations

Risk ID	Condition	Threat	Mitigation method

		Category	
RSK-01	Noise data from the network	Denial Of Service	Network size limitation - Protocol manager to analyze and respond to large network packet size attacks quickly and efficiently.
RSK-02	Replay attack for Network communication	Spoofing	Time stamp for Network packet - New time stamp for communication data
RSK-03	Data flow sniffing	Information Disclosure	TLS applies - TLS 1.3 with wolfSSL - After TLS session, all transmitted data becomes encrypted.
RSK-05	Face image and names stored in DB leaked	Tampering	TLS applies - TLS 1.3 with wolfSSL - After TLS session, all transmitted data becomes encrypted.
RSK-06	The network packet between the Jetson Nano board and the client pc is sniped	Tampering	TLS applies - TLS 1.3 with wolfSSL - After TLS session, all transmitted data becomes encrypted.
RSK-11	Saved DBs can be tampered with by hackers.	Tampering	Data encryption - AES_128_CBC - SHA_256
RSK-12	Normal user get administrator right (root privilege) and change the system.	Elevation Of Privilege	Minimum privileges for user - applying the minimum necessary privileges of file access to each user Close unused ports - Network ports should be blocked by default and only allowed if they are really needed for legitimate connections
RSK-13	Jetson Nano responds that it did not receive the data sent by the client.	Repudiation	Request and Response - Check the request and the appropriate response message - Re-request if there is no expected response
RSK-14	Attackers create so many connections that Jetson Nano cannot handle	Denial Of Service	Apply Firewall to Jetson Nano - Drop all connection except serviced port - Add DoS attack defences

5.5 Security Requirement

A security requirement is a statement of needed security functionality that ensures one of many different security properties of software is being satisfied. Security requirements are derived from industry standards, applicable laws, and a history of past vulnerabilities. Security requirements define new features or additions to existing features to solve a specific security problem or eliminate a potential vulnerability.

Security requirements provide a foundation of vetted security functionality for an application. Based on threat analysis, we establish secure requirements as a practical way to mitigate them. This secure requirement is merged with the existing functional requirement to determine the final spec.

Table 14. Security Requirements

Requirement ID	Descriptions	Related Risk ID
RQ-SEC-GEN-01	Authenticated communication should be implemented between server and client.	RSK-02 RSK-03 RSK-05 RSK-06 RSK-11
RQ-SEC-GEN-02	Secure mode should be implemented between server and client.	
RQ-SEC-GEN-03	In secure mode, all of data should be encrypted including time stamp.	
RQ-SEC-GEN-04	The server and the client must send and receive a request/response in the form of a message specified in the communication protocol.	RSK-13
RQ-SEC-SVR-01	In learning mode, the name of the person in front of the camera and the number of images to be collected must be input and registered in the DB.	RSK-01 RSK-11
RQ-SEC-SVR-02	In learning mode, images must be saved according to the number of images given.	RSK-01 RSK-11
RQ-SEC-SVR-03	Test Run Mode should not allow files other than the given video files.	RSK-01 RSK-11
RQ-SEC-SVR-04	The server must only allow the authenticated user can access the system through the authentication process including user ID/PW.	RSK-03 RSK-06
RQ-SEC-SVR-05	The server should store the user's ID, password, and authority in the DB.	RSK-03 RSK-06 RSK-12
RQ-SEC-SVR-06	The server must close the socket when authentication fails.	RSK-02 RSK-03 RSK-05 RSK-06 RSK-11
RQ-SEC-SVR-07	When the server is connected to either the secure port or the non-secure port, the other port must be closed.	
RQ-SEC-SVR-08	Only server administrator can change DB data of server.	RSK-12
RQ-SEC-SVR-09	The server must not be hang or crash due to an external DoS attack.	RSK-14
RQ-SEC-CLI-01	In learning mode, the client receives and transmits the number of images to be collected along with the user name.	RSK-01 RSK-11
RQ-SEC-CLI-02	The client should provide a UI to register the user ID/PW with the server.	RSK-03 RSK-06

RQ-SEC-CLI-03	The client must provide a UI to login to the server.	RSK-12

5.6 Quality Attributes Scenarios

Functionality and quality attributes are orthogonal. Functionality is the ability of the system to do the work for which it was intended. Achieving quality attributes must be considered throughout design, implementation, and deployment.

A quality attribute scenario is a quality-attribute-specific requirement. It consists of six parts.

- **Source of stimulus**.
  This is some entity (a human, a computer system, or any other actuator) that generated the stimulus.

- **Stimulus**.
  The stimulus is a condition that needs to be considered when it arrives at a system.

- **Environment**.
  The stimulus occurs within certain conditions. The system may be in an overload condition or may be running when the stimulus occurs, or some other condition may be true.

- **Artifact**.
  Some artifact is stimulated. This may be the whole system or some pieces of it.

- **Response**.
  The response is the activity undertaken after the arrival of the stimulus.

- **Response measure**.
  When the response occurs, it should be measurable in some fashion so that the requirement can be tested.

Below are security-related quality attribute scenarios.

Table 15. Quality Attributes

QA ID	Category	QA Six parts	Description	Related Requirement
SEC-QA1	Confidentiality	Stimulus	Attempt to intercept packets between server and client	RQ-SEC-GEN-01 RQ-SEC-GEN-02 RQ-SEC-GEN-03
		Source	Attacker in the middle	
		Environment	Normal network communication between server & client	
		Artifact	All data include user ID/PW & image data	
		Response	Data encryption	

		Response measure	100% encrypted data When a tester capture the packet and check the contents, he should not be able to guess the contents of the packet at all.	
SEC-QA2	Availability	Stimulus	Disguise someone's identity with stolen credentials (ID/PW)	RQ-SEC-SVR-04
		Source	Attacker disguised as a normal user	
		Environment	Normal log in operation	
		Artifact	User data from server	
		Response	2-factor authentication	
		Response measure	Reject login attempts in case of 2-factor authentication failure Even if the password is stolen, the server must be able to verify that the accessor is a user with valid privileges. 2fa authorized users must be 100% loginable.	
SEC-QA3	Integrity	Stimulus	Attempt to change DB data of server	RQ-SEC-SVR-05 RQ-SEC-SVR-08
		Source	General user to gain administrator privileges	
		Environment	General user access	
		Artifact	All data in DB	
		Response	Prohibition of escalation of privileges from general user to administrator in the system	
		Response measure	Re-execution of administrator authentication procedure when accessing user DB	

5.7 System Architecture with Security

Change the design as follows based on the identified Threat and risk.

The yellow part of the figure below shows enhanced security, and Firewall was added to the System Level to protect the system from DOS attacks.

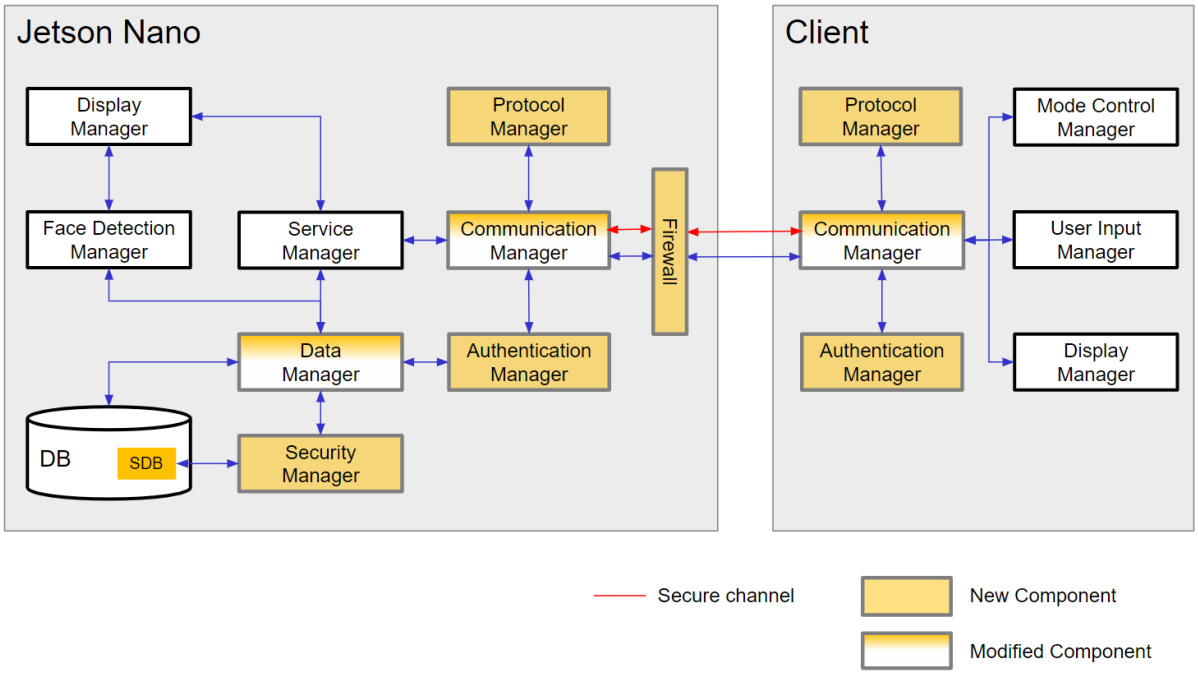


Figure 13. System Architecture

5.8 Design Modification with Secure Requirement

Table 16. Security Requirement

Security Design ID	Descriptions	Related Requirement ID
SD-01	Implementation of ‘Secure mode’ using TLS 1.3	RQ-SEC-GEN-02, RQ-SEC-GEN-03
SD-02	Implementation of ‘Protocol Manager’ module based on necessary data format	RQ-SEC-GEN-04
SD-03	Separation of administrator privilege to manage DB in learning mode	RQ-SEC-SVR-01, RQ-SEC-SVR-02 RQ-SEC-SVR-08
SD-04	Implemented a limited user operation	RQ-SEC-SVR-03, RQ-SEC-SVR-08
SD-05	Implementation of ‘Authentication Manager’ module based on authentication process	RQ-SEC-SVR-04
SD-06	Separation of ‘Authentication Manager’ domain to store credential data (user's ID/PW, authority)	RQ-SEC-SVR-05
SD-07	Modification of ‘Communication Manager’ to implement secure	RQ-SEC-SVR-06,

	mode	RQ-SEC-SVR-07
SD-08	Apply Firewall	RQ-SEC-SVR-09
SD-09	UI design considering secure mode	RQ-SEC-CLI-01, RQ-SEC-CLI-02 RQ-SEC-CLI-03

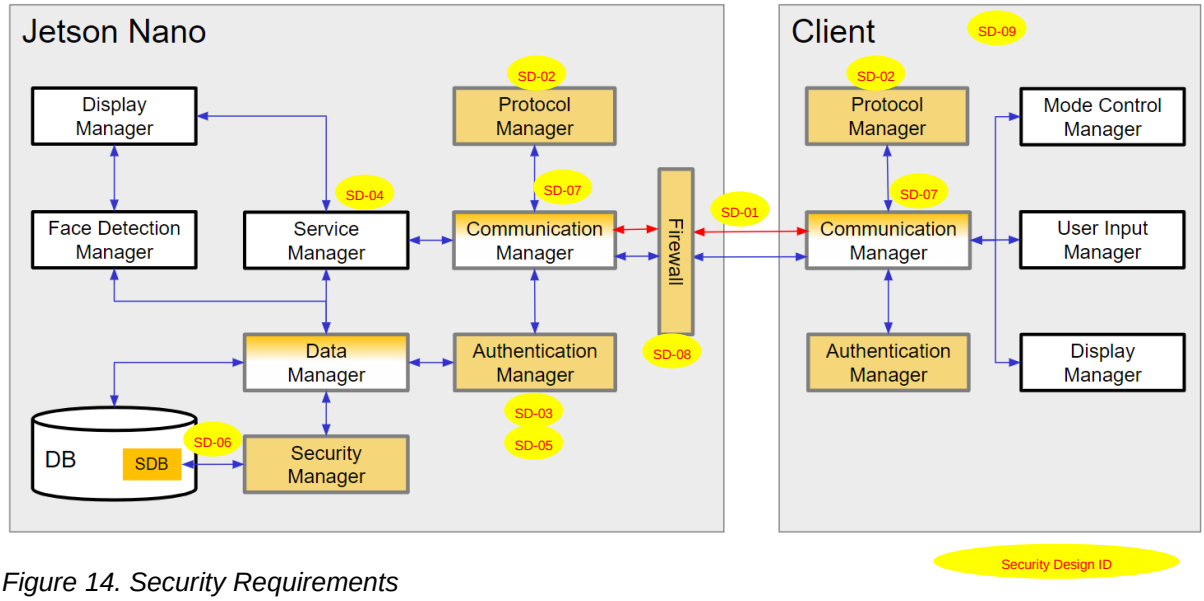


Figure 14. Security Requirements

5.9 Implementation

5.9.1 Secure Communication

For secure communication, all data communication between server and client must be encrypted. In addition, mutual authentication must be performed through public key authentication, and message integrity must be guaranteed. To ensure this, wolfssl, which implements the TLS 1.3 spec, is used. In this program, the port 50000 is used for unencrypted TCP traffic while port 55555 is used for encrypted traffic.

Transport Layer Security (TLS) is a cryptographic protocol designed to provide communications security over a network. The TLS protocol aims primarily to provide privacy and data integrity between two or more communicating computer applications. It runs in the application layer of the Internet and is itself composed of two layers: the TLS record and the TLS handshake protocols. It designed to prevent eavesdropping and tampering. The current version is TLS 1.3 defined in August 2018.

The wolfSSL embedded TLS library is a lightweight, portable, C-language-based SSL/TLS library targeted at IoT, embedded, and RTOS environments primarily because of its size, speed, and feature set. wolfSSL supports industry standards up to the current TLS 1.3, smaller than OpenSSL, offers a simple API, an OpenSSL compatibility layer, OCSP and CRL support, is backed by the robust wolfCrypt cryptography library.

Table 17. Security Design – SD01

No.	Requirement	Requirement No.
SD-01	Implementation of ‘Secure mode’ using TLS 1.3	RQ-SEC-GEN-02 RQ-SEC-GEN-03

Table 18. Security Design - TLS

	Design	Remark
Library	Use wolfSSL 4.7.1	* More useful in embedded ssl implementations(small, fast) * vulnerability free version (https://www.wolfssl.com/docs/security-vulnerabilities/)
TLS version	Only support 1.3	* Weak cipher suites have been removed * All handshake messages after the ServerHello are now encrypted → More Secure than prior version
Certificate	Generate Root-CA Certification, Server Certification	ECDSA prime256v1
Cipher suite	Only use TLS 1.3 Cipher suite	TLS_AES_256_GCM_SHA384. TLS_CHACHA20_POLY1305_SHA256. TLS_AES_128_GCM_SHA256. TLS_AES_128_CCM_8_SHA256. TLS_AES_128_CCM_SHA256.

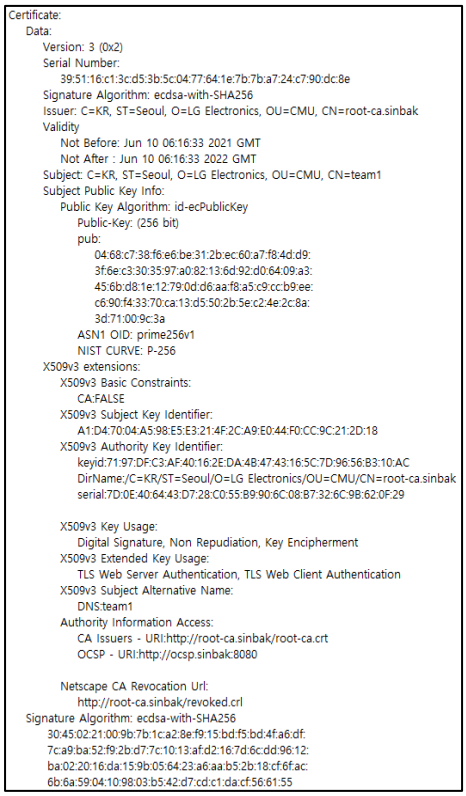


Figure 15. Certificate for TLS 1.3

5.9.2 Communication protocol

We have defined our own protocol for communication between server and client. Through this protocol, application-specific packets can be distinguished, and resiliency can be improved. To prevent tampering attacks, length checks are performed for each protocol item, and timestamp information is used to prevent replay attacks.

Table 19. Security Design - SD02

No.	Requirement	Requirement No.
SD-02	Implementation of ‘Protocol Manager’ module based on necessary data format	RQ-SEC-GEN-04

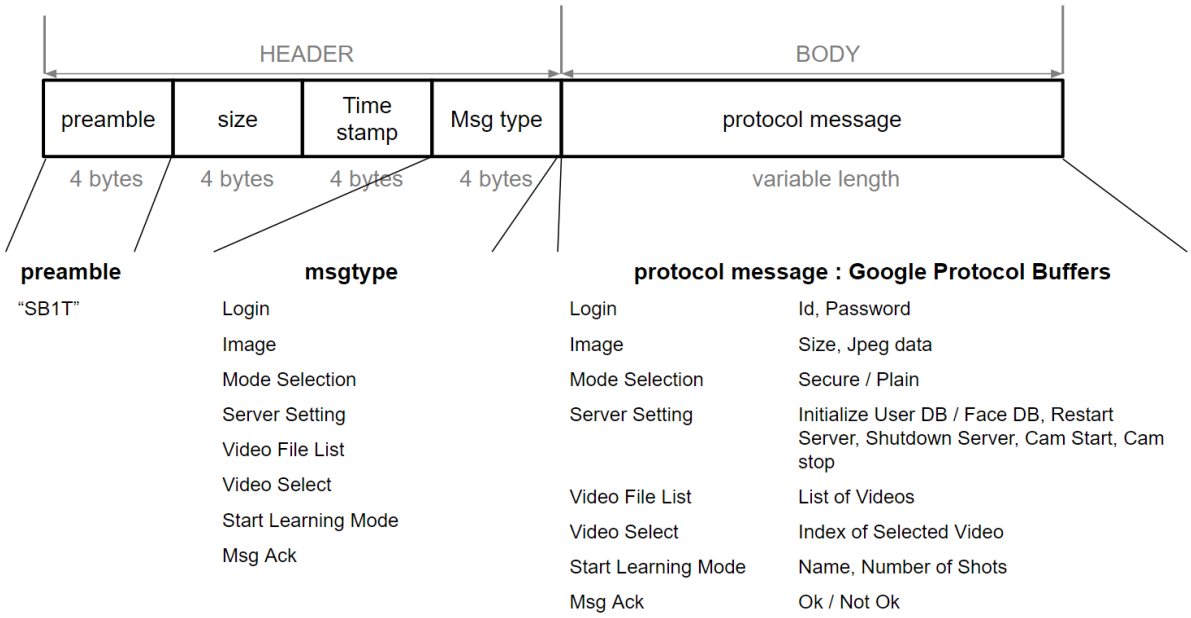


Figure 16. Protocol Definition

The above figure shows the defined protocol, the msg types are login, image, mode selection, server setting, and so on.

5.9.3 Authentication Manager

Permissions were classified by setting different permissions for each ID. Only users with administrator privileges can register new users through learning mode.

The user DB is encrypted so that attackers cannot see the data to prevent information disclosure and data tampering.

Table 20. Security Design - SD03

No.	Requirement	Requirement No.
SD-03	Separation of administrator privilege to manage DB in learning mode	RQ-SEC-SVR-01 RQ-SEC-SVR-02 RQ-SEC-SVR-08

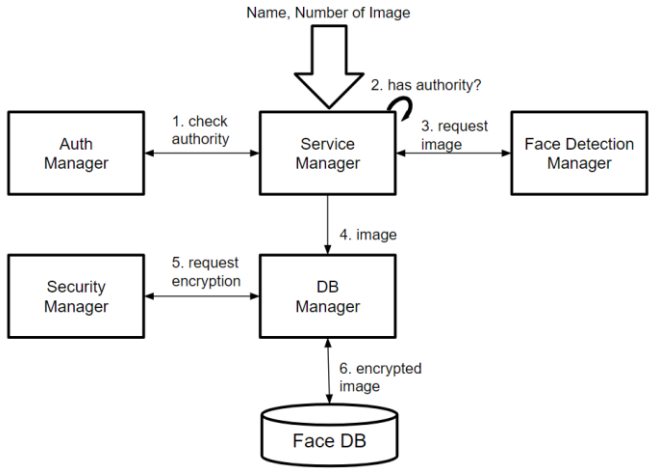


Figure 17. Secure Database

The figure above shows the procedure of learning mode, and only users with administrator privileges can register new users.

Table 21. Security Design - SD05,SD06

No.	Requirement	Requirement No.
SD-05	Implementation of 'Authentication Manager' module based on authentication process	RQ-SEC-SVR-04
SD-06	Separation of 'Authentication Manager' domain to store credential data (user's ID/PW, authority)	RQ-SEC-SVR-05

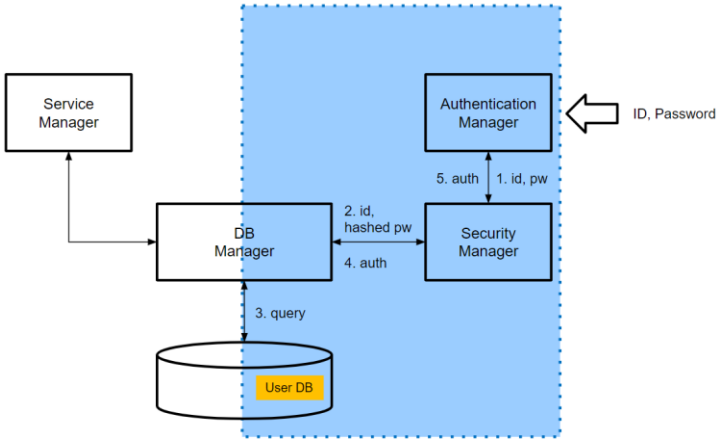


Figure 18. Authentication Manager

5.9.4 Privilege Management

Only a user with administrator privileges can run learning mode. Face data is encrypted and stored in the DB through SQL queries to prevent information disclosure.

Table 22. Security Design - SD04

No.	Requirement	Requirement No.
SD-04	Implemented a limited user operation	RQ-SEC-SVR-03 RQ-SEC-SVR-08

The figure below schematically shows that administrator privileges and general user privileges are displayed separately on the client.

Administrator

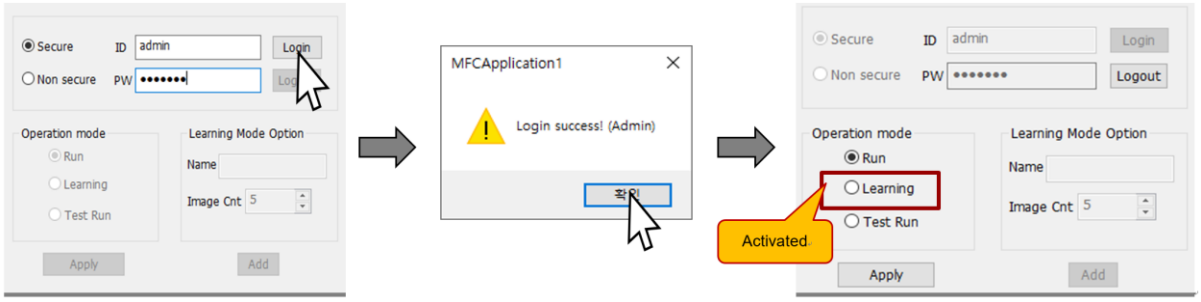


Figure 19. Client UI for Administrator

Normal user

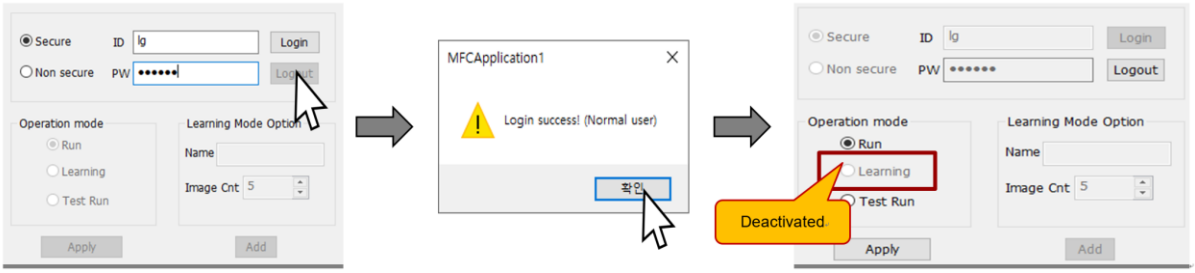


Figure 20. Client UI for Normal User

5.9.5 Communication Manager

Table 23. Security Design - SD07

No.	Requirement	Requirement No.
SD-07	Modification of ‘Communication Manager’ to implement secure mode	RQ-SEC-SVR-06 RQ-SEC-SVR-07

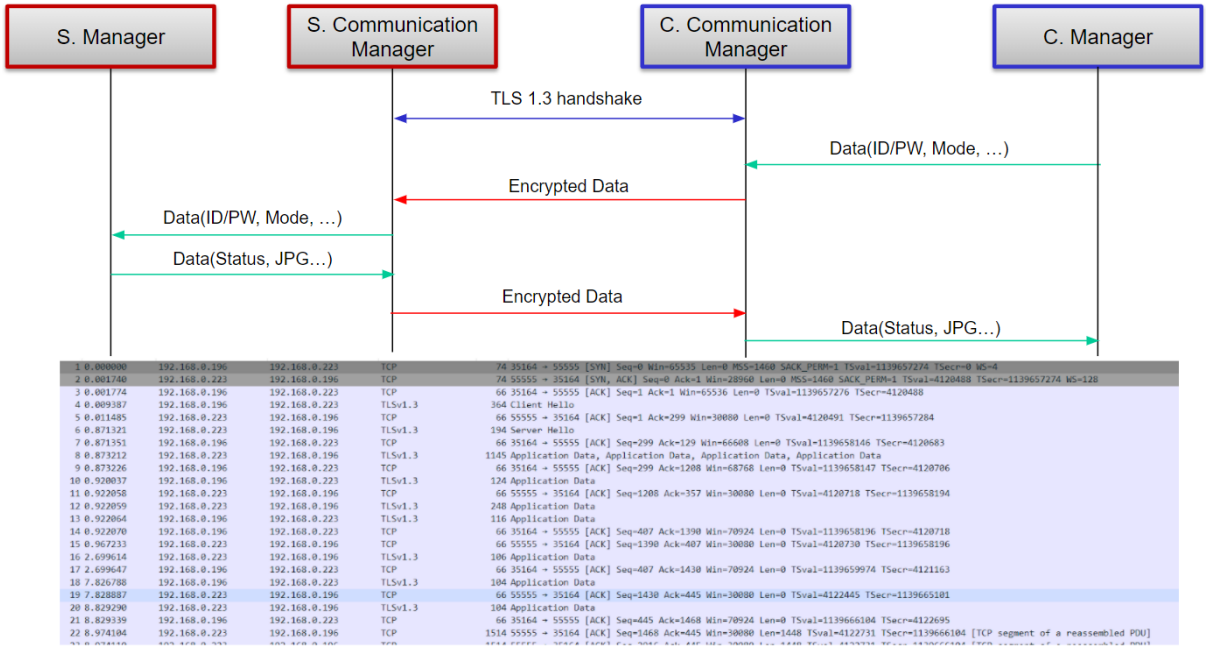


Figure 21. Secure Communication Manager

5.9.6 Defense Dos Attack

Apply a firewall to block unauthorized connections. Blocks access to ports except ports 22, 50000, and 55555.

Table 24. Security Design - SD08

No.	Requirement	Requirement No.
SD-09	UI design considering secure mode	RQ-SEC-CLI-01 RQ-SEC-CLI-02 RQ-SEC-CLI-03

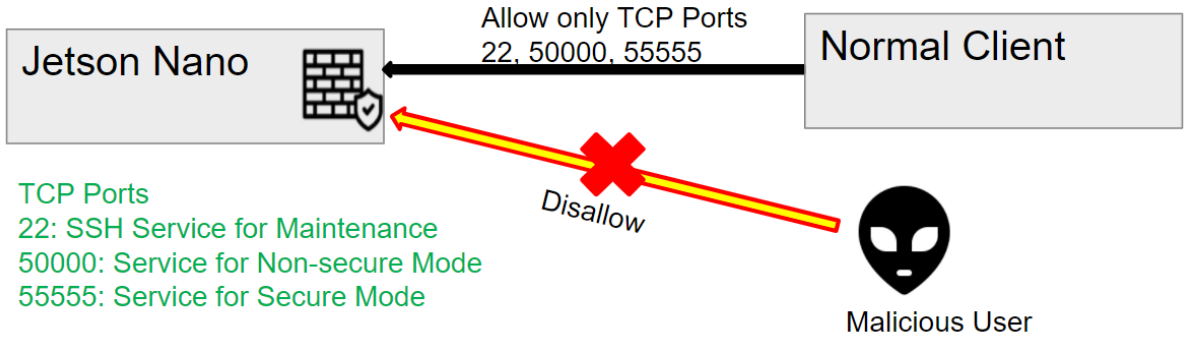


Figure 22. Firewall

Table 25. Security Design - SD09

No.	Requirement	Requirement No.
SD-09	UI design considering secure mode	RQ-SEC-CLI-01 RQ-SEC-CLI-02 RQ-SEC-CLI-03

6. **Phase1 - Test & Verification**

6.1 Functional Test

We validate the system against the functional requirements/specifications through functional test. To test each function of the software application, by providing appropriate input, verifying the output against the Functional requirements. This testing checks User Interface, APIs, Database, Security, Client/Server communication and other functionality of the Application Under Test.

We designed 24 test cases and ran the tests, all of them were passed.

https://drive.google.com/file/d/1avvoxG8JM5V3aDeXzDjdWqXfNR5TAA3E/view?usp=sharing

	A	B	C	D	E	F	G
1	Test Ca	Related Requ	Description	Precondition	Test Steps	Expected Result	Result
2	TC_01	RQ-SEC-GEN-02 RQ-SEC-GEN-03	Ensure all data is en	Running the Face Recognition Service on the server	1. User logs in to server in 2. Ensure data is encrypte	Captured packets are (	PASS
3	TC_02	RQ-SEC-SVR	Ensure User information is securely encrypted and stored in the DB	Connect ssh to	1. Verify user information (	USER's information sh	PASS
4	TC_03	RQ-SEC-SVR	The server must close the socket upon user authentication failure	Running the Face Recognition Service on the server	1. Verify that the server so 2. Client login with invalid 3. Check the server's sock	Socket status should n	PASS
5	TC_04	RQ-SEC-SVR	Ensure that servers	start firewall service Running the Face Recognition Service on the server User login and using face recognition service	1. Executing DoS Attacks ex) hping3 -S [server IP] -- 2. Check if face recognitio	Face recognition images should look normal	PASS
6	TC_05	RQ-SEC-SVR	Input validation	1. run program	1. input long length string 2. enter special characters	1. cannot enter more than 10 characters. (alphabet, numbers only)	PASS
7	TC_06	RQ-SEC-GEN	Operate either secur	1. run program	1. select either secure or r	1. only one mode can be selected	PASS
8	TC_07	RQ-SEC-SVR	Try log in with regist	1. run program	1. input registered id	1. success to log in	PASS
9	TC_08	RQ-SEC-SVR	Try log in with unreg	1. run program	1. input unregistered id	1. fail to log in	PASS
10	TC_09	RQ-SEC-SVR	Select operation mo	1. run program	1. log in with administrator	1. can select one of	PASS
11	TC_10	RQ-SEC-SVR	Select operation mo	1. run program	1. log in with user privilege	1. can select either	PASS
						1. enable Apply	

6.2Penetration Test

A penetration test is an authorized simulated cyberattack on a computer system, performed to evaluate the security of the system. The test is performed to identify vulnerabilities, including the potential for unauthorized parties to gain access to the system's features and data, as well as strengths, enabling a full risk assessment to be completed.

Our penetration test focus is on whether an attacker can tamper with network packets, escalate to administrator privileges, and steal stored photos and usernames. In the non-secure mode, an attacker can steal the user ID and password from the network packet and capture the image. However, in Secure mode, communication is performed using TLS 1.3, so the attacker could not capture the JPG file, and the server's DB data was protected through encryption.

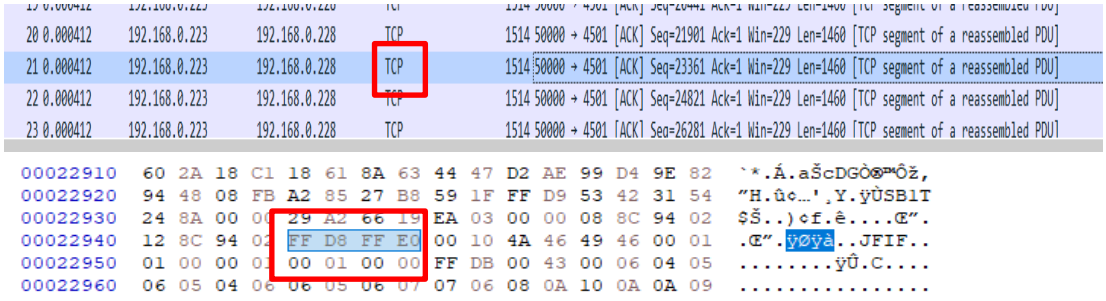


Figure 23 Packet dump in non-secure mode

Connection is not secure in Non-Secure mode. Attacker is able to get JPG in the middle

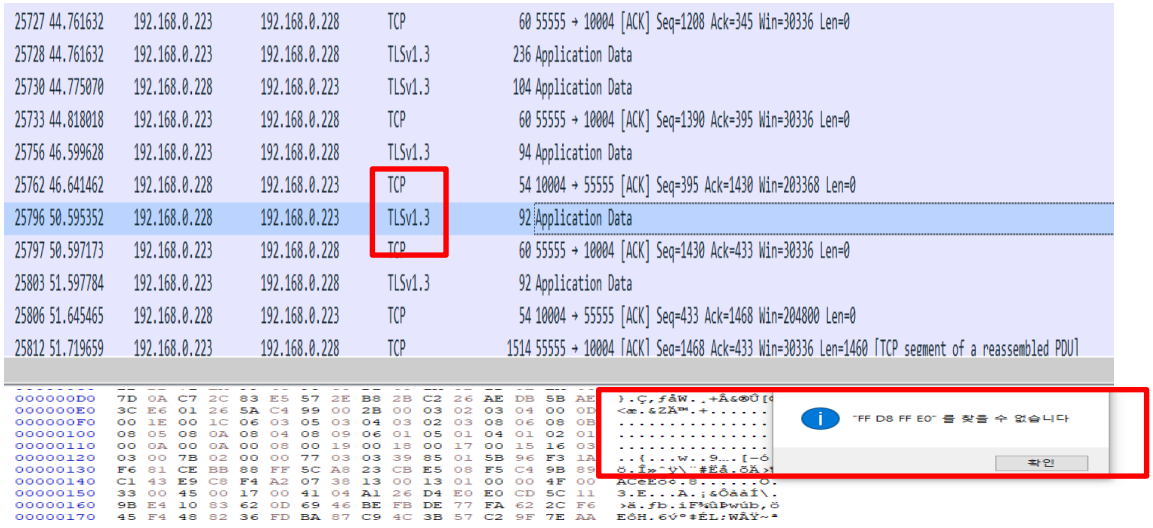


Figure 24 Packet dump in secure mode

Connection is protect by TLS 1.3 in Secure mode. Attacker can`t get JPG in the middle

7. Phase2 – Evaluation

7.1Introduction

As a team1, we used the output of team2 to evaluate the results. Source code and documents and test cases and their findings were given to us. We used these to identify risks, threats and vulnerabilities in the output of the team2 and tested for security.

The evaluation items are as follows:

- Design Analysis
  - Architecture Review
- Secure coding Analysis
  - Code Review
  - Static Analysis
  - Open Source Vulnerability Scan
  - Open Vulnerability Assessment Scanner
- Test
  - Dynamic Analysis
  - Fuzz test
  - Penetration test
  - Function Test

7.2 Design Analysis

7.2.1 Architecture Review

We conducted a comparison and architecture review with our outputs to see if they meet the customer's requirements or satisfy the design, and looked at whether STRIDE was considered.

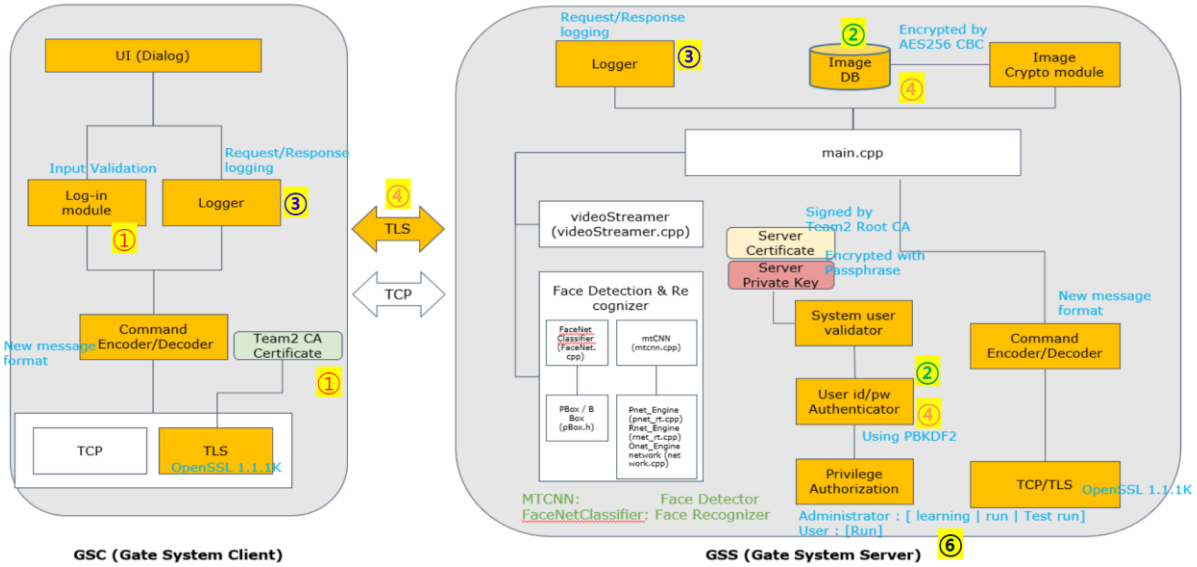


Figure 25. System Architecture of "Gate System"

The following are each review, and we found that dual Denial of Service items were not reflected in the design.

① **Spoofing**
Authenticate via TLS certificates between GSC and GSS

② **Tampering**
Encrypt User information and Image DB

③ **Repudiation**
Logging operation for non-repudiation

④ **Information disclosure**
Encrypt Sensitive data and communicate via TLS

⑤ **Denial of Service**
Could not find the design (ex. Firewall, Service manager, log rotation, etc)

⑥ **Elevation of Privilege**
Permission control by ID

7.2.2 Functional Requirement Review

We looked up the functional requirement document to confirm that the functional requirements were satisfied with the customer's requirements, but could not be found. So, based on the code implemented, we found that the following items were missing.

- In Learning Mode the interface should query for the name of the person in front of the camera and the number of samples to be collected.

7.3 Secure Coding Analysis

7.3.1 Code Review

In order to find vulnerabilities through developer code reviews, the entire team conducted code reviews focusing on changes compared to original codes. Through the code review, we found the problem as below, and we concluded that it needs to be corrected.

Ex #1. Insufficient size check: A crash occurs when dataLen is 0

```
int CSecurityDlg::HandleStreamData(unsigned int dataLen)
{
    unsigned int imagesize = dataLen;
    ssize_t readsize = 0;
    unsigned char* buff = NULL; /* receive buffer */
    CString str = _T("");

    buff = new unsigned char[imagesize];


    // decode image
    cv::imdecode(cv::Mat(imagesize, 1, CV_8UC1, buff), cv::IMREAD_COLOR, &(m_matImage));
    delete[] buff;
```

Figure 26 Engineer Code Review

Ex #2. Memory leak: Missing memory release when if status is false.

It doesn't matter as the program ends immediately, but it can cause problems afterwards.

```
if (m_allowedSystemCred.compare(out_hexstr) != 0)
    return false;

delete [] out_hexstr;
delete [] out_bin;
return true;
```

Figure 27. Missing memory release

7.3.2 Static Analysis - flaw finder

We perform static analysis to find vulnerabilities in a given source code. Flaw Finder and sonar cube were used, and the results of the flaw finder test are as follows. Seventy issues were found and 7 were found to require further examination.

Static Analysis with ‘flaw finder’.

Can check the CWE-based secure coding guide.

Out of 70 issues, 7 issues are meaningful flaws, the rest are considered as ‘False Positive’.

Analysis Summary

Hits = 70
Lines analyzed = 13555 in approximately 0.13 seconds (106361 lines/second)
Physical Source Lines of Code (SLOC) = 10142
Hits@level = [0] 79 [1] 24 [2] 41 [3] 0 [4] 5 [5] 0
Hits@level+ = [0+] 149 [1+] 70 [2+] 46 [3+] 5 [4+] 5 [5+] 0
Hits/KSLOC@level+ = [0+] 14.6914 [1+] 6.90199 [2+] 4.53559 [3+] 0.492999 [4+] 0.492999 [5+] 0
Dot directories skipped = 2 (--followdotdir overrides)
Minimum risk level = 1
Not every hit is necessarily a security vulnerability. You can inhibit a report by adding a comment in this form: // flawfinder: ignore Make *sure* it's a false positive! You can use the option --neverignore to show these.
There may be other security vulnerabilities; review your code!
See 'Secure Programming HOWTO' (https://dwheeler.com/secure-programs) for more information.

Figure 28 Analysis summary of flaw finder

The seven significant issues are mostly related to the following string manipulation related functions, and the direct analysis of the code showed that there was no logic problem and no significant content.

Ex #1. Does not check for buffer overflows with ‘sprintf’.

- ./LgFaceRecDemoTCP_Jetson_NanoV2/src/crypto_op.cpp:655: **[2]** (buffer) *sprintf: Does not check for buffer overflows (CWE-120). Use sprintf_s, snprintf, or vsnprintf. Risk is low because the source has a constant maximum length.*

```
sprintf(hexResult + (i * 2), "%02x", 255 & digest[i]);
```

Figure 29 string manupulation functions

7.3.3 Static Analysis – SonarQube

We spend considerable time performing static analysis with SonarQube. SonarQube needs to change the build environment of the arch64 environment to X86-based because it requires a Compile of Source.

To perform Static Analysis with SonarQube, the link process is not required during the build process, so not all required components are installed, and the required include files are taken directly from the device and placed in the building. We found that running SonarQube required 30% or more available space of the total storage device installed.

The following process is needed.

- Install NVidia Cuda Driver
- Install TensorRT
- Install CMake 3.7
- Install some include file from the device

The following results were obtained by performing Static Analysis

- No issue detected by Client.
- 1 Bugs, 2 Vulnerabilities, 14 Security Hotspots detected by Server.
- Can check various standards-based vulnerabilities (CERT, OWASP, Misra C++, etc).
- Can obtain compliance solutions.

It must be determined how far the source code is checked to perform Static Analysis. It is necessary to decide whether to include all open source codes or target only the modified parts. Under the agreement of all team members, it was concluded that only folders containing the current source would be targeted. It targeted a total of 45 files.

The above results are the results of Server Side's performance, and no significant issues were found as follows.

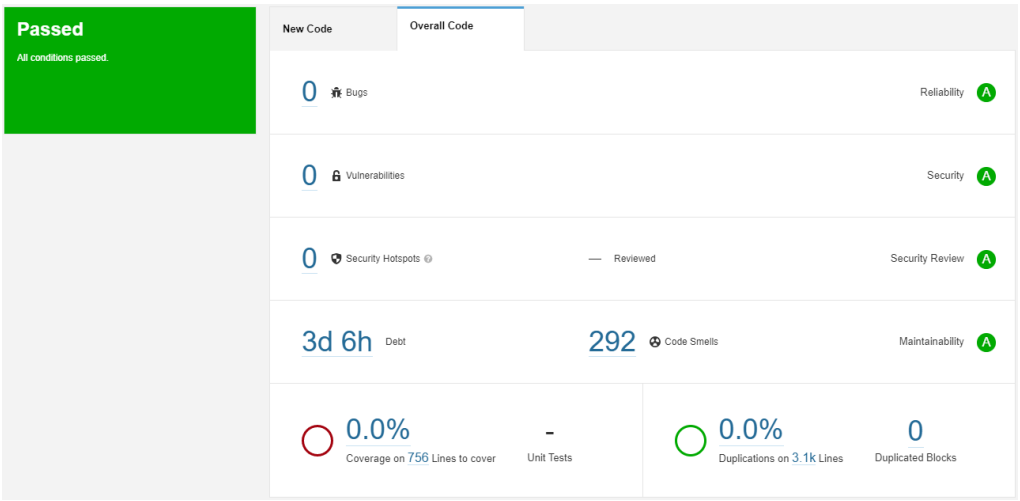


Figure 30 SonarQube result - Client Side

The following figure shows the results of Static Analysis on the Server side. Server side is one bug and two vulnerabilities and the rest is comment as below.

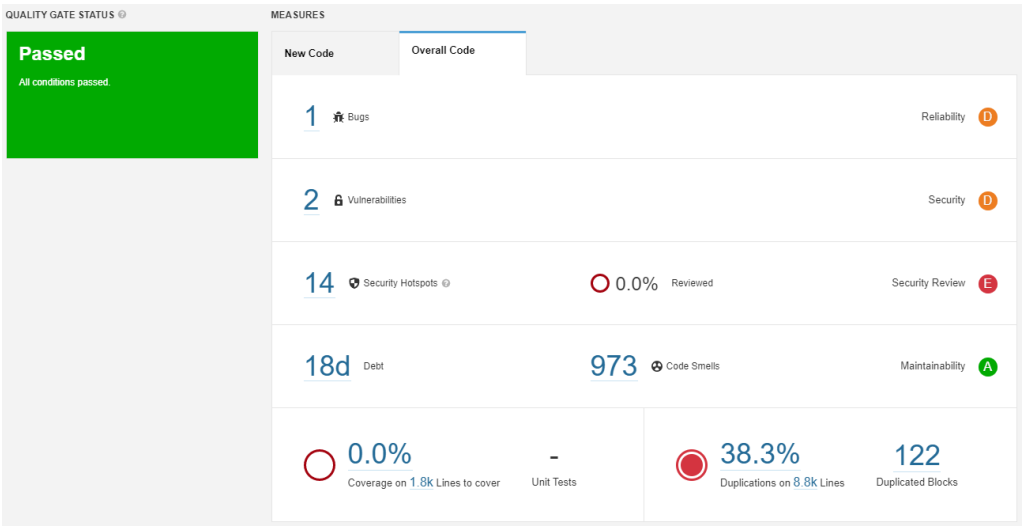


Figure 31 SonarQube result - Server Side

Each issue was analyzed in detail to confirm the above results. The bug, the most critical issue, is as follows and the analysis concluded that it is not a particularly problematic issue.

Issue review: Bug

If a pointer to an object is used as an exception, the code that will catch the exception may or may not have to delete the pointed-to object. This is even more complex in the exception case than in classical manual memory management, because of the distance between the `throw` statements and the matching `catch`.

Throwing by value is just simpler and less error prone.

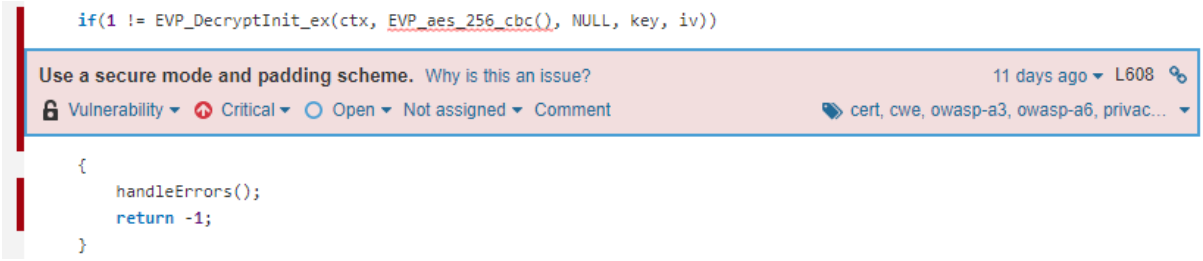
Compliant Solution

```
class E { /* Implementation */};
E globalException;

void fn ( int i )
{
  if ( i > 10 ) {
    throw ( globalException); // Throws a copy of the global variable
  }
  else {
    throw (E{} ); // Throws a new object
  }
}
```

Figure 32 Bug Review

Issue review: Vulnerability



Encryption operation mode and the padding scheme should be chosen appropriately to guarantee data confidentiality, integrity and authenticity:

- For block cipher encryption algorithms (like AES):

the GCM (Galois Counter Mode) mode which works internally with zero/no padding scheme, is recommended, as it is designed to provide both data authenticity (integrity) and confidentiality. Other similar modes are CCM, CWC, EAX, IAPM and OCB.

the CBC (Cipher Block Chaining) mode by itself provides only data confidentiality, it's recommended to use it along with Message Authentication Code or similar to achieve data authenticity (integrity) too and thus to prevent padding oracle attacks.

the ECB (Electronic Codebook) mode doesn't provide serious message confidentiality: under a given key any given plaintext block always gets encrypted to the same ciphertext block. This mode should not be used.

- For RSA encryption algorithm, the recommended padding scheme is OAEP.

Figure 33 Vulnerability Review

Both vulnerabilities have been identified in the same function, suggesting that En/Decryption with the AES256 CBC should be used together rather than alone with the CBC. In this case, the use of the GCM resolves the problem. However, since it is not the communication side that uses this function, it can be considered a non-significant result.

OpenSSL

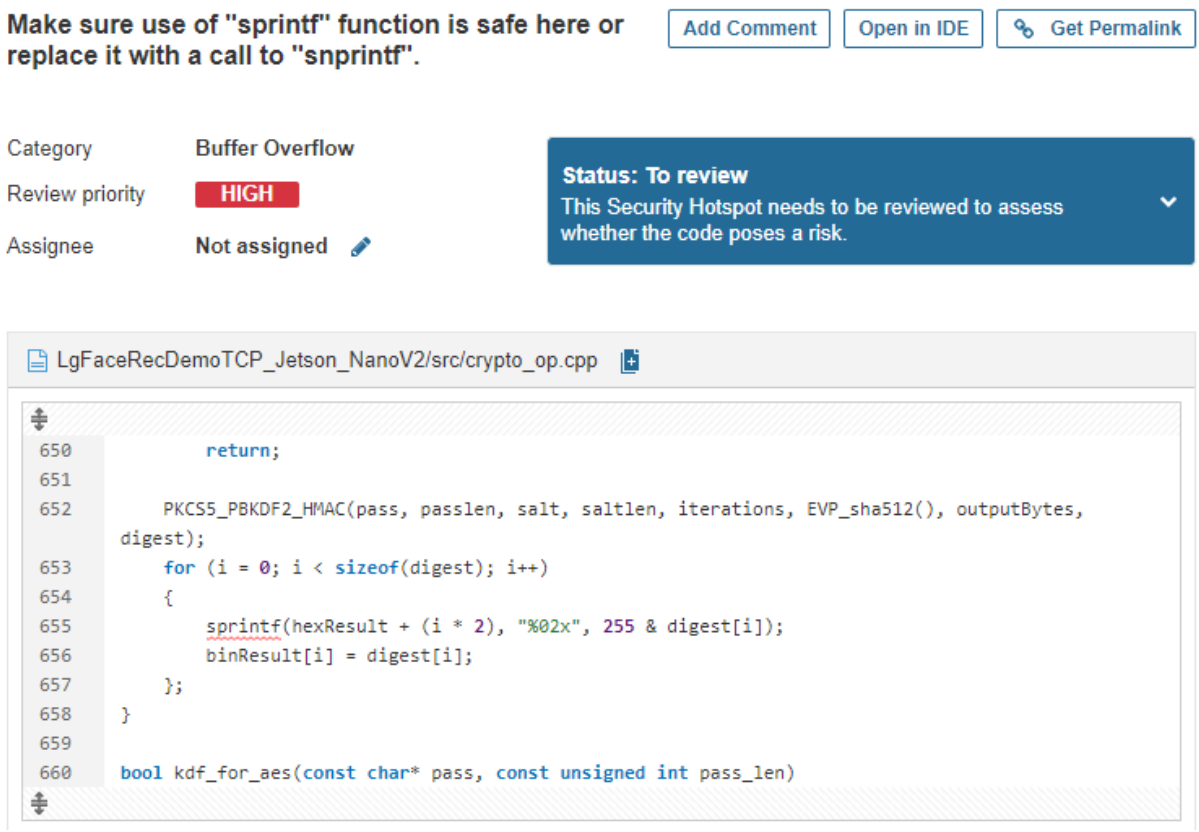
```
#include <openssl/evp.h>

// AES symmetric cipher is recommended to be used with GCM mode
EVP_aes_128_gcm() // Compliant

// RSA asymmetric cipher is recommended be used with OAEP padding
RSA_public_decrypt(flen, from, to, key, RSA_PKCS1_OAEP_PADDING); // Compliant
```

Issue Review: Security Hotspots

Finally, a review of the issues that correspond to comments. These are string manipulation functions, such as printf, retrieved from Flaw finder. These issues have already been reviewed using the results of the flaw finder and are considered false positives.



7.3.4 OSSVS(Open Source Security Vulnerability Scan)

Because the coverage of Static Analysis is limited to src folders, vulnerability analysis for all components used is not 100%. To overcome this, we search for issues registered in CVE, a known

vulnerability DB. The target components are listed below, and the search found that there was only one issue related to Nvidia Cuda.

Nvidia cuda

Printer-Friendly View

CVE-ID	
CVE-2020-5991	Learn more at National Vulnerability Database (NVD) • CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information
Description	
NVIDIA CUDA Toolkit, all versions prior to 11.1.1, contains a vulnerability in the NVJPEG library in which an out-of-bounds read or write operation may lead to code execution, denial of service, or information disclosure.	
References	
Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.	
• CONFIRM:https://nvidia.custhelp.com/app/answers/detail/a_id/5094 • URL:https://nvidia.custhelp.com/app/answers/detail/a_id/5094	
Assigning CNA	
Nvidia Corporation	
Date Record Created	
20200107	Disclaimer: The record creation date may reflect when the CVE ID was allocated or reserved, and does not necessarily indicate when this vulnerability was discovered, shared with the affected vendor, publicly disclosed, or updated in CVE.
Phase (Legacy)	
Assigned (20200107)	
Votes (Legacy)	

Figure 34 Open source CVE search

The version of cuda used in the build is 10.8. CVE scans have found these vulnerabilities in versions earlier than 11.1.1.
It seems that developers did not have been updated because of build dependency issue.

OpenSSL

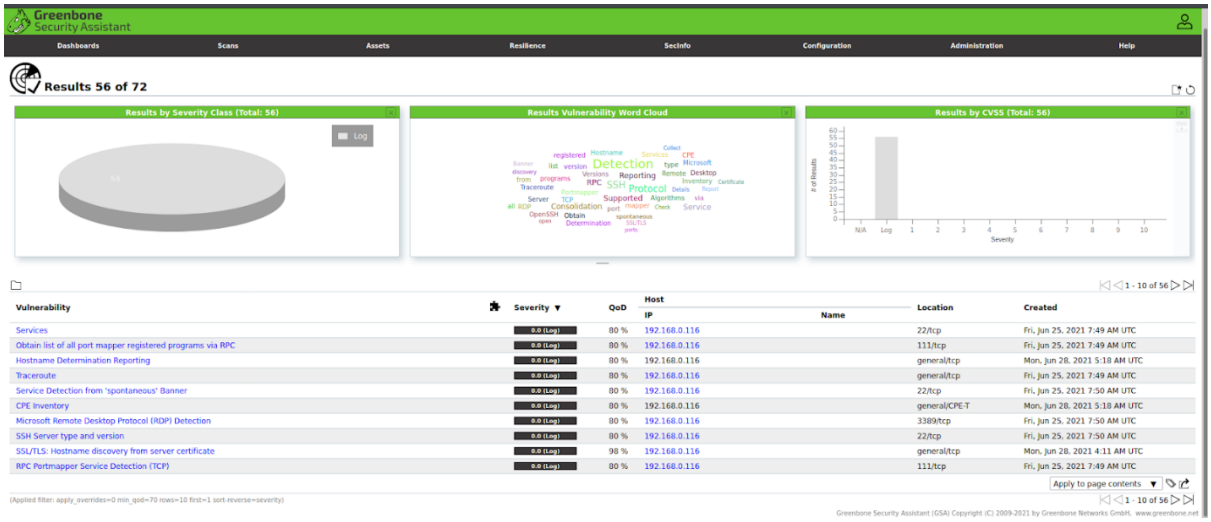
It is the openSSL 1.1.1k used in the build, and the CVE search found that there was a problem up to 1.1.1j, but the improvement was completed in 1.1.1k, confirming that it was not a problem.

- Gate system uses OepnSSL 1.1.1k version (latest version)
- Major changes between OpenSSL 1.1.1j and OpenSSL 1.1.1k
- ✓ Fixed a problem with verifying a certificate chain when using the X509_V_FLAG_X509_STRICT flag ([CVE-2021-3450])
- ✓ Fixed an issue where an OpenSSL TLS server may crash if sent a maliciously crafted renegotiation ClientHello message from a client ([CVE-2021-3449])

7.3.5 OpenVAS

OpenVAS is an acronym for open vulnerability assessment scan. Its capabilities include unauthenticated testing, authenticated testing, various high level and low level Internet and industrial protocols, performance tuning for large-scale scans and a powerful internal programming language to implement any type of vulnerability test. Test conducted using OpenVAS found two vulnerabilities as follows.

Service(port)	Subject	Vulnerability insight	CVE
SSH	OpenSSH <= 8.3p1 Command Injection Vulnerability	scp of OpenSSH allows command injection in spc.c via backtick characters in the destination argument.	CVE-2020-15778
General TCP	TCP Sequence Number Approximation Reset Denial of Service Vulnerability	The flaw is triggered when spoofed TCP Reset packets are received by the targeted TCP stack and will result in loss of availability for the attacked TCP services.component is used on an attacked system.	CVE-2004-0230



2 Results per Host

2.1 192.168.0.116

Host scan start Mon Jun 28 04:07:54 2021 UTC
Host scan end Mon Jun 28 04:20:59 2021 UTC

Service (Port)	Threat Level
22/tcp	Medium
general/tcp	Medium
22/tcp	Log
general/CPE-T	Log
22222/tcp	Log
111/tcp	Log
general/tcp	Log
3389/tcp	Log

Figure 35 Test Result of OpenVAS

3. Test – Dynamic Analysis

Dynamic Analysis with ‘ASan’.

- ‘Address Sanitizer’(ASan) can detect memory bugs
- No significant memory-caused crashes during testing on server/client

```
lg@LgFaceRecProject:~/jwlee/2team/specialist-team2/src/LgFaceRecDemoTCP_Jetson_NanoV2/build$ git diff
diff --git a/src/LgFaceRecDemoTCP_Jetson_NanoV2/CMakeLists.txt b/src/LgFaceRecDemoTCP_Jetson_NanoV2/CMakeLists.txt
index 23ace41..659be46 100644
--- a/src/LgFaceRecDemoTCP_Jetson_NanoV2/CMakeLists.txt
+++ b/src/LgFaceRecDemoTCP_Jetson_NanoV2/CMakeLists.txt
@@ -31,7 +31,7 @@ if(CUDA_VERSION_MAJOR GREATER 9)

 endif()

-set(CMAKE_CXX_FLAGS  "-Wno-deprecated-declarations")
+set(CMAKE_CXX_FLAGS  "-Wno-deprecated-declarations -fsanitize=address -fno-omit-frame-pointer")

 # tensorRT
 message("CUDA_TOOLKIT_ROOT_DIR = ${CUDA_TOOLKIT_ROOT_DIR}")
@@ -83,3 +83,4 @@ target_link_libraries(${PROJECT_NAME}
 )
 target_link_libraries(${PROJECT_NAME} ${OpenCV_LIBS})
 target_link_libraries(${PROJECT_NAME} OpenSSL::SSL OpenSSL::Crypto)
+target_link_libraries(${PROJECT_NAME} asan)
lg@LgFaceRecProject:~/jwlee/2team/specialist-team2/src/LgFaceRecDemoTCP_Jetson_NanoV2/build$
```

Figure 36 Makefile Configuration of ASAN

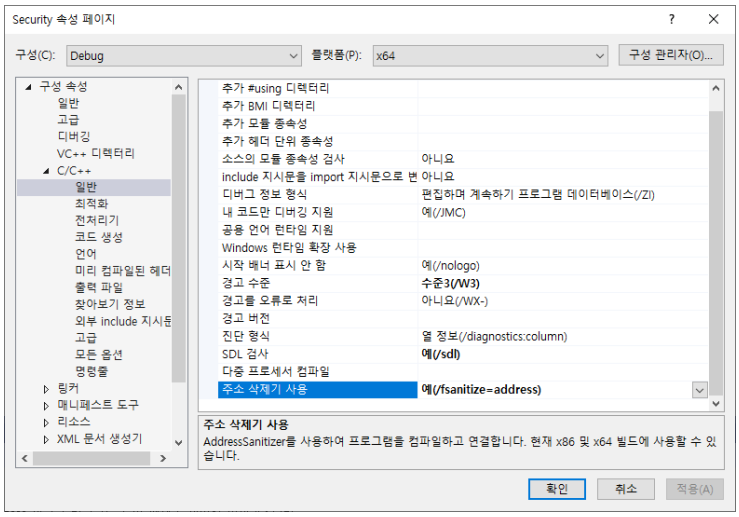


Figure 37 Configuration for ASAN(for Client)

7.4 Test

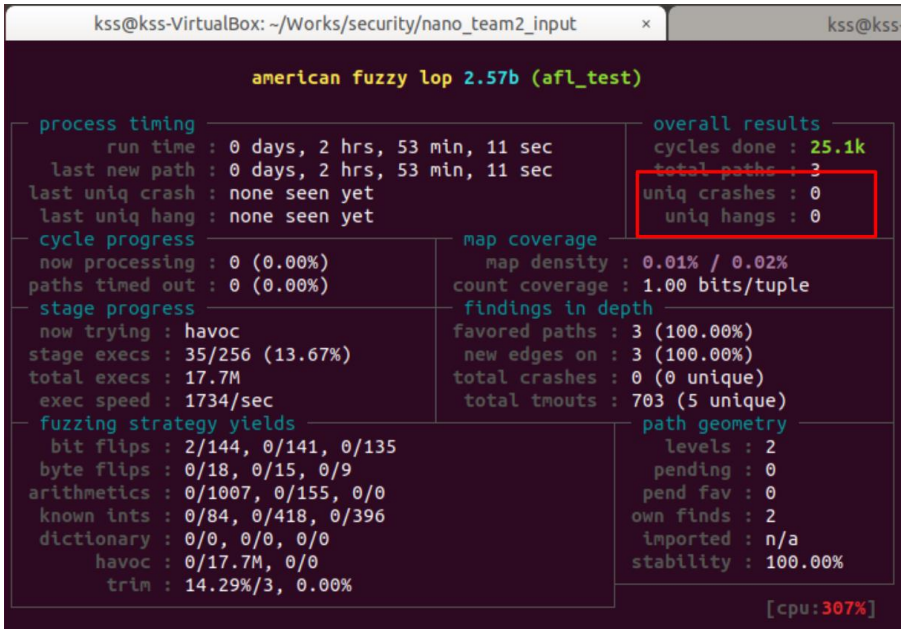
7.4.1 Fuzz Test (AFL)

Fuzz testing with ‘AFL’.

- Input validation is critical for security, so we performed AFL.
- The Server receives the port number and secure mode of operation from user.

```
portNum = atoi(argv[1]);
if(!strcmp(argv[2], "0")) bSecureMode = false;
```

- No Crashes, No hangs. But…



SEI CERT C Coding Standard

Use *strtol()* instead of *atoi()*

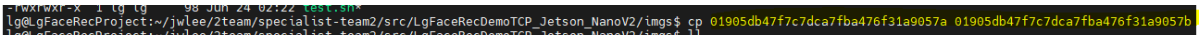
ERR34-C. Detect errors when converting a string to a number
Use one of the C Standard Library strto*() functions to parse an integer or floating-point number from a string. These functions provide more robust error handling than alternative solutions.

7.4.2 Fuzz Test (Manual Fuzz)

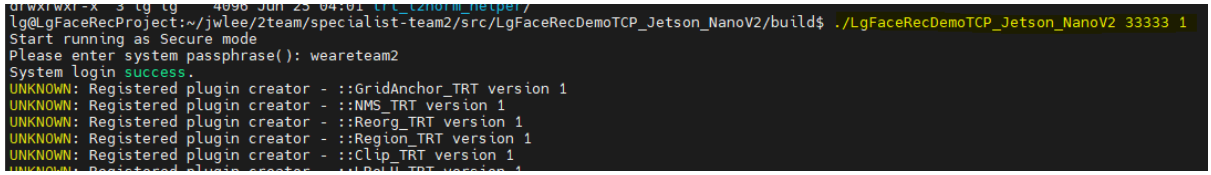
Fuzz testing with ‘Tampering’.

- The tampered data caused not only errors but also memory leaks.
- Error handling have to consider Memory leaks.

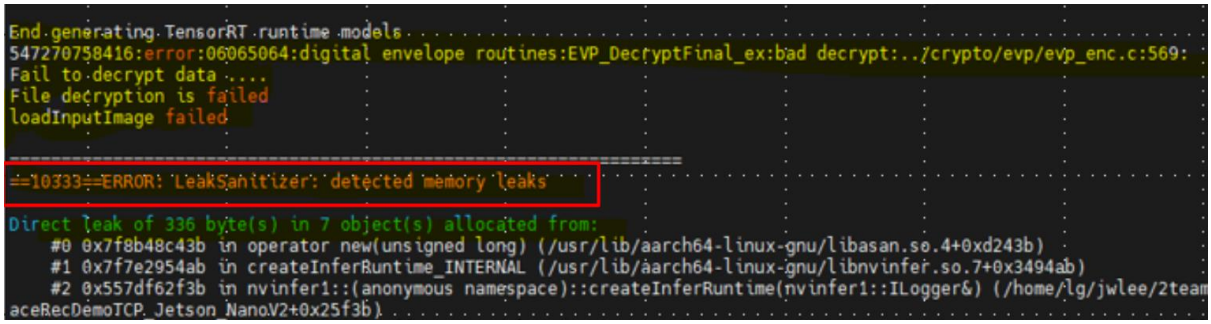
<Modify Image file>



<Run Server>



<Result>



7.4.3 Penetration Test

Service(port)	Subject	Vulnerability insight	CVE
General TCP	TCP Sequence Number Approximation Reset Denial of Service Vulnerability	The flaw is triggered when spoofed TCP Reset packets are received by the targeted TCP stack and will result in loss of availability for the attacked TCP services.	CVE-2004-0230

Penetration testing with ‘Dos Attack’.

- Attempted ARP(Address Resolution Protocol) spoofing
  → Client can`t receive Server`s data(face img,etc…)
- To avoid MITM (Man In The Middle Attack), firewall have to be considered.

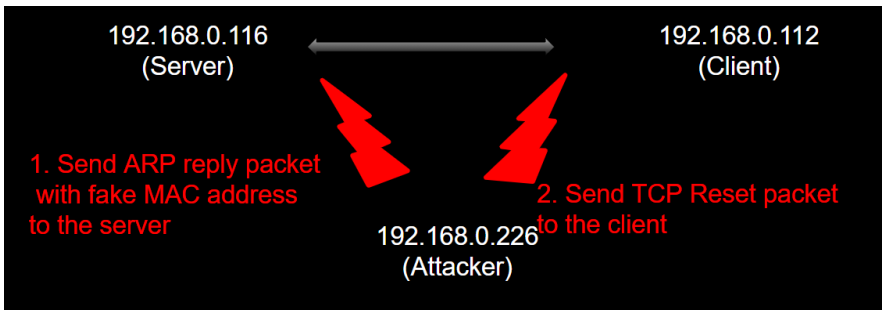


Figure 38 ARP Spoofing

Figure 39 Normal ARP cache of Server

Figure 40 Spoofed ARP cache of Server

7.4.4 Forensic Test

Full Memory Dump in Server

- Full physical memory dump and analysis
- Found some credential data in memory (passphrase, user name)

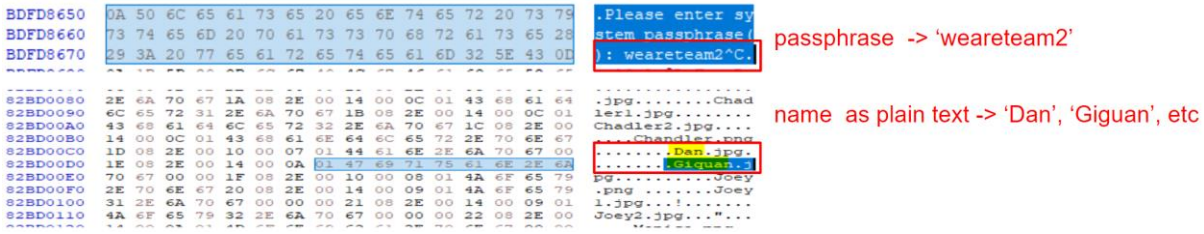
Memory dump using Lime(https://github.com/504ensicsLabs/LiME)

sudo insmod ./4.9.201-tegra/updates/dkms/lime.ko "path=/home/lg/jwlee/mem.lime format=lime“

lg@LgFaceRecProject:~/jwlee$ cat /proc/meminfo | grep MemTotal
MemTotal: 4059272 kB (<- 4GB memory)

lg@LgFaceRecProject:~/jwlee$ ll mem.lime
-r--r--r-- 1 root root 4259315808 Jun 29 05:10 mem.lime (<- full dumped)

Analysis dumped file using HxD(HxD - Freeware Hex Editor and Disk Editor | mh-nexus)
https://mh-nexus.de/en/hxd/



7.4.5 Functional Test

Testing with 'Test Case'.
Original test case by developers was passed all.

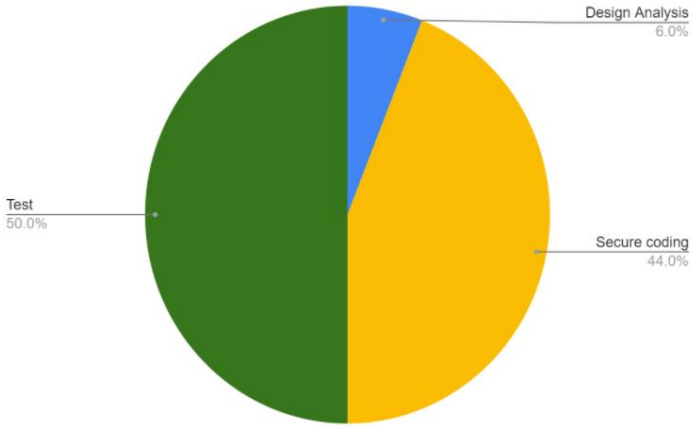
Operation mode stop	1. run program 2. log in	1. log in with user privileges	1. can select either Run or Test Run	PASS
Run as Test Run mode	1. run program 2. log in	1. log in 2. select Test Run 3. select one file 4. push Select button	1. selected file play	PASS
Run as Test Run mode	1. run program 2. log in	1. log in 2. select Test Run 3. don't select file 4. push Select button	1. show pop-up menu that Please select a video to play.	NA
Run as Test Run mode	1. run program 2. log in	1. log in 2. select Test Run	1. select button is activated	NA
Select Learning mode	1. run program 2. admin log in	1. log in as admin 2. select Learning mode	1. learning mode option is enabled	PASS
Learning mode input validation	1. run program 2. admin log in 3. learning mode	1. log in as admin 2. select Learning mode 3. input long length string to name	1. cannot enter more than 10 characters (alphabet only)	PASS
Learning mode input validation	1. run program 2. admin log in 3. learning mode	1. log in as admin 2. select Learning mode 3. input invalid number	1. only numbers 5 to 8 can be selected.	NA
Learning mode input validation	1. run program 2. admin log in 3. learning mode	1. log in as admin 2. select Learning mode 3. input invalid Name 4. click add button	1. show pop-up menu that Please enter a valid name. (Alphabet Only)	PASS
Learning mode disable add button	1. run program 2. admin log in 3. run or test mode	1. log in as admin 2. select Run or Test Run mode	1. add button disabled	PASS
Set server ip address	1. run program	1. input string not number 2. input number bigger than 255	1. can enter numbers only 2. input number only 0~255	PASS

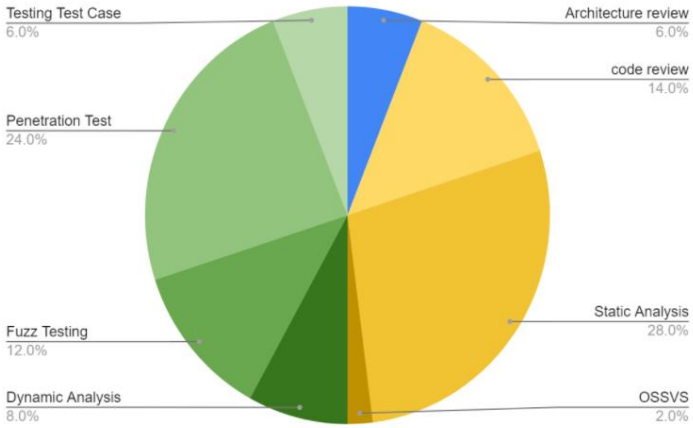
Some of Test Case are added to meet the security requirements as follows:
Requirements

- Learning Mode - User images can be added to the image database. In this mode the interface should query for the name of the person in front of the camera and the number of samples to be collected.
- Proper fault/error detection, recovery, and reporting.

No	Test case	Result	Description
24	In Learning Mode the interface should query for the name of the person in front of the camera and the number of samples to be collected.	Fail	The interface does not query the number of samples to be collected. **→ Failed to meet system requirements**
25	If the server is forcibly terminated, it should restart again.	Fail	If the server is forcibly terminated with 'sudo pkill' command, it was not restarted. **→ Insufficient system resiliency / robustness**

7.5 Evaluation Summary





Lesson	Activities	Vulnerabilities/Issues
Design Analysis	Architecture review	0
Secure coding	Eye inspection	3
	Flow finder	70 (but 63 is false positive)
	Sonar qube	17 (but 15 is false positive)
	OSSVS	0
	OpenVAS	2(but 1 is out of scope)
Test	Dynamic Analysis (ASAN)	0
	Fuzz Testing (AFL, Manual)	1
	Penetration Test (DoS, Memory)	2
	Testing Test Case	2

8. **Lessons and Learned**

8.1 Phase 1 - Development

Lesson	Activities	Learned
Asset Identification	Identifying assets Things attackers want Things you want to protect Stepping stones to either of these	Good: Knowing the assets that actually need to be protected can make an effort to constantly improve the quality of developments.
Security Risk Assessment	Brainstorming Security Risk and Rating 3 points of view Sniffing data flow Attack network communication Tampering DB data	Good: Listing the different scenarios of risk can have makes it clear what tasks need to be done Bad: Difficult to define exact criteria for scoring
Threat Analysis	Analysis threat using MS Threat modeling tool Drawing DFD Define Trust boundaries Categorizing threats with STRIDE	Good: The vague risk becomes clear through the tool Bad: Difficult to define exact criteria for scoring
Mitigation Threats	Establishing Threat Mitigation Plans Relay attack – add New time stamp Sniffing Data flow TLS 1.3 with wolfSSL Tampered DBs AES_128_CBC data encryption Elevation of Root privilege minimum privileges for user Attacking many connections Drop all except serviced port	Good: Possible to investigate various threat mitigation measures. Bad: Many of the mitigation measures initially determined were not implemented due to time reasons.
Security Requirements & Quality Attributes	Describe new security requirements and considering QA Confidentiality Integrity Availability	Good: Can guarantee security requirement with confidentiality, integrity, availability through Quality attribute process Bad: The boundary between requirements and QA was ambiguous, so it was not possible to accurately distinguish them.
Secure Coding & Static Analysis	Reviewing c++ code standard and selecting tool OWASP CERT CWE Misra-c selecting tool flawfinder	Good: Being able to look at and review the standards of secure coding Bad: Difficult to apply the tool to the ARM-based device system. Not spending more time on static analysis

	cppcheck	

8.2 Phase 2 - Evaluation

Lesson	Activities	Learned
Design Analysis	Architecture review	• Architecture should be designed considering all STRIDE.
Secure coding	code review	• Eye inspection code review are meaningful but have limited issue detection.
	Static Analysis	• Static analysis tool should consider several coding rule standards • Analysis process should be systematically managed.
	OSSVS	• The way to avoid open-source vulnerabilities is to always have the latest version.
Test	Dynamic Analysis	• Dynamic analysis can detect subtle flaws or vulnerabilities that cannot be detected by static analysis. • Dynamic and static analysis are complementary because a single approach cannot find all errors.
	Fuzz Testing	• Error handling have to consider Memory leaks.
	Penetration Test	• Trying to attack a system similarly to a malicious hacker can help you understand the importance of security.
	Testing Test Case	• A properly written requirement can create an accurate test case. • Vulnerabilities that could not be found through static analysis may be discovered through testing.

9. **Artifacts**

- Github : https://github.com/shinpark-security/tartan